ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10888566

LLM based Text Generation for Improved Low-resource Speech Recognition Models

Tohru Nagano, Gakuto Kurata,*
Samuel Thomas, Hong-Kwang J. Kuo, Daniel Bolanos, Hyun Jung and George Saon.†

*IBM Research, Tokyo, Japan
†IBM Research, Yorktown Heights, USA
tohru3@jp.ibm.com

***Abstract*—Limited transcribed spoken style data is a critical bottleneck in building automatic speech recognition (ASR) systems for low-resource languages. Prompting a large language model (LLM) to paraphrase input text can generate novel text data that is constrained to be semantically similar to the source data. We leverage this capability of LLMs to improve the performance of low-resource ASR systems by increasing the limited text training data while keeping the same spoken style. Since word sequences in the training data are now more diverse and the vocabulary of the ASR model is also expanded, this approach allows for building general purpose ASR without prior knowledge of various domains in the low-resource language. In our experiments with Brazilian Portuguese as a low-resource language, paraphrased data enhanced the n-gram language model (LM) used to build the weighted finite state transducer (WFST) for decoding with a Conformer-CTC speech recognition model, resulting in improvement of word error rate (WER) by 15.6% over the baseline model. Synthesizing the paraphrased text into speech and using it to fine-tune the acoustic model (AM) component helped to further improve the WER by 2.9%, achieving a combined improvement of 18.5%. We also demonstrate the usefulness of our proposed approach for high-resource languages like English.**

***Index Terms*—LLM, Data augmentation, N-gram, TTS, Speech splicing**

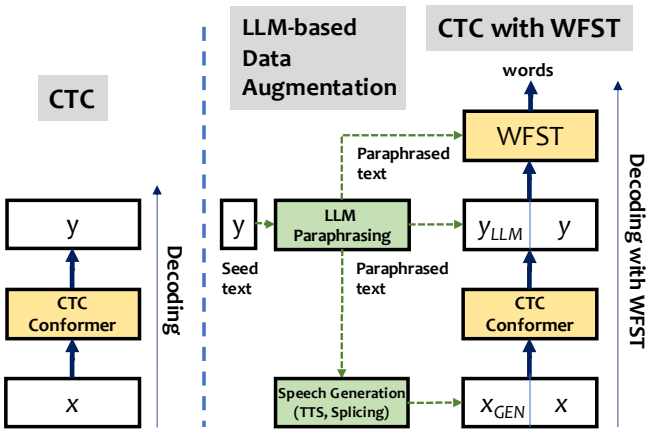

Fig. 1. Conventional CTC model (left), CTC model using WFST decoder (right). Proposed LLM and speech generation enhance text and audio training data.

I. Introduction

Neural network based large language models (LLMs) trained on vast amounts of text data have demonstrated high performance in tasks such as question answering, machine translation, sentence correction, sentence creation, and document summarization. On the basis of their high quality text generation ability, LLMs are used to augment training data for various natural language processing tasks. Data augmentation techniques with LLMs such as ChatGPT and GPT-4 [1] exhibits mostly consistent superior performance on cross-lingual commonsense reasoning tasks that are challenging for data synthesis [2]. LLMs have not only been shown to serve as excellent crowd-sourced annotators on multiple tasks, including user input and keyword relevance assessment [3], but synthetic data generated by LLMs has also been demonstrated to significantly enhance open intent detection capabilities [4]. In terms of speech recognition modeling, recent end-to-end (E2E) ASR systems have a monolithic structure that do not require phonetic representation or other linguistic knowledge, making it easier to construct speech recognition systems for a wide variety of languages [5]. These systems still require paired text and audio data for training and their performance is poor for tasks such as recognizing long-tail rare words. The integration of auxiliary neural language models, such as shallow fusion [6], [7] and density ratio fusion [8], into speech recognition models has been studied to compensate for the shortcomings of these end-to-end models and improves performance in rare-word recognition and domain adaptation. Unlike LLMs, these auxiliary models are however very specialized language models that are biased to the context in which they have been constructed. There have hence been attempts to use generalized LLMs for speech recognition. In addition to attempts in error correction [9], re-ranking [10], and the use of auxiliary networks [11], a system that directly connects a large speech encoder and LLM [12] has shown excellent performance. However, the computational cost of LLMs is very high, and needs to be improved for real-time decoding. To construct ASR models similar to LLMs which are trained on large amounts of data, there have also been attempts to gather resources for various speech recognition tasks [13] [14]; however, the amount of speech data that can be collected is much smaller than the amount of available text data, and a significant amount of effort is required to pair speech data with transcriptions. These constraints result in the lack of training data in many low resource languages.

In this paper, we attempt to leverage the high quality text generation abilities of LLMs described earlier to address the lack of training data resources for ASR models in low-resource languages. We hypothesize a practical solution that uses the knowledge encoded in LLMs to generate novel paraphrased sentences corresponding to the limited training data in these languages. With this approach we improve the LM and AM components by improving and expanding the data used to train these modules in multiple ways rather than using the LLM directly during decoding as an additional part of an ASR system. The LM is improved by using the generated paraphrased sentences to construct a better n-gram language model within an WFST framework, while the AM is enhanced by adapting the underlying Conformer-CTC using corresponding synthesized speech generated using text-to-speech (TTS) or speech splicing.

The main contributions of this research are: (1) by combining off-

the-shelf LLMs with existing speech synthesis mechanisms, we show it is possible to improve speech recognition accuracy for low-resource languages without collecting new data, (2) we present various insights on the usefulness of synthetic text generated by LLMs in improving speech recognition performance. These include:

- Paraphrasing by LLM can generate new text data that is constrained to be semantically similar to the input. This approach does not require any specific knowledge of various domains in the target language. General purpose speech recognition is improved, as shown by consistent reduction in error rates across test sets in different domains.
- The LLM generated data by itself is not as as good as the original text data although we apply various data selection techniques to improve the quality of selected data. The synthetic data is however complementary to the original text data and helps in combination with the original data.
- Increasing the amount of generated text increases the ASR model's vocabulary without detrimental effects and improves the accuracy of the ASR model.
- The generated text can be used to produce novel acoustic data via TTS synthesis / audio splicing that can help improve the acoustic model and hence the ASR model further.

We demonstrate the usefulness of our approach in settings suitable for real-time ASR applications and deployment using a model that has a weighted finite state transducer based on an n-gram LM and a Conformer-CTC model adapted using synthesized audio as its AM component. We carefully study the performance of the system with varying amounts of synthetic data and demonstrate its usefulness for both low and high-resource languages.

II. Augmentation with LLM

A. Paraphrasing with Large Language Models

The training data for speech recognition consists of pairs of input feature sequences x and output character sequences y. We perform paraphrasing of this text data y using an LLM to create a natural paraphrase y_{LLM}. The LLM takes as input a fixed text prompt *"Please rewrite the following sentences in English without changing their meaning:"* followed by a single sentence from the original training text data. Random sampling decoding with the LLM is used to obtain different paraphrased sentences by changing the random sampling seed. Table I shows examples of the paraphrases obtained. We use the granite-20b-multilang [15] model as the LLM, which supports multiple languages. By combining paraphrasing instructions and example sentences, sentences with different grammar and word choices but with constrained meaning can be generated. After some trial and error, we created a prompt that produced effective results. Since the LLM was not trained solely for paraphrasing, it may output unexpected texts, which we exclude from the results. More details of the filtering procedure are in section III-B.

B. Conformer CTC with WFST-based Decoding

The text expanded by the LLM is used to strengthen the external language model used during speech recognition decoding and to train the Conformer-CTC. Connectionist Temporal Classification (CTC) [16] is a typical non-autoregressive model that learns the alignment between input speech and output tokens on a frame-by-frame basis and predicts the target sequence. In our experiments, we use the Conformer encoder [17] and an intermediate CTC [18], which adds the loss of the intermediate layer to the CTC loss. This approach has the highest performance among non-autoregressive models [19].

In addition, CTC character-based WFST Viterbi decoding [20] is used, which enables vocabulary and language models to be efficiently incorporated into CTC decoding, thereby improving recognition accuracy even with LM enhancement alone (Fig.1 right). While greedy decoding selects the 1-best sequence of symbols, the probability distribution of character occurrence for each frame is input to the WFST decoder. After lexical probabilities are added, it is converted into the most likely word sequence. This behavior is similar to DNN-HMM speech recognition [21], but in CTC character-based WFST decoding, the character string output from CTC is used as the input [22] [23].

III. Experiment

A. Data and Model

The basic experimental setup involves 460 hours of telephone bandwidth, unpublished Brazilian Portuguese speech recognition training data, with manual transcription amounting to 3.6M word tokens. The test data consists of a total of 17.1 hours of audio across nine domains (bank, short-form inquiries, payment, dialog, insurance, fintech, ivr, ted). The data contains words specific to each domain, and the bank, dialog, and ted domains have relatively long utterances.

A Conformer-CTC model is trained on 2300 hours after 0.1x speed/tempo augmentation is applied to the original 460 hours of audio. We use a learning rate of 1e-04 and train the model for 20 epochs. The Conformer encoder consists of 10 Conformer blocks, has a total of 175M parameters, and outputs 41 types of Portuguese characters. The input audio is converted into 40-dimensional speaker-independent log-mel features every 10 msec; every two frames are stacked, delta and double delta coefficients are calculated, and the features are finally converted into 240-dimensional features every 20 msec. During fine-tuning with the generated speech, the Conformer-CTC model is adapted for 4 epochs.

B. Text Generation

We use the LLM to acquire paraphrases of sentences in the original training data. The temperature for inference is set to 1.0, and the maximum additional token length is set to 100. The prompt provided to the LLM is *"Reformular o texto com o mesmo sentido em português:"* in Portuguese (*Rephrase the text with the same meaning in Portuguese:*). We add each training sentence to this prompt in order to acquire paraphrased sentences using the LLM.

In the next step, inappropriate sentences are filtered out from the paraphrased sentences. Since the LLM output sometimes includes sentences that are not actual paraphrases, such as "I understand. The following is a paraphrase," only sentences most similar to the input sentence are used. In practice, 95.0% of the LLM output consisted of one sentence, while 5% consisted of two or more sentences (with an average of 1.1 sentences). The sentence with the highest similarity was selected using bert-base-multilingual-uncased [24]. We used fastText [25] to delete sentences in languages other than the input language. Using a single random seed, the resulting set of paraphrased sentences has on average about 4.5M words, which was about 25% larger than the 3.6M word input.

C. Speech Generation

The audio data for training was extended by generating speech data with the paraphrased text using TTS and speech splicing. To create paired text and audio data (x_{GEN}, y_{LLM}) for training, we utilized the three methods described in the following sections.

1) TTS and Voice Conversion: We used a TTS model based on LPCNet [26] trained with 15 hours of speech. As the model is single-speaker, we applied voice conversion OpenVoice V2 [27] to make the voice quality closer to that of the training data and to increase variability in the training data. In one experiment, we compared

TABLE I

AN EXAMPLE OF PARAPHRASING IN ENGLISH USING AN LLM. PROMPT IS THE INPUT FOR THE LLM, AND CONSISTS OF A FIXED *instruction* AND **ORIGINAL SENTENCE FOR PARAPHRASING**.

Prompt	⟨user⟩ *Please rewrite the following sentences in English without changing their meaning:* **well it seems like my account has been locked for some reason and i don't know why happened um yes suddenly yesterday** ⟨assistant⟩
Output 1	It appears that my account has been locked, and I'm unsure of the reasons behind this action. Yesterday, unexpectedly, my account was locked.
2	My account has been locked, and I'm puzzled as to why this occurred. Suddenly, on the previous day, I found that my account was inaccessible.
3	I think my account has been locked, but I don't know the exact reason. Yesterday, all of a sudden, I realized my account was disabled.
4	My account has been locked, and I'm confused about what triggered this action. Suddenly, my account became inaccessible, and I can't figure out why.
5	It seems that my account has been locked, and I'm not aware of the cause. Yesterday, without any prior notice, I noticed that my account was locked.

TABLE II

PORTUGUESE ASR RESULTS. (LEFT) AMOUNT OF TEXT DATA GENERATED BY THE LLM, ORDER OF THE N-GRAM MODEL CREATED FROM GENERATED TEXT, SIZE OF THE WFST GRAPH (MBYTE), THE VOCABULARY SIZE, AND AMOUNT OF SPEECH DATA AND SPEECH GENERATION METHOD. (RIGHT) WER AND AVERAGE FOR THE NINE TEST SETS (◇ - BASELINE, ⋆ - BEST RESULTS).

Generated Data					**WER**									
Text	N-gram	WFST	Vocab	Speech	*bank*	*shortform*	*inquiries*	*payment*	*dialog*	*insurance*	*fintech*	*ivr*	*ted*	*Average*
	(Greedy decoding)				29.4	10.3	21.5	10.9	30.4	25.9	15.1	7.6	24.8	19.5
	2		35K		26.3	9.6	28.8	9.9	14.0	24.3	12.9	9.0	24.3	17.7
	4	106M	35K		26.3	9.8	27.7	10.2	13.7	22.3	13.1	8.6	24.0	17.3 ◇
x5	4	224M	76K		25.5	8.3	22.3	8.6	13.6	21.2	11.9	8.3	21.2	15.7
x10	4	320M	93K		25.5	7.8	20.4	8.4	13.4	21.4	11.2	8.2	20.7	15.2
x32	4	652M	132K		25.5	7.6	17.6	8.1	13.3	21.1	11.0	7.4	19.6	14.6
x10	4	320M	93K	x1 TTS	25.4	7.7	20.1	8.0	13.0	20.4	11.0	7.0	20.0	14.7
x10	4	320M	93K	x1 TTS/VC1	25.7	7.5	20.5	8.0	13.1	19.6	10.8	7.2	19.7	14.6
x10	4	320M	93K	x1 TTS/VC2	25.7	7.5	20.9	8.0	13.0	20.1	10.7	7.0	19.7	14.7
x32	4	652M	132K	x1 TTS	25.2	7.6	17.2	7.8	13.0	19.7	10.8	6.7	19.2	14.1 ⋆
x32	4	652M	132K	x1 Splicing.G	25.6	7.5	16.9	8.1	13.2	20.4	10.8	7.5	19.9	14.4
x32	4	652M	132K	x1 Splicing.P	25.4	7.5	16.8	8.3	13.3	19.1	10.8	7.1	19.6	14.2
x32	4	652M	132K	x10 Splicing.P	25.2	7.3	16.5	8.3	13.1	19.2	10.7	7.1	19.5	14.2
x48	4	846M	159K	x1 TTS	25.4	7.5	16.6	7.8	13.0	20.2	10.9	6.6	19.1	14.1 ⋆

the model that uses a single speaker vector randomly selected from the training data (TTS/VC1) with the one that uses a new speech vector created by averaging two randomly selected speaker vectors (TTS/VC2).

2) Speech Splicing with CNN Model : Splicing.P: Speech splicing [28] is a method that, like early TTS, takes text as input and generates speech corresponding to the input sentence by concatenating speech fragments corresponding to each word and phoneme. We investigated two different speech splicing methods that use different alignment techniques. The pool of speech fragments is created from the training data, and one fragment that matches the word or phoneme is used at random to generate speech data.

Splicing.P uses an existing DNN-HMM-based speech recognition model and uses phone-based forced alignment for each frame to extract phoneme fragments. For dictionary words, a word-based pool is used, and for unknown words, a transformer G2P created from a Portuguese dictionary is used to convert words into phoneme sequences. Speech fragments corresponding to these phonemes are retrieved from the pool and concatenated.

3) Speech splicing with CTC Forced Alignment : Splicing.G: Splicing.G is an alignment method that uses the CTC model [5] and performs character-based alignment without phonetic symbols. However, the output of the CTC model is peaky, and in many cases, only one frame per character will show a CTC peak. To obtain the correct speech segments for each word and character, we use several heuristics. The alignment of characters within a word is based on the center of the peak and the peak itself. If the distance between the last peak of the previous word and the first peak of the next word is within four frames, it is assumed that there is no pause between the words, and the middle of the two peaks is used as the boundary. If the distance is more than four frames, it is assumed that there is a short pause, and the boundary is set four frames away from each peak. Words that match the vocabulary of the pool are spliced on a word-by-word basis, and for unknown words, the audio is synthesized from the character-by-character pool.

D. Experiment

The experimental results are shown in Table II. The baseline results are shown in the first block. The result of greedy decoding for CTC without WFST is shown on the first line. Although the 2-gram model shows decent results for several test sets, the average WER is better for the 4-gram model. Therefore, we will use the 4-gram language model for all subsequent experiments.

Next, examining the results of using paraphrases with the LLM, accuracy improves as the amount of generated data increases, i.e., x5, x10, and x32. We interpolate the language model created from the original transcription $\mathcal{L}(y)$ with that created from the paraphrases $\mathcal{L}(y_{\text{LLM}})$ to create a language model for decoding: $\mathcal{L}_{interp} =$

$\lambda\mathcal{L}(y) + (1-\lambda)\mathcal{L}(y_{\text{LLM}})$. The λ with the lowest average WER was used (search in increments of 0.1 and not tuned for each test data). If we focus on the number of vocabulary words, the baseline vocabulary consists of 35k words. With data generation increased to x5, the vocabulary approximately doubles to 76k words; with x10, it triples to 93k words; and with x32, it quadruples to 132k words. Recognition accuracy improves the most in the x32 case, reducing the WER from 17.3% to 14.6%, which is a 15.6% improvement.

In the third block of Table II, we show the results of fine-tuning Conformer-CTC using generated speech while using a fixed x10 LM. The amount of TTS speech data used was 410 hours, which was slightly less than the amount of the original speech data. TTS/VC1 and TTS/VC2 apply voice conversion using speaker vectors extracted from the training data. The voice conversion does not change the speed or style of the speech, so the amount of speech remains the same. When fine-tuning with TTS speech, the bottom five layers of the 10-layer Conformer block were frozen. Comparing TTS, TTS/VC1, and TTS/VC2, the result for TTS/VC1 was slightly better at 14.6%, representing a 3.9% improvement in the WER compared with the 15.2% without acoustic fine-tuning.

In the final block, we compared three different speech generation methods while using a fixed x32 LM. The result for the more established Splicing.P was 14.2%, which was better than the 14.4% for Splicing.G, and almost the same accuracy as the 14.1% for TTS. Finally, even when the amount of data for Splicing.P was increased tenfold by changing the random seed for selecting speech synthesis fragments, the accuracy did not change. When data generation using the LLM was increased to x48, the vocabulary increased, but the WER did not change compared with the x32 case.

E. Experiment with English Data

We also conducted experiments in a different language, English, where the Conformer-CTC base model was a strong one built on 20k hours of publicly available and in-house telephone bandwidth training data. We investigated whether LLM augmentation was still effective at expanding the LM corpus.

TABLE III
RESULTS FOR ENGLISH. WER OF SWITCHBOARD, CALLHOME AND TWO CALL CENTERS (◇ - BASELINE, ⋆ - BEST RESULTS).

Generated Data				WER			
				Hub5'00			
Text	N-gram	WFST	Vocab	SWB	CH	cc1	cc2
	4	134M	29K	8.6	13.0	18.1	21.9 ◇
x10	4	254M	72K	8.4	12.9	17.5	21.3
x48	4	581M	131K	8.3	12.7	17.3	20.9 ⋆

We created a baseline LM from the Switchboard 2.7M word transcription corpus. We then input a sentence consisting of a prompt, as shown in Table I and each sentence from the corpus into the LLM, generating 10 and 48 paraphrases. The other procedures were the same as those for the Portuguese experiment. For testing, we used Switchboard and CallHome test sets, which are similar to the training text, as well as two unrelated proprietary call center test sets to see if the data augmentation would help both types of test sets: matched and mismatched compared with the original LM text. The two internal call center test sets cc1 and cc2 have different speaking styles and contain approximately 4.6 and 3.0 hours of audio respectively. Examining the results for Text x10 and x48 in Table III, we can see that accuracy has improved not only for Switchboard and CallHome but also cc1 and cc2, confirming that LLM text augmentation improves English speech recognition, across a variety of domains.

IV. DISCUSSION

According to the experimental results, LLM augmentation serves two roles: expanding the context variety and providing new vocabulary. To investigate the characteristics of the acquired paraphrases, we examined the Portuguese test set perplexity (PPL) up to x48 (Fig. 2) and found that the PPL decreased as the amount of data increased. However, the PPL of the language model generated from paraphrased data alone (transparent boxes in Fig. 2) did not decrease beyond the PPL of the transcribed corpus. It is possible that the style of the sentences generated by the LLM did not match the speech, indicating that there is room for adjusting the prompts. On the other hand, the PPL of the interpolated model (hatched boxes) is better than that of the transcribed corpus (dotted line). This means that paraphrased expressions are not as good as the original sentences on their own, but that the generated data complements the original text data and is useful for combining with the original data.

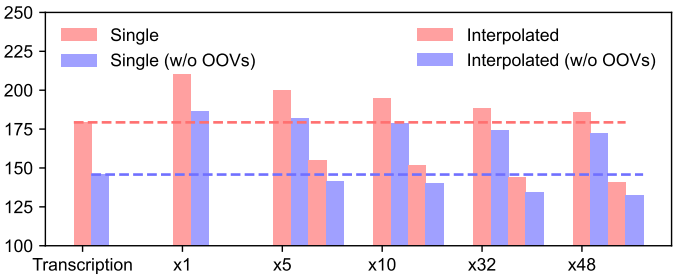

Fig. 2. Test-set perplexity of the language model for the transcription (leftmost) and the language model for the paraphrase.

For speech generation we considered TTS and two splicing methods. Since TTS requires a model that corresponds to the target language, and Splicing.P requires a phone-level alignment tool, a combination of a DNN-HMM-based speech recognition model and a language model is necessary. However, Splicing.G natively supports multiple languages, making it preferable for deployment across multiple languages. Since the accuracy of Splicing.G is insufficient, it is necessary to improve accuracy by using methods such as the CTC alignment model [29], which aims for non-peaky output.

For languages with a large amount of training data, it may be possible to demonstrate the effectiveness of this method in domain adaptation scenarios rather than in improving the general purpose speech recognition model. We also investigated the computational cost of the WFST decoding at runtime. In our experiment, the overall decoding real-time factor (RTF) was 0.24, with the RTF used for the WFST beam search being only around 0.03.

V. CONCLUSION

In this paper we have leveraged LLMs for ASR in low-resource languages by generating novel text training samples as paraphrases of the original data. By extending the language model and additionally adapting the Conformer-CTC AM with synthetic speech derived from the generated data, this method reduced the WER by 18.5%. We further demonstrate that this method can also improve performance on English and can be used for domain adaptation in large resource languages. With regard to text generation for ASR, there is potential to address new recognition tasks by using various kinds prompts for text generation and augmentation. These include creating a spoken style corpora of paraphrases with different dialects and emotions, and generating novel data from responses to questions related to particular domains or tasks. The improvement from using LLMs is significant, and more accuracy improvements are expected with further research.

REFERENCES

[1] OpenAI, "Gpt-4 technical report," arxiv.org/abs/2303.08774, 2024.

[2] Chenxi Whitehouse, Monojit Choudhury, and Alham Fikri Aji, "LLM-powered data augmentation for enhanced cross-lingual performance," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Houda Bouamor, Juan Pino, and Kalika Bali, Eds., Singapore, Dec. 2023, pp. 671–686, Association for Computational Linguistics.

[3] Xingwei He, Zhenghao Lin, Yeyun Gong, A-Long Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, and Weizhu Chen, "AnnoLLM: Making large language models to be better crowdsourced annotators," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 6: Industry Track)*, Yi Yang, Aida Davani, Avi Sil, and Anoop Kumar, Eds., Mexico City, Mexico, June 2024, pp. 165–190, Association for Computational Linguistics.

[4] Yihao Fang, Xianzhi Li, Stephen Thomas, and Xiaodan Zhu, "ChatGPT as data augmentation for compositional generalization: A case study in open intent detection," in *Proceedings of the Fifth Workshop on Financial Technology and Natural Language Processing and the Second Multimodal AI For Financial Forecasting*, Chung-Chi Chen, Hiroya Takamura, Puneet Mathur, Remit Sawhney, Hen-Hsen Huang, and Hsin-Hsi Chen, Eds., Macao, 20 Aug. 2023, pp. 13–33, -.

[5] Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, "Scaling speech technology to 1,000+ languages," arxiv.org/abs/2305.13516, 2023.

[6] Anjuli Kannan, Yonghui Wu, Patrick Nguyen, Tara N. Sainath, ZhiJeng Chen, and Rohit Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1–5828.

[7] Ke Hu, Tara N. Sainath, Bo Li, Nan Du, Yanping Huang, Andrew M. Dai, Yu Zhang, Rodrigo Cabrera, Zhifeng Chen, and Trevor Strohman, "Massively multilingual shallow fusion with large language models," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[8] Erik McDermott, Hasim Sak, and Ehsan Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 434–441.

[9] Srijith Radhakrishnan, Chao-Han Yang, Sumeer Khan, Rohit Kumar, Narsis Kiani, David Gomez-Cabrero, and Jesper Tegnér, "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*, Houda Bouamor, Juan Pino, and Kalika Bali, Eds., Singapore, Dec. 2023, pp. 10007–10016, Association for Computational Linguistics.

[10] Takuma Udagawa, Masayuki Suzuki, Gakuto Kurata, Nobuyasu Itoh, and George Saon, "Effect and analysis of large-scale language model rescoring on competitive asr systems," in *Interspeech 2022*, 2022, pp. 3919–3923.

[11] Yosuke Higuchi, Tetsuji Ogawa, Tetsunori Kobayashi, and Shinji Watanabe, "Bectra: Transducer-based end-to-end asr with bert-enhanced encoder," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[12] Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, and Xie Chen, "An embarrassingly simple approach for llm with strong asr capacity," arxiv.org/abs/2402.08846, 2024.

[13] Rosana Ardila, Megan Branson, Kelly Davis, Michael Henretty, Michael Kohler, Josh Meyer, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber, "Common voice: A massively-multilingual speech corpus," arxiv.org/abs/1912.06670, 2020.

[14] Xinjian Li, Shinnosuke Takamichi, Takaaki Saeki, William Chen, Sayaka Shiota, and Shinji Watanabe, "Yodas: Youtube-oriented dataset for audio and speech," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[15] Ibrahim Abdelaziz, Kinjal Basu, Mayank Agarwal, Sadhana Kumaravel, Matthew Stallone, Rameswar Panda, Yara Rizk, GP Bhargav, Maxwell Crouse, Chulaka Gunasekara, Shajith Ikbal, Sachin Joshi, Hima Karanam, Vineet Kumar, Asim Munawar, Sumit Neelam, Dinesh Raghu, Udit Sharma, Adriana Meza Soria, Dheeraj Sreedhar, Praveen Venkateswaran, Merve Unuvar, David Cox, Salim Roukos, Luis Lastras, and Pavan Kapanipathi, "Granite-function calling model: Introducing function calling abilities via multi-task learning of granular tasks," arxiv.org/abs/2407.00121, 2024.

[16] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks," in *Proceedings of the 23rd International Conference on Machine Learning*, New York, NY, USA, 2006, ICML '06, pp. 369–376, ACM.

[17] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang, "Conformer: Convolution-augmented Transformer for Speech Recognition," in *Proc. Interspeech 2020*, 2020, pp. 5036–5040.

[18] Jumon Nozaki and Tatsuya Komatsu, "Relaxing the Conditional Independence Assumption of CTC-Based ASR by Conditioning on Intermediate Predictions," in *Proc. Interspeech 2021*, 2021, pp. 3735–3739.

[19] Yosuke Higuchi, Nanxin Chen, Yuya Fujita, Hirofumi Inaguma, Tatsuya Komatsu, Jaesong Lee, Jumon Nozaki, Tianzi Wang, and Shinji Watanabe, "A comparative study on non-autoregressive modelings for speech-to-text generation," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2021, pp. 47–54.

[20] George Saon, Daniel Povey, and Geoffrey Zweig, "Anatomy of an extremely fast lvcsr decoder," in *Proceedings of Interspeech*, 2005.

[21] George Saon, Hong-Kwang J. Kuo, Steven Rennie, and Michael Picheny, "The ibm 2015 english conversational telephone speech recognition system," in *Interspeech 2015*, 2015, pp. 3140–3144.

[22] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," arxiv.org/abs/1507.08240, 2015.

[23] Haşim Sak, Andrew Senior, Kanishka Rao, Ozan İrsoy, Alex Graves, Françoise Beaufays, and Johan Schalkwyk, "Learning acoustic frame labeling for speech recognition with recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4280–4284.

[24] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," arxiv.org/abs/1810.04805, 2019.

[25] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov, "Bag of tricks for efficient text classification," arxiv.org/abs/1607.01759, 2016.

[26] Zvi Kons, Slava Shechtman, Alex Sorin, Carmel Rabinovitz, and Ron Hoory, "High Quality, Lightweight and Adaptable TTS Using LPCNet," in *Proc. Interspeech 2019*, 2019, pp. 176–180.

[27] Zengyi Qin, Wenliang Zhao, Xumin Yu, and Xin Sun, "Openvoice: Versatile instant voice cloning," arxiv.org/abs/2312.01479, 2024.

[28] Rui Zhao, Jian Xue, Jinyu Li, Wenning Wei, Lei He, and Yifan Gong, "On addressing practical challenges for rnn-transducer," arxiv.org/abs/2105.00858, 2021.

[29] Wei Wang, Xun Gong, Hang Shao, Dongning Yang, and Yanmin Qian, "Text Only Domain Adaptation with Phoneme Guided Data Splicing for End-to-End Speech Recognition," in *Proc. INTERSPEECH 2023*, 2023, pp. 3347–3351.