LLM as decoder: Investigating Lattice-based Speech Recognition Hypotheses Rescoring Using LLM

Sheng Li*, Yuka Ko† and Akinori Ito‡
* National Institute of Information and Communications Technology (NICT), Kyoto, Japan
† Nara Institute of Science and Technology (NAIST), Nara, Japan
‡ Tohoku University, Sendai, Japan
E-mail: sheng.li@nict.go.jp, ko.yuka.kp2@is.naist.jp, akinori.ito.a2@tohoku.ac.jp

***Abstract*—With the strong representational power of large language models (LLMs), generative error correction (GER) for automatic speech recognition (ASR) aims to provide semantic and phonetic refinements to address ASR errors. However, the previous LLM GER method used n-best lists, which are simpler, more efficient, and more vulnerable to the issue of hypothesis space collapse. Hypothesis space collapse is a phenomenon in speech recognition where the range of possible interpretations becomes too narrow. This paper proposes replacing n-best hypotheses with lattice, a more flexible ASR output format, to improve the LLM GER and reduce the likelihood of hypothesis space collapse. Experiments on CSJ corpus show that compared with using n-best hypotheses, using lattice can improve the performance of Japanese speech recognition.**

I. Introduction

Automatic speech recognition (ASR) is one of the most natural human-machine interfaces for various spoken language applications. However, various factors, such as background noise, accents, speech clarity, and the quality of the recording equipment, always influence ASR results. These influences can lead to inaccuracies in transcriptions, affecting the overall effectiveness and reliability of speech recognition systems.

The automatic generative error correction (GER) task uses pre-trained models to automatically revise and enhance written content, including documents, articles, and translations. This process involves correcting grammar, syntax, style, and coherence errors and tailoring language to specific requirements or preferences. GER can effectively refine ASR outputs by correcting transcription errors. This improves the quality and usability of transcribed text for applications like captioning, transcription services, and voice-controlled systems.

Pretrained large neural network-based language models (LMs) have been applied to GER tasks linked to ASR in recent years. To lower the substitution error in Mandarin speech recognition, Zhang et al. [1] suggested a spelling corrector based on the transformer [2]. The effectiveness of BERT [3] in identifying spelling errors was enhanced by [4], using the soft-masking technique as a link between the error detector and corrector. Using the BERT distilling knowledge, Futami et al. [5] created soft labels for ASR training. Additionally, certain studies have been conducted [6], [7] to enhance ASR rescoring by BERT. Furthermore, BERT has also been effectively used in multi-modal research in voice-language pretraining [8], [9], [10] or vision-language pretraining [11], [12], [13], [14].

More recently, Chen et al. [15] proposes combining large language models (LLMs) into a speech recognition system. However, n-best lists are more prone to hypothesis space collapse due to their fixed and limited number of alternatives. In speech recognition, "hypotheses space collapse" refers to a phenomenon where the range of possible interpretations of an input (the hypothesis space) becomes too narrow or restricted. This can lead to several issues: The system might consider only a limited set of possible transcriptions, ignoring other possible alternatives. This can reduce the recognition system's accuracy and robustness. As for the n-best list used in [15], once the list is generated, any transcription not in the top-N hypotheses is ignored, which can significantly narrow the hypothesis space. While n-best lists are simpler and more efficient, they are more vulnerable to hypotheses space collapse due to their limited representation ability.

Compared to the n-best list, lattices are less likely to hypothesize space collapse because they maintain a broader and more flexible representation of possible transcriptions. The lattice structure allows for more diverse hypotheses, reducing the likelihood of collapsing the hypothesis space. This paper addresses the issue of n-best hypotheses and enhances the LLM GER. As depicted in Figure 1, we replace n-best hypotheses with lattice, a more compact ASR output format.

II. Related Work

A. LLMs for ASR task

Chen et al. [15] have suggested integrating LLMs into a voice recognition system more recently. In this research, LLM is used for second-pass rescoring in the output transcriptions produced by the ASR system (n-best decoding hypotheses) for GER error correction. This work presents LoRA [16], which allows for efficient learning of the n-best to transcription mapping without affecting the pre-trained parameters of the LLM. LoRA works by inserting a neural module with a small number of extra trainable parameters to approximate the full parameter updates, thereby avoiding the need to tune the entire set of parameters of a pre-trained model.

By adding trainable low-rank decomposition matrices to the current layers of LLMs, this technique allows the model to adjust to fresh input while maintaining the original LLMs' fixed structure to preserve prior knowledge. In particular, LoRA injects low-rank decomposition matrices to reparameterize

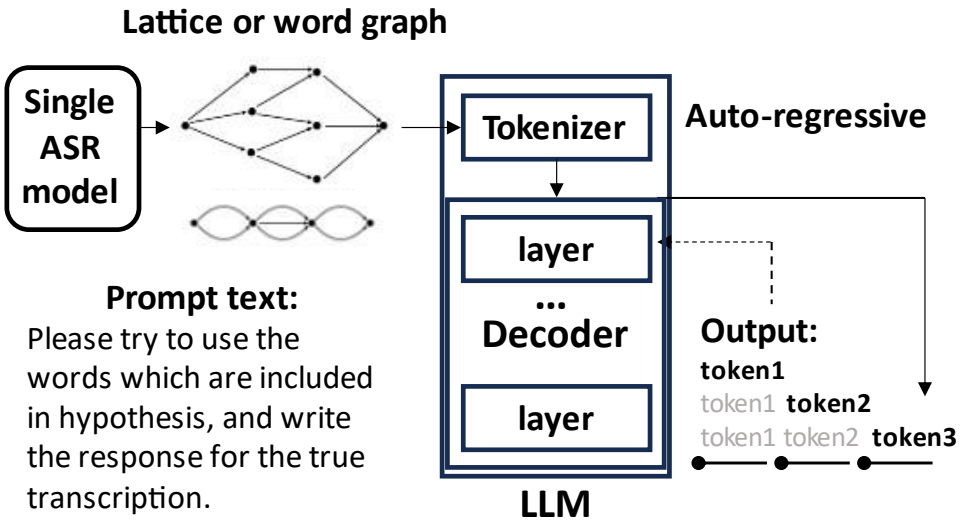

Fig. 1: The flowchart of the proposed LLM GER method using lattice output from ASR system instead of n-best hypotheses.

each model layer, which is described as a matrix multiplication. The representations produced by the LLM are, therefore, not warped by task-specific tailoring. Simultaneously, the adapter module gains the capacity to forecast the actual transcription based on the n-best estimates. Thanks to effective training, the method makes use of a large-scale language model that should be able to comprehend the task description and identify correlation in the n-best list.

III. Improving LLM GER using Lattice for ASR

As mentioned in previous work [15], a significant challenge arises when previous LLM GER fails to yield improvements at relatively low Character Error Rates (CER). Our proposed method addresses this issue by enhancing LLM GER. As depicted in Figure 1, we replace n-best hypotheses with lattice, a more compact ASR output format (as shown in Figure **??**). In speech recognition, a lattice is a graph representation of multiple potential word sequences generated during decoding. It captures various hypotheses of the spoken utterance, allowing for more accurate recognition by considering alternative paths and word sequences. Lattices are useful for handling ambiguity and improving recognition accuracy, especially in noisy or complex speech environments.

LLMs obtain the entire hypothesis space, which is compactly stored in lattice format from the speech recognition system, and then derive the 1-best hypothesis. In a sense, this accomplishes the traditional function of a speech recognition decoder. Therefore, we can say that the LLM is the decoder.

There are two lattice formats in speech recognition and machine translation areas as follows:

- Kaldi's lattice format is a representation of possible transcriptions from speech recognition, structured as directed acyclic graphs. Each node signifies a point in time, and each arc between nodes represents a hypothesized word with associated acoustic scores and transition probabilities. This format allows for multiple hypotheses to be stored efficiently, enabling further processes like rescoring with more sophisticated models to improve transcription accuracy. These lattices can be pruned, determinized, and minimized for efficiency.

- The PLF (Python Lattice Format) used in Moses represents word lattices as directed acyclic graphs with nodes and edges labeled with words and weights. Nodes are ordered topologically and assigned numerical IDs, with edges indicating the word, probability, and distance between nodes. The format allows efficient decoding of multiple sentence hypotheses by Moses.

For LLM inferences, the PLF lattice format is generally more friendly than Kaldi's native lattice format. This is because PLF is designed to be easily readable and manipulable using Python, which is commonly used in LLM development. PLF's structure (nodes and edges with word and weight labels) aligns well with typical data processing pipelines used in LLMs. In contrast, while efficient for speech recognition tasks, Kaldi's format is more complex and less intuitive for direct use with LLMs, as shown in Figure 2.

IV. Experiments

A. Experimental Settings

1. LLM used: The experiment employs the Japanese-Llama-2-7b model[1]. The Japanese-Llama-2-7b model is presum-

[1]huggingface.co/elyza/ELYZA-japanese-Llama-2-7b

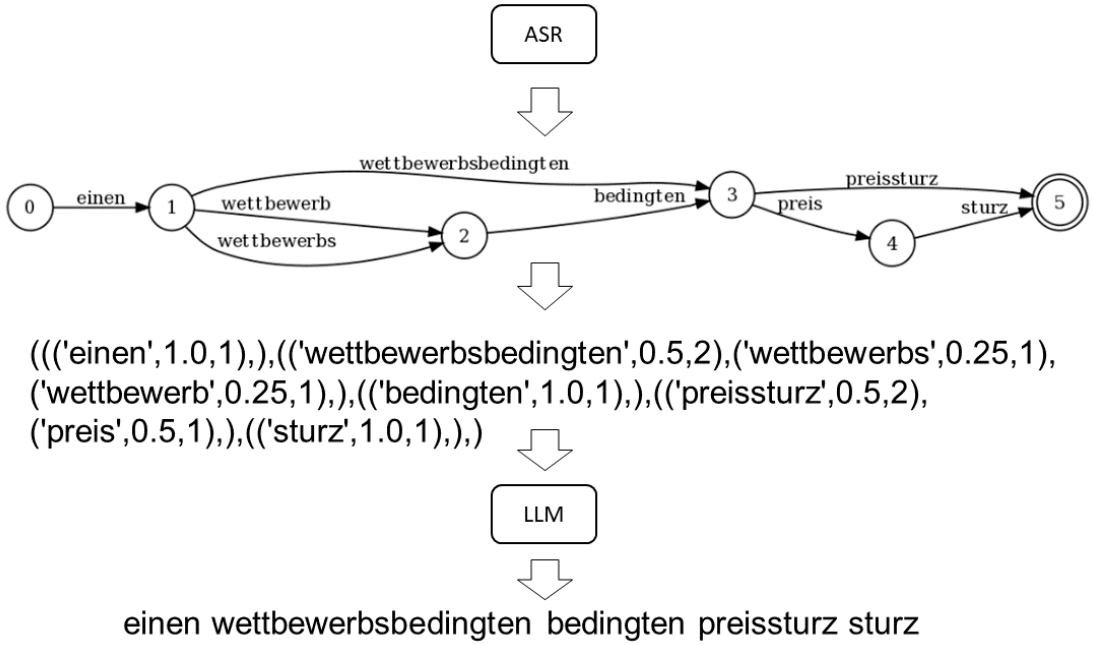

Fig. 2: Convert graph representation of lattice from ASR system to text-based format for LLM (example cited from Moses Users Manual [17], and the meaning is: a competition-related price drop).

ably a variant of the LLaMA (Large Language Model by Meta AI) adapted for Japanese language processing.

2. Dataset: We use the "Corpus of Spontaneous Japanese (CSJ)" [18]. Three official evaluation sets ($Eval_1$, $Eval_2$, and $Eval_3$), each containing ten lecture recordings [19], are used to evaluate the speech recognition results.
3. ASR model: We use the very deep resnet tdnn CTC model (similar to the Chain model in Kaldi) in our previous work [20], [21] for generating lattice for this paper. For training ASR models, we use approximately 250 hours of lecture recordings as the training set (Train APS (academic)) with similar settings according to other benchmarks [22], [23], [24], [19].
4. Training LLM GER models: Implementing our LLM GER models is based on the repository[2] for GER task [25]. For the LLM test set, we extracted each 424 utterances from each $Eval_1$ (2918), $Eval_2$ (3067), and $Eval_3$ (2484). The left $Eval_1$, $Eval_2$, and $Eval_3$ not included in the LLM test set were gathered and used as our LLM train set (7197 utterances in total). In CSJ data, the training is performed on an NVIDIA Tesla A6000 GPU using 8-bit training. The hyperparameters for finetuning are 17 epochs, learning rate 1e-4, batch size 64, and LoRA rank 4. We used an English-Japanese prompt based on automatic translation following the previous default English prompt.[3]
5. Evaluations tool: We use NIST-SCTK to evaluate the Word Error Rate (WER%).

B. Experimental Results

In this paper, we made four systems for experimental comparisons as follows:

- 1-best: This is the direct 1-best hypothesis output from the above-mentioned ASR model in the last subsection.
- N-best: We use the LLM GER method described in [15] to restore the n-best hypotheses (n=10). The top 10 hypotheses are extracted from the lattice using the default acoustic weight by the lattice-to-nbest tool [4] of Kaldi.
- Lattice: We use the LLM GER method with the same setting as the N-best experiments, just replacing the n-best hypotheses with the PLF-formated lattice hypotheses. Since the CSJ words are with part-of-speech (POS) and Katakana pronunciations (Kana), we remove these POS and Kana from the PLF-format lattice.
- Lattice + POS + Kana: The LLM GER experimental settings are the same as the Lattice experiment but keep the POS and Kana from the PLF-format lattice.

Table I shows WERs in each CSJ evaluation set. We observe that the LLM GER lattice (without POS and Kana) results improved in eval02 and eval03 and resulted in overall improvement compared to n-best hypotheses. On the other hand, there were no improvement in LLM GER lattice with POS and Kana for leveraging lattice rescoring in LLM. We

[2]https://github.com/Hypotheses-Paradise/Hypo2Trans

[3]We observed roughly lower loss and lower CER trends in English-Japanese prompt compared to default English prompt (https://github.com/Hypotheses-Paradise/Hypo2Trans/blob/main/H2T-LoRA/templates/H2T-LoRA.json).

[4]https://kaldi-asr.org/doc/lattice-to-nbest_8cc.html

TABLE I: WER in CSJ Eval test sets. Each evaluation data consists of 424 utterances sampled from the original CSJ evaluation data.

	$Eval_1$ (424)	$Eval_2$ (424)	$Eval_3$ (424)
1-best	**5.6**	10.9	10.9
N-best	5.9	**10.4**	10.9
Lattice	6.0	**10.4**	**10.4**
Lattice + POS + Kana	9.4	18.9	12.3

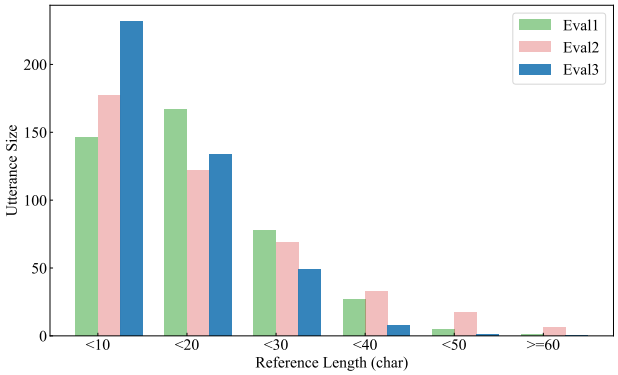

Fig. 3: The reference length distribution in $Eval_1$, $Eval_2$ and $Eval_3$.

expect that these tag information became confusion for LLM model we expect.

C. Further Discussions

1) Performance in each reference length regime: We further analyze the reason that n-best and lattice-based LLM GER methods could not outperform the 1-best hypotheses in $Eval_1$. Figure 3 shows the utterance numbers of $Eval_1$, $Eval_2$ and $Eval_3$ in each reference length regime. We see that $Eval_3$ consists of relatively shorter sentences compared to $Eval_1$. The $Eval_3$ has the most number of sentences which are shorter than 10. Figures 4, 5 and 6 show the averaged WER scores in each reference length regime. These results show that the WERs were lower for samples when the reference lengths are shorter than 10 in all evaluation sets. In $Eval_3$ and $Eval_2$, all WER scores were lower in length regimes shorter than 50. We expect that recovering errors from long sentences remains challenging for the LLM GER method. We also expect that using small LLM (7B) may make the LLM GER more difficult to process longer inputs because it is trained with data. We expect that because there are more small size output in $Eval_2$ and $Eval_3$ (especially when the reference length is shorter 10) than $Eval_1$, the LLM GER lattice results became better in $Eval_2$ and $Eval_3$.

2) Output examples: To see the trends by LLM GER correction, Table II show some output examples both in proposed LLM GER lattice and baseline 1-best. In Example 1, 2 and 3, we see all of the 1-best outputs include error which is caused by phonetic similarity. These 1-best outputs are not natural as Japanese and have major problems, such as the lack of meaning in words. On the other hand, in LLM GER Lattice,

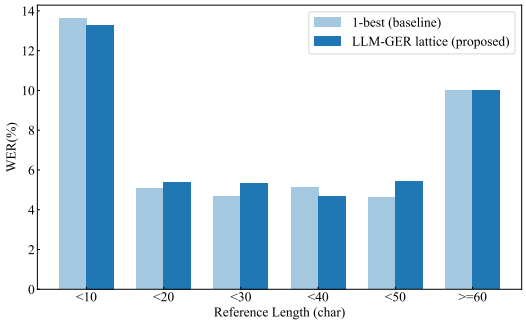

Fig. 4: WERs in $Eval_1$

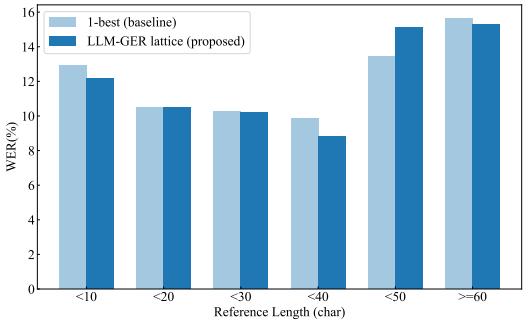

Fig. 5: WERs in $Eval_2$

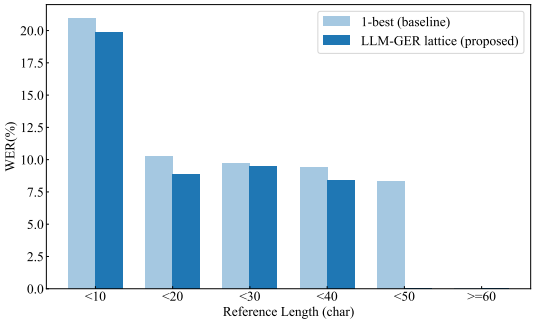

Fig. 6: WERs in $Eval_3$

Fig. 7: Averaged WERs in each reference length regimes in $Eval_1$, $Eval_2$ and $Eval_3$.

we see more recovery with correction and the corrected outputs became more similar to references. In Example 4, we see some of unknown tokens were included in 1-best output but they were converted to correct word. In $Eval_2$, we observed some of the 1-best outputs included unknown tokens. After the correction by LLM GER lattice, we observed all of the unknown tokens were removed or substituted with corrected words.

3) The advantage for using lattice: We find the process of the lattice-based data using LLM has a resemblance to the traditional ASR decoder. Here's the process within a traditional decoder for language model rescoring, which involves re-evaluating the paths in a lattice using more sophisticated models. The initial decoding uses a simpler language model to assign scores to different paths in the lattice. In rescoring, these initial scores are replaced with scores from a more complex,

TABLE II: Example Japanese sentences in baseline 1-best and proposed LLM GER lattice.

Example 1: Phonetic similar errors ($Eval_3$)	
1-best (baseline)	えっとニセコの あのっぷり スキー場というところに行ってきました
LLM GER Lattice (proposed)	えとニセコの **アンヌ** スキー場というところに行ってきました
Reference	えとニセコの **アンヌ**プリスキー場というところに行ってきました
Example 2: Phonetic similar errors ($Eval_3$)	
1-best (baseline)	当然全域 に慣れてない
LLM GER Lattice (proposed)	**と全然雪** に慣れてない
Reference	**と全然雪** に慣れてない
Example 3: Phonetic similar errors ($Eval_2$)	
1-best (baseline)	今回はディバイドと呼ばれる周波数 へへん環境 を用いた為に周波数および
LLM GER Lattice (proposed)	今回はディバイドと呼ばれる周波数 **変換** を用いた為に周波数および
Reference	今回はディバイダーと呼ばれる周波数 **変換** 器を用いた為に周波数および
Example 4: Correcting unknown tokens ($Eval_2$)	
1-best (baseline)	関係してきてるのではないでしょ <UNK> <UNK> <UNK> <UNK> か
LLM GER Lattice (proposed)	関係してきてるのではないでしょ う か
Reference	関係してきてるのではないでしょ う か

typically larger, language model (i.e., LLM in this paper). The words with new scores reflect a more accurate evaluation of word sequences, improving overall transcription accuracy.

To sum up, the method we proposed in this paper leverages the strengths of advanced LLMs to refine the transcription output by reassessing the probabilities of more diverse hypotheses in the lattice.

V. Conclusion

This paper proves that using LLM-based GER can directly work on lattice-structured output from the speech recognition system. Experiments on CSJ corpus show that compared with using n-best hypotheses, using lattice can improve the performance of Japanese speech recognition. In the future, we will conduct more experiments to demonstrate the method's effectiveness in a broader range of settings.

Acknowledgment

We would like to thank the Tohoku-NICT matching funding's support.

References

[1] S. Zhang, M. Lei, and Z. Yan, "Investigation of transformer based spelling correction model for ctc-based end-to-end mandarin speech recognition." in *Proc. Interspeech*, 2019, pp. 2180–2184.

[2] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *CoRR abs/1706.03762*, 2017.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. NAACL*, 2019, pp. 4171–4186.

[4] S. Zhang, H. Huang, J. Liu, and H. Li, "Spelling error correction with soft-masked bert," *arXiv preprint arXiv:2005.07421*, 2020.

[5] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the knowledge of BERT for sequence-to-sequence ASR," *CoRR*, vol. abs/2008.03822, 2020.

[6] J. Salazar, D. Liang, T. Q. Nguyen, and K. Kirchhoff, "Masked language model scoring," *arXiv:1910.14659*, 2019.

[7] J. Shin, Y. Lee, and K. Jung, "Effective sentence scoring method using BERT for speech recognition," in *Proc. ACML*, 2019, pp. 1081–1093.

[8] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for asr," in *Proc. IEEE-ICASSP*, 2020, pp. 7694–7698.

[9] W.-N. Hsu, Y.-H. H. Tsai, B. Bolte, R. Salakhutdinov, and A. Mohamed, "Hubert: How much can a bad teacher benefit asr pre-training?" in *Proc. IEEE-ICASSP*, 2021, pp. 6533–6537.

[10] C. Wang, Y. Wu, S. Liu, M. Zhou, and Z. Yang, "Curriculum pre-training for end-to-end speech translation," *arXiv preprint arXiv:2004.10093*, 2020.

[11] J. Lu, D. Batra, D. Parikh, and S. Lee, "Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks," *arXiv preprint arXiv:1908.02265*, 2019.

[12] L. H. Li, M. Yatskar, D. Yin, C.-J. Hsieh, and K.-W. Chang, "Visualbert: A simple and performant baseline for vision and language," *arXiv preprint arXiv:1908.03557*, 2019.

[13] L. Zhou, H. Palangi, L. Zhang, H. Hu, J. Corso, and J. Gao, "Unified vision-language pre-training for image captioning and vqa," in *Proc. AAAI*, 2020, pp. 13 041–13 049.

[14] X. Li, X. Yin, C. Li, P. Zhang, X. Hu, L. Zhang, L. Wang, H. Hu, L. Dong, F. Wei *et al.*, "Oscar: Object-semantics aligned pre-training for vision-language tasks," in *Proc. ECCV*, 2020, pp. 121–137.

[15] 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," *ArXiv*, vol. abs/2309.15701, 2023.

[16] J. E. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *ArXiv*, vol. abs/2106.09685, 2021.

[17] P. Koehn, "MOSES Statistical Machine Translation System User Manual and Code Guide," 2014. [Online]. Available: http://www.statmt.org/moses/manual/manual.pdf

[18] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[19] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the corpus of spontaneous japanese," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[20] S. Li, X. Lu, R. Takashima, P. Shen, T. Kawahara, and H. Kawai, "Improving CTC-based Acoustic Model with Very Deep Residual Time-delay Neural Networks," in *Proc. Interspeech 2018*, 2018, pp. 3708–3712.

[21] ——, "Improving very deep time-delay neural network with vertical-attention for effectively training ctc-based asr systems," in *2018 IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 77–83.

[22] T. Moriya, T. Shinozaki, and S. Watanabe, "Kaldi recipe for Japanese spontaneous speech recognition and its evaluation," in *Autumn Meeting of ASJ*, no. 3-Q-7, 2015.

[23] N. Kanda, X. Lu, and H. Kawai, "Maximum a posteriori based decoding for CTC acoustic models," in *Proc. INTERSPEECH*, 2016, pp. 1868–1872.

[24] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-Attention based End-to-End speech recognition with a deep CNN Encoder and RNN-LM," in *Proc. INTERSPEECH*, 2017.

[25] 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.