MULTI-STAGE LARGE LANGUAGE MODEL CORRECTION FOR SPEECH RECOGNITION

Jie Pu, Thai-Son Nguyen, and Sebastian Stüker

Zoom Video Communications, Karlsruhe, Germany

ABSTRACT

In this paper, we investigate the usage of large language models (LLMs) to improve the performance of competitive speech recognition systems. Different from traditional language models that focus on one single data domain, the rise of LLMs brings us the opportunity to push the limit of state-of-the-art ASR performance, and at the same time to achieve higher robustness and generalize effectively across multiple domains. Motivated by this, we propose a novel multi-stage approach to combine traditional language model re-scoring and LLM prompting. Specifically, the proposed method has two stages: the first stage uses a language model to re-score an N-best list of ASR hypotheses and run a confidence check; The second stage uses prompts to a LLM to perform ASR error correction on less confident results from the first stage. Our experimental results demonstrate the effectiveness of the proposed method by showing a 10% $\sim$ 20% relative improvement in WER over a competitive ASR system — across multiple test domains.

***Index Terms*—** Speech Recognition, Large Language Models

1. INTRODUCTION

Large language models (LLMs) such as ChatGPT [1], Llama [2] have gained more and more interests in several research communities as well as in industrial groups. With the capacity of modeling long-range dependencies and natural interactions with the input, they have been changing the conventional ways of solving NLP problems. Motivated by the potential usefulness, there have been an increasing number of studies that employ LLMs for improving different ASR tasks.

Recent works in using external LMs for improving recognition performance can be folded into three categories: language model fusion [3, 4], hypothesis re-scoring (re-ranking) [5, 6] and leveraging prompt-based LLMs [7, 8]. Firstly, LM fusion refers to the approaches that fuse the likelihood scores from ASR and LMs during decoding inference. Hypothesis re-scoring methods use unconstrained LMs to re-score and identify the best output from an N-best list of ASR hypotheses. More recently, prompt-based approaches advance traditional methods with the use of LLMs that support prompting. While these different works prove effective, they share the same limit in which a decent amount of in-domain data was required for training and fine-tuning their key language models. The need for in-domain data reduces their practical usefulness, as in many real-world applications, in-domain data are not available e.g., due to user privacy concerns or hardness of data collection. How to make ASR systems domain independent is always challenging. Due to the lack of generalization, many models, despite high performance on one or two in-domain test sets, can be quite fragile and make many mistakes when evaluated in just slightly different domains [9, 10].

In this work, we study general LLMs such as GPT-J [11] and ChatGPT for correcting the transcript output of multi-domain end-to-end ASR systems while not exploiting any in-domain data for LMs. Specifically, our work is based on two findings found when we employ LLMs for ASR correction and when in-domain textual data is not present. First, we observe that if both the ASR and the LLM have low confidence in identifying the best hypothesis among an N-best list, then the decision based on their fusion scores tends to be inaccurate (the selected hypothesis is more likely to contain errors). Secondly, prompt-based LLM approaches are powerful but hit a limit as they steer more towards written language and lack of acoustic context in their revision process.

To overcome these drawbacks, we propose a multi-stage ASR correction pipeline that utilizes different aspects of LMs in different stages to explore better transcript output. In the first stage, we use a LM to re-score the N-best list of ASR hypotheses and obtain utterance-level scores. If the discrimination of these utterance-level scores does not surpass a confidence level, then the N-best list will be passed to the second stage for further correction. In the second stage, we proposed a novel prompt design that exploits word-level dependencies from the N-best list itself to provide a final transcript revision. Experiments showed that with the proposed correction pipeline, we can achieve 10% $\sim$ 20% word-error-rate (WER) relative improvement consistently while not using any in-domain data for LMs. Our best system for the LibriSpeech benchmark could reach 1.3% WER on LibriSpeech test-clean set, which sets a new state-of-the-art.

2. METHODOLOGY

We propose a multi-stage correction pipeline using multiple language models as illustrated in Figure 1. The first stage

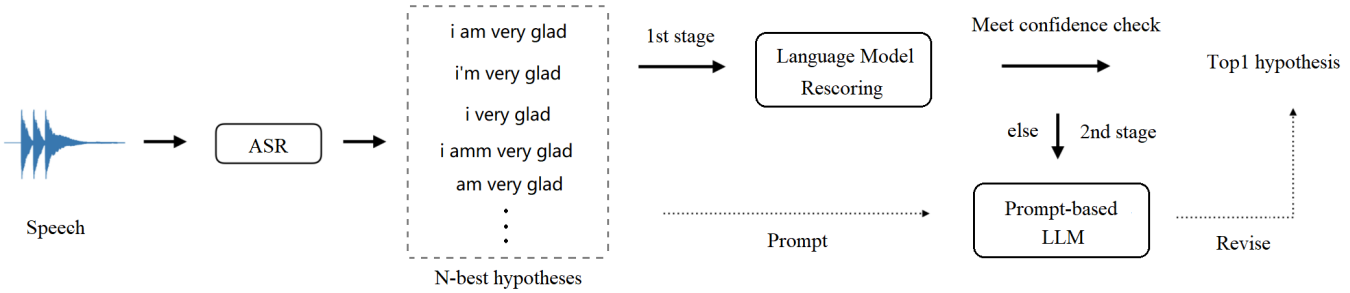

Fig. 1. Overview of the multi-stage correction pipeline. The first stage uses a LM for re-scoring an N-best list hypotheses while the second stage uses a prompt-based LLM. The confidence check will decide if the output is the best hypothesis from the first stage, or will be revised in the second stage.

uses a LM for re-scoring while the second stage requires a LLM that supports prompting. The input of the pipeline is an N-best list of hypotheses from an ASR system. We perform a confidence check after the first stage to decide if the output stays as the best hypothesis from the re-scoring, or will be revised in the next stage.

2.1. Stage 1: Language Model Re-scoring

Assume that we have an N-best list of hypotheses $\mathbf{y_1}, \dots \mathbf{y_N}$ which are generated from the decoding inference for an utterance $\mathbf{x}$. In this stage, we employ a LM that provides a sentence-level probability for hypothesis $\mathbf{y_i}$. Following the works in [5, 6], we calculate the score $Score(\mathbf{y_i})$ used for re-scoring as:

$$Score(\mathbf{y_i}) = \log \mathcal{P}_{\text{ASR}}(\mathbf{y_i}|\mathbf{x}) + \alpha \log \mathcal{P}_{\text{LM}}(\mathbf{y_i}) \qquad (1)$$

where $\mathcal{P}_{\text{LM}}(\mathbf{y_i})$ is the LM probability for hypothesis $\mathbf{y_i}$, and $\mathcal{P}_{\text{ASR}}(\mathbf{y_i}|\mathbf{x})$ is the probability obtained from the ASR. The weight α and the size of the N-best list N are found using a development set.

2.2. Confidence Check

In our study, we want to classify the obtained re-scoring scores into either high confidence or low confidence. Inspired by [12], we normalize the scores of one N-best list with *Softmax* function and simply evaluate $Score_{\text{best}}$, the highest normalized score. If $Score_{\text{best}}$ is larger than a threshold β (high confidence), then this hypothesis with the highest score is believed to be good enough and the process completes at this step. Otherwise, we will further perform the next stage.

2.3. Stage 2: Large Language Model Correction

In this stage, we utilize a LLM that supports prompting to perform transcript correction. The idea of this stage is to exploit the linguistic context and the links between the words across the N-best list to form a final transcript, rather than simply re-select a hypothesis. For that, we propose a prompting approach which accepts a list of hypotheses and a set of additional rules for a constrained output. The details of the prompt design is described in Algorithm 1. We categorize different aspects that inspired us for this prompt as follows:

Algorithm 1 Prompt for LLM

Input: an N-best list from ASR, containing N hypotheses. They are ranked by scores during beam search **y**[1], **y**[2], ..., **y**[n].

Prompt: I want you to check and correct potential errors in one sentence according to the following rules. Here is the sentence to work on: **y**[1].
You need to first consider the following variant sentences and try to pick corrected words from them: **y**[2], **y**[3], ... **y**[n].

Additional rules for this modification:

1. If any word in the original sentence looks weird or inconsistent, then replace it with a corresponding word from variant sentences.
2. You don't have to modify the original sentence if it already looks good.
3. Keep the sentence structure and word order intact.
4. Only replace words in the original sentence with ones from variant sentences. Do not simply add or delete words.
5. Try to make the corrected sentence have the same number of words as the original sentence.
6. Ignore punctuation.
7. Use U.S. English.
8. Output only one modified sentence and no explanation.

Output: the revised hypothesis $\hat{\mathbf{y}}$[1]

- **New words**. Rule 4 restricts the LLM to only use words from the N-best list, otherwise the LLM may use synonyms in the correction process. We also notice that a LLM such as ChatGPT tends to format ASR transcripts by adding conjunctions or removing repetitions, which results in more coherent sentences, however does not fit the purpose of speech transcription.

- **Creativity**. Rules 3 and 5 are designed to confine the creativity and the level of freedom in the LLM, by restricting the structure and length of its output sentence, so that the output will stay close to a verbatim transcript for the input speech.
- **Output standardization**. Rule 7 can be changed to other English variants and spelling systems, e.g. U.K. Rules 6 and 8 are included for the convenience of ASR evaluation. These rules can be extended and optimized if exists a NLP downstream task that consumes ASR transcripts such as translation or summarization.

Herein we primarily experiment with ChatGPT, but those design principles can be easily extended to other LLMs.

3. EXPERIMENTAL SETUP

3.1. Data Sets

We used several publicly available English data sets for our experiments:

- **LibriSpeech** (**LS**) [13] is a collection of around 960 hours of read speech from audio books, which are a part of the LibriVox project. We used the standard split for train, validation and test sets (test-clean, test-other).
- **Common Voice V8** (**CV**) [14] consists of about 900 hours of English transcribed audio where speakers record text from Wikipedia. This data set has a large variation in quality and speakers, as anyone can submit recorded contributions.
- **TED-LIUM 3** (**TL**) [15] contains 452 hours of speech from TED talks. This data set represents presentation speech which is a popular domain nowadays.
- **Multilingual LibriSpeech** (**MLS**) [16] is an extension of LibriSpeech and contains 44.5K hours of English speech. It is also derived from audio books of LibriVox. We use this dataset for the benchmark when a large of mount of data available.

3.2. ASR Models

We extract 40 log-mel filterbank coefficients with mean normalization as the input features, and use SpecAugment [17] for data augmentation during training. Labels are generated from a sub-word tokenzier with the vocab size of 4000 units.

The attention-based sequence-to-sequence ASR models were built and trained following [18]. In all experiments, we used the same encoder network consisting of two convolutional layers and six layers of bidirectional LSTMs with 1,280 cells, and the decoder network with two unidirectional LSTM layers with 1,280 cells. For optimization, we adopt an unified training schedule in which ASR models get updated every 8000 tokens, and trained for 50 epochs.

Table 1. WERs in % of different ChatGPT configurations on LibriSpeech dev-clean set. GPT-J and ChatGPT language models and the ASR model trained on LS and MLS, are used.

Model	Temperature	Allow new words	WER
ChatGPT-3.5	0.7	Y	2.6
ChatGPT-4	0.7	Y	1.7
ChatGPT-4	0.5	N	1.5
ChatGPT-4	0.2	N	**1.4**

Table 2. WERs in % of different N-best list sizes, the weight α and the confidence level threshold β on LibriSpeech dev-clean set. GPT-J and ChatGPT language models and the ASR model trained on the multi-domain data (LS, CV and TL) are used. For each N, the optimal values of α and β are presented here. Given the confidence threshold, a percentage % of total speech utterances will be sent to the stage 2.

N-best	α	β	WER	Percentage to Stage 2
5	3.0	0.70	**2.1**	23.0
8	4.5	0.45	2.2	7.8
16	4.5	0.60	2.2	19.7

3.3. Language Models

For *Stage 1*, we used GPT-J [11] with 6-billion parameters. It is an auto-regressive and decoder-only transformer model. For *Stage 2* we explored ChatGPT-3.5 and ChatGPT-4 with versions released on March 2023. Their optimal configurations have been investigated on two specific factors: 1) whether to allow new words outside of the N-best lists, i.e. Rule 4 in the proposed prompt, and 2) the hyper-parameter *temperature* in ChatGPT. Table 1 shows a comparison between different configurations. We can see that ChatGPT-4 performs better than ChatGPT-3.5 due to its capability to handle more complex instructions in prompts [1]. Lowering the value of *temperature* reduces the randomness and creativity of ChatGPT-4's output, thus helps to reduce the WER. The best performance is obtained when disallowing new words.

3.4. Finding Hyper-parameters

Table 2 shows the WER numbers for different N-best list sizes, the weight α in Equation 1, and the confidence level threshold β, which are tuning parameters for our correction pipeline. We have tested the beam size $N = [5, 8, 16]$ and tuned the weight α ranged from 1 to 5 with a step size 0.1. The confidence level threshold β was tuned with the range from 0 to 1 with a step size 0.05. As we can see from the table, the best configuration is $N = 5$, $\alpha = 3.0$ and $\beta = 0.7$. These numbers will be used and fixed for all other experiments. Besides, only 23% of total processed speech utterances will be sent to *Stage 2* and revised by a prompt-based LLM (77% utterances will be directly outputted from the stage 1), thus alleviates the total computational cost of the pipeline.

Table 3. WERs in % of large-scale ASR evaluation. The best performance is in bold.

Method	In-domain LM	Libri-test-clean	Libri-test-other	Multilingual LibriSpeech
MLS [16]	N	2.1	4.0	6.8
	Y	1.8	3.5	**5.9**
Wav2vec 2.0 Large [19]	N	2.2	4.5	-
	Y	1.8	3.3	-
Pre-training + Noisy student [20]	N	1.6	3.3	-
	Y	1.4	**2.6**	-
Whisper Large-v2 [9]	N	2.5	4.9	6.2
Our ASR	N	1.6	4.2	6.6
+ Correction Stage 1	N	1.5	3.9	6.3
+ Correction Stage 2	N	2.4	4.4	6.2
+ Correction Stage 1 & 2	N	**1.3**	3.4	6.0

Table 4. WERs in % of multi-domain evaluation. The best performance is in bold.

Method	Libri-test-clean	CV	TL
ASR	2.8	15.3	7.0
+ Correction Stage 1	2.5	13.9	6.8
+ Correction Stage 2	2.7	13.9	6.9
+ Correction Stage 1 & 2	**2.1**	**13.4**	**6.5**

4. RESULTS

4.1. Multi-domain Evaluation

To evaluate how the proposed approach is generalized to different domains, we used an ASR model trained on a mix of three popular data sets (LibriSpeech, Common Voice and TED-LIUM), and examined the proposed correction pipeline with the conventional test sets of these domains. WER results are shown in Table 4.

Overall, the use of GPT-J and ChatGPT helps to improve the recognition accuracy, even when we did not do any LM adaptation with in-domain textual data. To explore the exact benefit from individual stages, we performed the pipeline with single stages solely, i.e., setting β (the confidence level threshold) to be 1 or 0. For *Stage 2* only: the correction with ChatGPT is effective but inconsistent, which gives a large improvement on one test set (Common Voice), but smaller on other test sets. For *Stage 1* only: the improvement gained by employing GPT-J is higher and on-par with the use of an in-domain LM i.e., compared to the results in [17] of similar models and test sets.

The best performance is consistently obtained when performing the full pipeline (*Stage 1* + *Stage 2*). On LibriSpeech test set, the proposed approach achieves 25% WER relative improvement over the ASR baseline (2.8 $\rightarrow$ 2.1). With the help of the LLMs and the proposed confidence check, we could achieve a large WER relative reduction of 16% after *Stage 1* (2.5 $\rightarrow$ 2.1). This clearly shows the benefits of combining different LMs in our multi-stage correction pipeline.

4.2. Large-scale ASR Evaluation

To examine if the proposed approach still works for large-scale ASR, we created a new training data set by merging the LibriSpeech with a large amount of English read speech from the Multilingual LibriSpeech corpus. We trained a new ASR model with the same size and similar optimizations.

As shown in Table 3, our large-scale ASR baseline performs well on all the read speech sets, especially on test-clean. Its performance is comparable to the best systems reported in [16, 19, 20, 21]. When applying single correction stages (similarly to what had been done in Section 4.1), we observed a contrast result. The re-scoring in *Stage 1* still gives consistent improvement but the prompt-based ChatGPT correction in *Stage 2* downgrade the performance on LibriSpeech. We manually reviewed this phenomenon and found the correction of ChatGPT could produce undesired changes. Specifically, the correction would steer more towards written language, instead of a reasonable verbatim transcript for the input speech. Also, ChatGPT often tries to correct grammar errors in speech but introduces degradation in WER. This observation reveals the weakness of a solely prompt-based correction for ASR.

With the full pipeline, the proposed approach was shown to successfully mitigate the incorrect correction introduced by ChatGPT. At the end, we can achieve a consistent improvement of 10-19% relatively cross three test sets in this large-scale ASR benchmark. On LibriSpeech test-clean, our result of 1.3% WER has made a new state-of-the-art record.

5. CONCLUSION

In this paper, we propose a novel multi-stage approach to leverage different LLMs for ASR correction. The proposed approach has been shown to provide 10% $\sim$ 20% relative improvement for competitive and multi-domain ASR systems. In future, we would like to experiment with more LLMs for both stages, and investigate if the use of available in-domain data can further improve the correction performance.

6. REFERENCES

[1] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al., "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[3] 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," *IEEE ICASSP*, pp. 1–5828, 2018.

[4] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, "Cold fusion: Training seq2seq models together with language models," *Proc. Interspeech*, pp. 387–391, 2018.

[5] 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 *Annual Conference of the International Speech Communication Association*, 2022.

[6] Xianrui Zheng, Chao Zhang, and Philip C Woodland, "Adapting gpt, gpt-2 and bert language models for speech recognition," in *IEEE ASRU*, 2021, pp. 162–168.

[7] Mengxi Nie, Ming Yan, Caixia Gong, and D Chuxing, "Prompt-based re-ranking language model for asr," *Proc. Interspeech 2022*, pp. 3864–3868, 2022.

[8] Rao Ma, Mengjie Qian, Potsawee Manakul, Mark Gales, and Kate Knill, "Can generative large language models perform asr error correction?," *arXiv preprint arXiv:2307.04172*, 2023.

[9] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale weak supervision," in *ICML*, 2023, pp. 28492–28518.

[10] Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Paden Tomasello, Jacob Kahn, Gilad Avidov, Ronan Collobert, and Gabriel Synnaeve, "Rethinking evaluation in asr: Are our models robust enough?," *arXiv preprint arXiv:2010.11745*, 2020.

[11] Ben Wang and Aran Komatsuzaki, "GPT-J-6B: A 6 Billion Parameter Autoregressive Language Model," 2021.

[12] Tim Pearce, Alexandra Brintrup, and Jun Zhu, "Understanding softmax confidence and uncertainty," *arXiv preprint arXiv:2106.04972*, 2021.

[13] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *IEEE ICASSP*, 2015, pp. 5206–5210.

[14] 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 preprint arXiv:1912.06670*, 2019.

[15] François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia Tomashenko, and Yannick Esteve, "Ted-lium 3: Twice as much data and corpus repartition for experiments on speaker adaptation," in *SPECOM*. Springer, 2018, pp. 198–208.

[16] Vineel Pratap, Qiantong Xu, Anuroop Sriram, Gabriel Synnaeve, and Ronan Collobert, "Mls: A large-scale multilingual dataset for speech research," *arXiv preprint arXiv:2012.03411*, 2020.

[17] Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *Interspeech 2019*, 2019.

[18] Thai-Son Nguyen, Sebastian Stueker, Jan Niehues, and Alex Waibel, "Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation," in *IEEE ICASSP*, 2020, pp. 7689–7693.

[19] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, pp. 12449–12460, 2020.

[20] Yu Zhang, James Qin, Daniel S Park, Wei Han, Chung-Cheng Chiu, Ruoming Pang, Quoc V Le, and Yonghui Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.

[21] William Chan, Daniel Park, Chris Lee, Yu Zhang, Quoc Le, and Mohammad Norouzi, "Speechstew: Simply mix all available speech recognition data to train one large neural network," *arXiv preprint arXiv:2104.02133*, 2021.