



# Better Pseudo-labeling with Multi-ASR Fusion and Error Correction by SpeechLLM

Jeena Prakash\*, Blessingh Kumar\*, Kadri Hacioglu, Bidisha Sharma, Sindhuja Gopalan, Malolan Chetlur, Shankar Venkatesan, Andreas Stolcke

Uniphore Systems, India & USA

{jeena, blessingh, kadri.hacioglu, bidisha, sindhuja, malolan.chetlur, shankar.venkatesan, andreas.stolcke}@uniphore.com

## Abstract

Automatic speech recognition (ASR) models rely on high-quality transcribed data for effective training. Generating pseudo-labels for large unlabeled audio datasets often relies on complex pipelines that combine multiple ASR outputs through multi-stage processing, leading to error propagation, information loss and disjoint optimization. We propose a unified multi-ASR prompt-driven framework using postprocessing by either textual or speech-based large language models (LLMs), replacing voting or other arbitration logic for reconciling the ensemble outputs. We perform a comparative study of multiple architectures with and without LLMs, showing significant improvements in transcription accuracy compared to traditional methods. Furthermore, we use the pseudo-labels generated by the various approaches to train semi-supervised ASR models for different datasets, again showing improved performance with textual and speechLLM transcriptions compared to baselines.

**Index Terms:** speech recognition, pseudo-labeling, semi-supervised ASR, transcription, LLM, speechLLM.

## 1. Introduction

To achieve good generalization, an ASR model must be trained on diverse datasets that capture a wide range of accents, dialects, genres, and other speech patterns. Although semi-supervised and self-supervised learning methods have reduced the need for high-quality audio datasets for model training, supervised ASR models are still crucial for commercial applications in many domains. Obtaining such datasets by human annotation is costly and time-consuming.

Speech data synthesis [1] is a promising way to address data scarcity, enabling advancements in domain adaptation, recognition of rare names, numeric transcription, and low-resource languages [2–4]. Recent work leverages LLMs for text generation and multi-speaker TTS models for speech synthesis [5]. However, effective integration requires high-quality TTS models that produce naturalistic audio, as excessive synthesized data can degrade ASR performance on spontaneous and conversational speech.

Large amounts of untranscribed audio data are often available, but the lack of transcriptions limits their usability. A common approach is iterative pseudo-labeling, where unlabeled data is transcribed over multiple iterations as the acoustic model evolves [6]. The accuracy of pseudo-labels depends on the strength of the base ASR model, and its errors can propagate to the final model.

Another common approach for transcribing large audio datasets is combining multiple ASR models. Ensemble methods have been widely used to enhance ASR performance.

The NIST recognizer output voting error reduction (ROVER) method integrates ASR outputs through a voting mechanism [7, 8], while other approaches employ machine learning techniques for model fusion [9, 10]. Additionally, researchers have explored combining ASR models of various architectures at the hypothesis level to improve accuracy [11, 12].

Few studies have explored combining multiple end-to-end (E2E) ASR models. It is often assumed that the extensive parameters of a single E2E ASR model provide enough flexibility to handle diverse speech domains, including noisy and varied speech styles. However, different E2E models exhibit varying accuracy across domains and accents. Hojo et al. [13] integrate acoustic information from multiple E2E ASR models with an external language model (LM) tailored to the target domain. Text-based methods for improving ASR transcriptions include LM rescoring [14] and neural LM-based error correction, which converts incorrect recognitions to ground truth sentences [15, 16]. Additionally, sequence-to-sequence multimodal ASR error correction models have been proposed [17–19]. However, ensemble and LM-based approaches require careful tuning to align with the error patterns of the underlying ASR model.

The rise of generative LLMs has sparked growing interest in using them as ASR correctors [20–24]. Authors of [25] leverage N-best lists generated by ASR and perform LoRA fine-tuning of LLM for generating best hypothesis. Instead of 1-best or N-best hypotheses, [26] utilize word confusion networks generated by the ASR system and perform ASR error correction demonstrating improved performance. Hu et al. [27] further extend textual approaches by developing a multimodal model that incorporates discrete audio tokens as an additional input. Here, we consider a speechLLM architecture [28], a versatile recent variant of multimodal LLMs, based on continuous audio embeddings. However, there is a lack of comparative studies evaluating the effectiveness of LLM-based generative approaches against multi-ASR ensemble methods for pseudo-label generation and ASR error correction.

In this work, we conduct a comparative study of three distinct approaches for generating pseudo-labels from untranscribed audio data, which are then used to train ASR models. First, we introduce a *multi-ASR ensemble pipeline*, which integrates three well-established large-scale end-to-end ASR models. We hypothesize that each ASR model brings unique perspectives to transcription generation, and by effectively combining them, we can produce more accurate hypotheses. Next, we propose a *multi-ASR textual LLM-based architecture*, where the prompt includes the confusion sets generated from the three ASR models. While this method improves upon the ensemble approach by refining transcriptions, it disregards the acoustic information present in the original speech. Although acoustic or

\*These authors contributed equally to this work

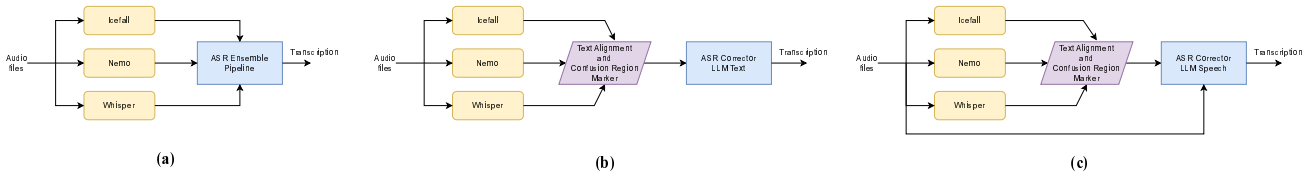


Figure 1: Comparison of different approaches for generating pseudo labels, (a) Multi-ASR ensemble pipeline, (b) Multi-ASR textual LLM-based architecture, (c) Multi-ASR speechLLM-based architecture.

confidence scores from the ASR models can partially compensate for this limitation, incorporating the original speech signals into the correction process remains a key challenge. To address this, we propose a novel *multi-ASR speechLLM-based architecture*, which adopts a more comprehensive approach leveraging both textual hypotheses and acoustic evidence. This model is finetuned to learn from disagreements among the ASR ensemble. Finally, we validate that transcriptions generated by the strategies proposed achieve performance comparable to human annotation, when used for ASR training. Multi-ASR speechLLM-based error correction provides an effective way to utilize large-scale untranscribed audio data in semi-supervised ASR training.

The rest of the paper is organized as follows. In Section 2, we describe the three approaches for generating pseudo labels. Section 3 presents the database, experimental setup and results and we provide our conclusions in Section 4.

## 2. Architectures for Pseudo-Labeling

We describe three architectures for the generation of pseudo-labels for untranscribed speech: (a) a multi-ASR ensemble pipeline, (b) multi-ASR with textual LLM postprocessing, and (c) multi-ASR with speechLLM postprocessing, as illustrated in Figure 1(a), 1(b) and 1(c), respectively. All approaches integrate three ASR models—Custom Icefall model, Nemo Parakeet, and OpenAI Whisper. The conventional ensemble pipeline applies a complex cascade of rule-based processing to derive final transcriptions. In contrast, the textual LLM-based approach simplifies this by directly structuring ASR outputs for LLM finetuning. The speechLLM-based approach further enhances this by incorporating both textual and speech inputs, leveraging audio evidence for more accurate predictions.

### 2.1. Multi-ASR Ensemble Pipeline

In this framework, we perform the fusion of outputs from all three recognizers and use the consensus of a majority at the word level to generate a final transcription. As shown in Figure 1(a), we pass the audio concurrently through the three ASR models, namely, Icefall\*, Nemo Parakeet\*, and OpenAI Whisper\*. As shown in Figure 2, the multi-ASR ensemble pipeline uses one-pass decoding for Nemo and Whisper, while Icefall follows a two-pass strategy. In the first pass, we decode the audio using a greedy strategy, and the character error rate (CER) is computed between each pair of ASR outputs. Utterances with zero CER (perfect matches) are excluded from further processing. Non-matching transcriptions proceed to a second decoding pass, where we use an LM. We train the LM using all the unmatched transcripts generated during the first-pass decoding, which is an offline process. The second decoding pass, using Icefall’s fast beam search, refines uncertain outputs. A final CER-based comparison aligns Icefall’s second-pass outputs

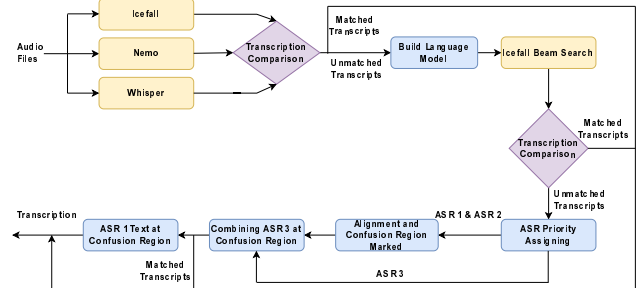


Figure 2: Multi-ASR ensemble pipeline for pseudo-labeling.

with Nemo and Whisper transcriptions.

Based on the CERs of the pairs of mismatched utterances, we rank the three ASR models as ASR-1, ASR-2, and ASR-3, such that ASR model with the lowest average CER is designated as ASR-1. A textual alignment is performed between ASR-1 and ASR-2, and confusion regions are marked, thus highlighting the region of uncertainty between the two transcripts. Transcription from ASR-3 is considered to resolve this uncertainty between ASR-1 and ASR-2. An alignment is performed between ASR-3 and the confusion regions of ASR-1 and ASR-2. If ASR-3 agrees with any of the ASR-1 or ASR-2 transcripts, the uncertainty is resolved, and that region is marked as confident. For the remaining uncertain regions, the transcript from ASR-1 is used to generate the pseudo-labels.

### 2.2. Multi-ASR Textual LLM-based Architecture

In this architecture, we utilize LLMs that have strong capabilities of text understanding, contextual reasoning and text generation. We use Llama 3.2 1B instruct model [29] as the textual LLM to refine uncertain ASR outputs by its learned language and world knowledge, and ability to follow task specific instructions. However, to effectively handle ASR ensemble outputs particularly in the form of textual confusion networks, finetuning is necessary for task specific adaptation.

As shown in Figure 1(b), we perform text alignment between the hypotheses generated from the three ASR models and construct confusion networks in textual form that highlight areas of disagreement and uncertainty. For finetuning the LLM, we prepare the instruction tuning data. The instructions assign a role to the LLM, describes the task and includes the confusion set generated from the text alignment module.

Instead of relying on manually engineered rules, extensively optimized heuristics and a pipeline of post-processing modules, the textual LLM learns how to weigh different hypotheses, correct ASR errors and produce more accurate transcriptions in an end-to-end manner.

### 2.3. Multi-ASR speechLLM-based Architecture

While the confusion networks as described in Section 2.2 provide valuable textual information to LLMs for correcting the transcriptions, the acoustic information present in the original

\*<https://github.com/k2-fsa/icefall>

\*<https://rb.gy/a9y6c3>

\*<https://huggingface.co/openai/whisper-large-v3>

speech is ignored. With speechLLMs, we can create a more holistic system that considers the textual hypotheses along with the underlying acoustic evidence.

In the proposed speechLLM-based framework, we initially derive the hypotheses from three ASR models and create the textual confusion network as described in Section 2.2. We prepare the training data, which are comprised of triplets that include a speech waveform, an instruction with transcription containing confusion sets, and a ground truth response as target output. We create this training data by adding the original audio to the finetuning data created for the textual LLM as follows.

```

audio_path: "sample_xxx.wav"
instruction: "Use the text provided
and correct the mistakes made by ASR.
For better reliability 3 ASRs are used
for transcription and the low
confidence regions are marked as
confusion regions within different
brackets `all|<>|[]` right
two|<too>|[too] bye`"
response: "all right you too bye"

```

Although there exist several multimodal LLM architectures, we select a model combining a speech encoder and a textual LLM, linked by a lightweight adapter for modality alignment [28, 30]. This data-efficient approach requires minimal task-specific supervision since the foundational models are pre-trained on large-scale audio and text data. However, parameter-efficient finetuning is essential for modality alignment, task adaptation, and domain adaptation. The modality alignment through the adapter ensures that the speech encoder output is effectively mapped into the textual LLM’s input representation space, allowing the model to interpret and utilize acoustic information along with ASR hypotheses and instructions. The task adaptation through the textual LLM ensures that it correctly interprets its multimodal input and generates the outputs aligned with the task’s requirements. Additionally, finetuning of speech encoder and textual LLM achieves domain adaptation ensuring the model handles the acoustic conditions and linguistic characteristics relevant to the product setups and use cases. By considering these three objectives in a unified manner and finetuning all components jointly, we achieve performance levels surpassing the textual LLM-based method.

### 3. Experiments

#### 3.1. Datasets used

To effectively compare the presented strategies for generating pseudo-labels, we use diverse datasets that include conversational, telephony and read-speech data in different domains from both public and in-house sources. Table 1 summarizes the details of ASR training, LLM finetuning and test corpora used in our experiments. For the training sets in Table 1, we generate pseudo-labels using different approaches, which are then used to train corresponding ASR models. LLM finetuning data, comprising 27K training samples, is used for finetuning both text and speechLLMs. Since the foundation models have already been trained on vast amounts of audio and text, our focus is on constructing an efficient task-specific dataset that provides high-quality supervision. We employ different test sets to evaluate the WER between ground-truth labels and pseudo-labels generated by the different architectures. The test sets include three in-house call center datasets, DefinedAI data\*, multi-domain data

\*Website: <https://www.defined.ai>

Table 1: Duration and domain information for different training and test sets used in the experiments.

Dataset	Duration (hours)	Domains
Train data		
Librispeech	960	Audiobook
DefinedAI	1410	Banking, Insurance, Retail, Telco
LLM finetuning data		
DefinedAI	289	Banking, Insurance, Retail, Telco
Test data		
Wow	18.21	Autoinsurance, Automotive, Medicare
Wow	18.21	Medical, Home Service, Customer Service
Call center 1	2.00	Medical
Call center 2	7.62	Telco
Call center 3	12.29	Telco
Gigaspeech [31]	36.92	Multi-domain
Librispeech test-clean	5	Audiobook
Librispeech test-other	5	Audiobook
DefinedAI banking	30.27	Banking
DefinedAI insurance	39.70	Insurance
DefinedAI retail	33.02	Retail
DefinedAI telco	42.76	Telco

from Wow AI\* along with open source datasets. Since these datasets are not open-source, we also ensure that they are not seen by any external open-source ASR or language models, thus avoiding data contamination. There is no overlap between the DefinedAI training, LLM finetuning, and test data.

#### 3.2. Experimental setup

We compare the various pseudo-labeling methods by computing transcription accuracy for various test and training sets, and finally training Icefall models with the training sets and their automatic transcripts.

##### 3.2.1. Multi-ASR ensemble pipeline

The Icefall model that is part of the ASR ensemble is trained in-house using the standard Zipformer recipe from the Icefall toolkit with 65 million parameters. Training used DefinedAI conversational data combined with in-house real and simulated call center data, totaling 6,600 hours of training data. The pre-trained Nvidia Parakeet RNNT model, with 1.1 billion parameters, was trained on 64,000 hours of data, including both public and proprietary datasets. The pretrained OpenAI Whisper-large-v3 model, with 1.5 billion parameters, was trained on a 680,000 hours of multilingual audio data, incorporating both weakly labeled and pseudo-labeled data.

We perform decoding in parallel using 3 NVIDIA A10 GPUs in the first pass and single NVIDIA A10 GPU in the second pass. In second pass, the Icefall K2 framework is used to build the language model, and the decoding method used is fast beam search n-best LG, where beam search is used to efficiently generate the top n-best hypotheses guided by an LM to improve the transcription quality. The number of paths  $n$  is set to 200. After second-pass decoding, textual alignment is performed based on Levenstein distance as a heuristic.

##### 3.2.2. Multi-ASR Textual LLM-based Architecture

For postprocessing using a textual LLM, we use Llama 3.2 1B instruct as the LLM. The software ecosystem for this experiment is based primarily on the Huggingface framework. Model development and experiments were conducted using four NVIDIA A10G GPUs with 24GB of memory each. We finetune the textual LLM using the trainer class with QLoRA (4-bit) applied to all its linear layers. The LoRA rank was selected as 32

\*<https://wow-ai.com/data.html>

Table 2: WER (%) of pseudo-labels generated from different individual ASR models as well as with multi-ASR ensemble pipeline, textual LLM postprocessing, and speechLLM-based postprocessing, with respect to the ground-truth transcriptions.

Dataset	Icefall	Nemo	Whisper	Multi-ASR ensemble	Textual LLM	Speech LLM
Train data						
Defined	15.63	14.54	19.36	14.36	11.60	<b>9.30</b>
Librispeech	9.37	<b>1.12</b>	3.30	1.94	2.83	1.95
Test data						
Call center 1	12.04	14.97	19.11	12.92	11.39	<b>9.85</b>
Call center 2	14.47	17.41	21.11	14.76	14.60	<b>13.57</b>
Call center 3	15.32	18.10	23.69	15.52	15.29	<b>14.59</b>
Wow	13.29	17.13	17.70	12.97	<b>12.49</b>	12.57
Gigaspeech	15.89	13.32	16.13	12.99	12.10	<b>11.75</b>
Librispeech test-clean	8.15	<b>1.74</b>	3.30	2.22	2.96	2.26
Librispeech test-other	15.11	<b>3.08</b>	5.66	4.06	5.21	4.50
DefinedAI banking	13.38	16.47	19.88	13.85	11.67	<b>10.45</b>
DefinedAI insurance	13.53	15.70	19.07	13.77	11.18	<b>9.45</b>
DefinedAI retail	15.39	17.45	20.38	15.35	12.97	<b>11.79</b>
DefinedAI telco	13.71	15.58	18.83	13.75	11.25	<b>10.21</b>

with  $\alpha = 128$ ; we used a dropout rate of 0.05. The batch size was set to 4 per GPU, with gradient accumulation of 8 steps, resulting in an effective batch size of 128. The learning rate was managed using a cosine scheduler over 10 to 15 epochs, with a maximum learning rate of  $2 \times 10^{-4}$ . We implement a linear warm-up for the first 20% of the total number of iterations. For optimization, we employ the AdamW (weighted Adam) optimizer. We use its default parameters including  $\beta_1 = 0.9$ ,  $\beta_2 = 0.999$ , and  $\epsilon = 10^{-8}$ , without applying any weight decay. For inference, we use the greedy search with temperature 0. The number of new tokens is limited to 512.

### 3.2.3. Multi-ASR SpeechLLM-based Architecture

For speechLLM postprocessing we use the Qwen2-Audio speech/audio foundation model [32], which has been natively integrated into the HuggingFace ecosystem. The finetuning and inference setup for this model is same as for the textual LLM, as described in Section 3.2.2.

### 3.2.4. ASR model training with pseudo labels

We train two sets of Icefall ASR models using the generated pseudo-labels, one using Librispeech training data, the other using DefinedAI training data, as described in Table 1. For each training set, we train versions based on ground truth transcription as well as pseudo-transcriptions from the multi-ASR ensemble, textual LLM postprocessing, and speechLLM postprocessing pipelines. All the Icefall ASR models mentioned here are trained using the standard Zipformer recipe from the Icefall toolkit and the training setup is same as mentioned in 3.2.1.

## 3.3. Results

We first evaluate the accuracy of the various pseudo-labeling methods employing the three different frameworks as well as each individual ASR model, on all training and test sets listed in Table 1. As a performance metric, we use word error rate (WER) between the ground truth transcription and the machine-generated transcription, as shown in Table 2. We observe that the top-performing ASR varies depending on the datasets, making it challenging to select a single ASR system for generating pseudo-labels. The multi-ASR ensemble achieves a better/comparable WER to the best-performing individual ASR, highlighting the approach’s ability to effectively combine the Icefall, Nemo, and Whisper models. The textual LLM outperforms the ensemble approach and all individual ASRs, except on the Librispeech datasets. This can be attributed to the fact that the Nemo model has seen Librispeech data during training.

Table 3: WER (%) of ASR models trained using ground truth text, a single ASR system output (Icefall), or transcriptions from the proposed ensemble pipelines, for both Librispeech and DefinedAI data.

Training data→	Librispeech training			
Test Data↓	Ground truth	Multi-ASR ensemble	Textual LLM	SpeechLLM
Librispeech test-clean	3.40	3.38	3.87	3.22
Librispeech test-other	8.32	8.44	8.93	8.34
Training data→	DefinedAI training			
DefinedAI banking	9.0	10.3	9.1	8.7
DefinedAI insurance	8.2	9.5	8.4	8.0
DefinedAI retail	11.2	12.3	11.4	10.9
DefinedAI telco	8.8	9.9	8.9	8.6

The improvement is further enhanced with the speechLLM-based corrector. The fusion of audio with the textual confusion network allows the speechLLM to better disambiguate among the generated choices, compared to the textual LLM. For both types of LLM, however, the improvement is greater for the DefinedAI datasets, due to the domain alignment with the finetuning data, demonstrating the importance of domain adaptation. At the same time, we see improvements for other domains, showing the robustness of LLM-based postprocessing.

Table 3 shows the WERs for the Icefall ASR models trained with ground truth transcription and the pseudo-transcriptions generated by different approaches on either Librispeech or DefinedAI data. The evaluation uses matched test sets for each of the training sets. For Librispeech test-clean data, the model based on speechLLM transcriptions achieved 3.22% WER compared to 3.40% using ground truth transcriptions. For test-other, the speechLLM-based result is 8.34% while the one based on just the multi-ASR ensemble is 8.44%. For DefinedAI-trained models, speechLLM-generated transcriptions outperform human-labeled ground truth, highlighting possibly better-than-human (but not easily measurable) transcription accuracy. Additionally, ASR using textual LLM postprocessing surpasses the multi-ASR ensemble approach.\*

In all cases, the results show that the proposed speechLLM-based transcription method generates transcripts close to ground truth, outperforming all other pseudo-labeling approaches.

## 4. Conclusions

We have explored the use of textual and speechLLMs to improve ASR post-processing for pseudo-labeling, moving beyond a traditional cascaded method. The introduction of LLMs enable a unified framework, where a textual LLM refines ASR outputs based on confusion networks and a speechLLM further consults the audio data for improved disambiguation decisions, in a single step. After casting the post-processing as an instruction-following task we have finetuned the base models in a parameter- and data-efficient manner. Thorough evaluations on multiple datasets covering diverse domains and acoustic conditions we demonstrate that textual LLMs significantly outperform a conventional multi-ASR approach and speechLLMs provide further gains by jointly using speech and textual inputs. Our findings highlight the effectiveness of LLM-driven approaches for pseudo-labeling, and hence, for semi-supervised ASR training and adaptation. In short, by unifying pseudo-labeling into a single-stage instruction-following framework, we significantly simplify the process while avoiding error propagation, information loss and disjoint optimization, resulting in improved pseudo-labels for semi-supervised ASR training.

\*We also performed experiments with the LLM finetuning data added to the training of the baseline ASR methods, which reduced WER by only 0.1% to 0.2% absolute. This confirms that the additional data used in LLM learning is *not* the primary source of the observed gains.

## 5. Acknowledgments

We would like to express our sincere gratitude to Aravind Ganapathiraju for his invaluable guidance. We also extend our thanks to Rohit Mishra and Stephen L for their support.

## 6. References

- [1] Z. Huang, G. Keren, Z. Jiang, S. Jain, D. Goss-Grubbs, N. Cheng, F. Abtahi, D. Le, D. Zhang, A. D’Avirro *et al.*, “Text generation with speech synthesis for asr data augmentation,” *arXiv preprint arXiv:2305.16333*, 2023.
- [2] R. Joshi and A. Singh, “A simple baseline for domain adaptation in end to end ASR systems using synthetic data,” in *Proceedings of the Fifth Workshop on e-Commerce and NLP (ECNLP 5)*. Dublin, Ireland: Association for Computational Linguistics, 2022, pp. 244–249. [Online]. Available: <https://aclanthology.org/2022.ecnlp-1.28/>
- [3] A. Fazel, W. Yang, Y. Liu, R. Barra-Chicote, Y. Meng, R. Maas, and J. Droppo, “Synthasr: Unlocking synthetic data for speech recognition,” 2021. [Online]. Available: <https://www.amazon.science/publications/synthasr-unlocking-synthetic-data-for-speech-recognition>
- [4] R. Zevallos, N. Bel, G. Cámbara, M. Farrús, and J. Luque, “Data augmentation for low-resource quechua asr improvement,” in *Proc. Interspeech 2022*, 2022, pp. 3518–3522.
- [5] S. Cornell, J. Daresky, Z. Duan, and S. Watanabe, “Generating data with text-to-speech and large-language models for conversational speech recognition,” in *Proc. Interspeech 2024*, 2024, pp. 6–10.
- [6] Q. Xu, T. Likhomanenko, J. Kahn, A. Hannun, G. Synnaeve, and R. Collobert, “Iterative pseudo-labeling for speech recognition,” in *Proc. Interspeech 2020*, 2020, pp. 1006–1010.
- [7] J. G. 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.
- [8] H. Schwenk and J.-L. Gauvain, “Combining multiple speech recognizers using voting and language model information,” in *Sixth International Conference on Spoken Language Processing*, 2000.
- [9] T. Utsuro, Y. Kodama, T. Watanabe, H. Nishizaki, and S. Nakagawa, “An empirical study on multiple lvcsr model combination by machine learning,” in *Proceedings of HLT-NAACL 2004: Short Papers*, 2004, pp. 13–16.
- [10] M. Matsushita, H. Nishizaki, T. Utsuro, Y. Kodama, and S. Nakagawa, “Evaluating multiple lvcsr model combination in nctir-3 speech-driven web retrieval task,” in *INTERSPEECH*, 2003, pp. 1205–1208.
- [11] J. H. M. Wong, Y. Gaur, R. Zhao, L. Lu, E. Sun, J. Li, and Y. Gong, “Combination of end-to-end and hybrid models for speech recognition,” in *Interspeech*, 2020, pp. 1783–1787.
- [12] P. Swietojanski, A. Ghoshal, and S. Renals, “Revisiting hybrid and gmm-hmm system combination techniques,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 6744–6748.
- [13] K. Hojo, D. Mori, Y. Wakabayashi, K. Ohta, A. Ogawa, and N. Kitaoka, “Combining multiple end-to-end speech recognition models based on density ratio approach,” in *2023 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2023, pp. 2274–2279.
- [14] J. Kolehmainen, Y. Gu, A. Gourav, P. G. Shivakumar, A. Gandhe, A. Rastrow, and I. Bulyko, “Personalization for bert-based discriminative speech recognition rescoring,” in *Interspeech*, 2023, pp. 366–370.
- [15] S. Zhang, M. Lei, and Z. Yan, “Automatic spelling correction with transformer for ctc-based end-to-end speech recognition,” *arXiv preprint arXiv:1904.10045*, 2019.
- [16] O. Hrinchuk, M. Popova, and B. Ginsburg, “Correction of automatic speech recognition with transformer sequence-to-sequence model,” in *Icassp 2020-2020 IEEE international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2020, pp. 7074–7078.
- [17] B. Lin and L. Wang, “Multi-modal asr error correction with joint asr error detection,” in *ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2023, pp. 1–5.
- [18] Y. Li, P. Chen, P. Bell, and C. Lai, “Crossmodal ASR error correction with discrete speech units,” in *2024 IEEE Spoken Language Technology Workshop (SLT)*, 2024, pp. 431–438.
- [19] R. Dong, Y. Li, D. Xu, and Y. Long, “Cross-modal parallel training for improving end-to-end accented speech recognition,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 10 396–10 400.
- [20] M. Nie, M. Yan, C. Gong, and D. Chuxing, “Prompt-based re-ranking language model for asr,” in *INTERSPEECH*, 2022, pp. 3864–3868.
- [21] B. Koilakuntla, P. Rana, P. Ahuja, S. Konjeti, and J. Vepa, “Leveraging large language models for post-transcription correction in contact centers,” in *Proc. Interspeech 2024*, 2024, pp. 2038–2039.
- [22] C.-H. H. Yang, Y. Gu, Y.-C. Liu, S. Ghosh, I. Bulyko, and A. Stolcke, “Generative speech recognition error correction with large language models and task-activating prompting,” in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.
- [23] 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.
- [24] J. Pu, T.-S. Nguyen, and S. Stüker, “Multi-stage large language model correction for speech recognition,” *arXiv preprint arXiv:2310.11532*, 2023.
- [25] C. Chen, Y. Hu, C.-H. H. Yang, S. M. Siniscalchi, P.-Y. Chen, and E.-S. Chng, “Hyporadise: An open baseline for generative speech recognition with large language models,” *Advances in Neural Information Processing Systems*, vol. 36, pp. 31 665–31 688, 2023.
- [26] K. Everson, Y. Gu, H. Yang, P. G. Shivakumar, G.-T. Lin, J. Kolehmainen, I. Bulyko, A. Gandhe, S. Ghosh, W. Hamza *et al.*, “Towards asr robust spoken language understanding through in-context learning with word confusion networks,” in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 856–12 860.
- [27] Y. Hu, C. Chen, C. Qin, Q. Zhu, E. Chng, and R. Li, “Listen again and choose the right answer: A new paradigm for automatic speech recognition with large language models,” in *Findings of the Association for Computational Linguistics: ACL 2024*. Bangkok, Thailand: Association for Computational Linguistics, 2024, pp. 666–679. [Online]. Available: <https://aclanthology.org/2024.findings-acl.37/>
- [28] W. Cui, D. Yu, X. Jiao, Z. Meng, G. Zhang, Q. Wang, Y. Guo, and I. King, “Recent advances in speech language models: A survey,” 2025. [Online]. Available: <https://arxiv.org/abs/2410.03751>
- [29] A. G. *et al.*, “The llama 3 herd of models,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.21783>
- [30] N. Das, S. Dingliwal, S. Ronanki, R. Paturi, Z. Huang, P. Mathur, J. Yuan, D. Bekal, X. Niu, S. M. Jayanthi, X. Li, K. Mundnich, M. Sunkara, S. Srinivasan, K. J. Han, and K. Kirchhoff, “Speechverse: A large-scale generalizable audio language model,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.08295>
- [31] G. Chen, S. Chai, G. Wang, J. Du, W.-Q. Zhang, C. Weng, D. Su, D. Povey, J. Trmal, J. Zhang *et al.*, “Gigaspeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio,” in *Proc. Interspeech 2021*, 2021, pp. 3670–3674.
- [32] Y. C. *et al.*, “Qwen2-audio technical report,” 2024. [Online]. Available: <https://arxiv.org/abs/2407.10759>