

Merging the Knowledge of LLMs for Automatic Speech Recognition

Hayato Futami¹, Tatsuya Kawahara¹

¹ Graduate School of Informatics, Kyoto University, Japan

futami@sap.ist.i.kyoto-u.ac.jp

Abstract

Automatic speech recognition (ASR) systems, trained on paired speech-text data, have been improved by leveraging language models (LMs) trained on text-only data. LM fusion methods such as shallow fusion and density ratio are well-established methods that incorporate external LMs during ASR decoding. However, they incur additional computational costs due to LM inference, which is particularly problematic for recent larger LMs. In this study, we propose incorporating external LMs via model merging. This method integrates the LMs directly into the parameters of an LLM-based ASR model, requiring no additional computational cost at inference. We formulate domain extension and transfer via arithmetic operations on LoRA parameters. Experimental evaluations were conducted for the domain adaptation of LLM-based ASR trained on CSJ and LibriSpeech. We show that our LM merging consistently improved the ASR performance in the target domains, without degrading inference speed or memory footprint.

Index Terms: speech recognition, LLM, model merging

1. Introduction

End-to-end automatic speech recognition (ASR) models have achieved significant progress in recent years, demonstrating excellent performance in general domains [1, 2, 3]. This success has been driven mainly by large-scale, speech-text paired data coupled with scalable model architectures. Consequently, these models often suffer from performance degradation in specific domains lacking sufficient paired speech-text data. As text-only data is much easier to collect, it has been widely explored to apply external language models (LMs) trained on text of the target domains. Shallow fusion [4] is a well-established approach that runs a target-domain LM during ASR decoding, where the log-linear interpolation of the ASR and the LM scores are used. Density ratio fusion [5] is an extension of shallow fusion for better domain transfer, where the score of a source-domain LM is further subtracted from the interpolated score.

Recently, following the remarkable success of large language models (LLMs) in natural language processing [6, 7], there has been a significant trend toward applying LLMs in the field of ASR. LLM-based ASR is a prominent application, where pre-trained LLMs are fine-tuned for speech processing tasks by incorporating speech encoders [8, 9, 10]. These models can leverage the knowledge of powerful LLMs as foundations, becoming the new standard approach for ASR [11]. Beyond ASR, they are emerging as foundation models capable of solving a wide range of speech processing tasks [12, 13]. This study also focuses on the LLM-based ASR approach. LLM-based ASR incorporates the knowledge of an LLM at the initialization phase before ASR training, which cannot be dynamically adjusted according to target domains at inference time. To

this end, LM fusion approaches are applied to adapt LLM-based ASR to the target domain. Recently, domain-specific LMs are often built by fine-tuning from foundational LLMs. It is computationally expensive to use such large LMs in shallow fusion or density ratio fusion, where the LMs are called at every decoding step [14, 15].

In this study, we propose integrating the knowledge of LLMs into ASR models via model merging. Model merging is a technique that integrates multiple models by performing arithmetic operations in their weight spaces [16, 17, 18]. Our proposed LM merging integrates LMs directly into the ASR model, whereas conventional LM fusion methods rely on off-the-shelf LMs at decoding steps. A key advantage of our method is that it does not introduce any additional modules or computational overhead over the ASR model. In this study, we utilize an LLM-based ASR model adapted from a pre-trained LLM using LoRA [19], following [9]. Domain-specific LMs are also built via LoRA fine-tuning from the same LLM. We perform arithmetic operations between the ASR and LM adapters. We propose two merging methods for domain extension and transfer. Domain extension merging corresponds to shallow fusion, where the adapter weights of the target-domain LM are added to ASR. Domain transfer merging corresponds to density ratio fusion, where the adapter weights of the source-domain LM are further subtracted. These methods are further enhanced by TIES-merging [20], an advanced model merging method that addresses task interference in merging.

We conducted experimental evaluations of ASR domain adaptation in Japanese and English. We built ASR models based on LLM-jp-3-980M [21] and LLaMA3.2-1B [7], trained on CSJ-SPS [22] and LibriSpeech [23] corpora, respectively. The models were then adapted to CSJ-APS and SPGISpeech [24] domains using LMs. First of all, we confirmed that the proposed LM merging methods consistently improved the ASR performance in the target domains. We also compared them with conventional LM fusion methods (shallow fusion and density ratio fusion) and n -best rescoring [25, 26]. We observed that while LM merging did not perform as well as LM fusion, it was competitive to n -best rescoring on CSJ. It is important to note that our method does not introduce any additional parameters or latency, unlike LM fusion and rescoring. We also found that the combination with LM fusion or rescoring further improved the ASR performance.

2. Related work

2.1. Language model adaptation for ASR

As end-to-end ASR models are trained on paired speech-text data, external LMs are often adapted to leverage the knowledge of text-only data. Shallow fusion is a widely used approach that

applies LMs during decoding, in the form of score interpolation between ASR and LM [4, 15]. Density ratio [5] extends shallow fusion by subtracting the source-domain LM score to facilitate domain transfer. Internal LM estimation [27] instead subtracts the internal LM score from the end-to-end model. Recently, delayed fusion [14] has been proposed as an extension of shallow fusion to calculate the LM score with a delay, thereby reducing the number of LLM inference calls. However, these LM fusion methods suffer from the computational overhead of invoking an LM inference call at each step (or every few steps in [14]).

N -best rescoring is also a standard approach that applies LMs as a post-processing step, where n -best hypotheses from ASR are rescored using the LM score [25, 26]. Recently, with the advancement of LLMs, generative error correction has emerged [28, 29]. Given n -best hypotheses, LLMs are prompted to generate the correct transcripts. These post-processing methods require only a single LLM inference call after decoding, which is computationally less expensive than LM fusion. However, they increase the total parameter footprint. Moreover, their performance is usually limited for small n , especially in the case of greedy decoding ($n = 1$), although greedy decoding is preferred in recent LLM-based ASR.

Some studies apply LMs via knowledge distillation [30, 31], where the knowledge of a teacher LM is transferred to a student ASR model. LMs are applied during ASR training, thus the method does not add LM inference cost. However, since the LMs are only utilized during training and remain static during inference, lacking the flexibility required for domain adaptation.

2.2. Model merging

Model merging is a technique that merges the parameters of two or more models to build a new model, fusing their knowledge [17, 18]. Task arithmetic [16] formulates model merging via task vectors, which represent the parameter differences induced by fine-tuning on a specific task. The study introduces three arithmetic operations: task vector negation, addition, and task analogies. Task vector addition enables building a multi-task model, and task analogy facilitates improved domain generalization, without requiring access to data or additional training. Subsequently, several advanced task arithmetic methods have been proposed. For instance, TIES-merging [20] and DARE [32] sparsify parameters based on their magnitude to mitigate task interference. Task arithmetic has been further extended to parameter-efficient fine-tuning modules, such as LoRA [33, 34].

While task arithmetic has been applied to a wide range of domains and tasks including LLMs and image recognition [17], several studies have applied it to ASR. In [35], task arithmetic between ASR models has been explored for domain expansion and to simulate training data expansion for low-resource ASR. In [36], a synthetic-to-real task vector is used to mitigate the gap of training on synthetic data. In [37], task arithmetic is applied to language expansion for speech translation models. In [38], rare word recognition and translation are tackled via task arithmetic. That study focuses on merging ASR models trained on synthetic data containing rare words, which differs from our approach of merging LMs into ASR, i.e., cross-modal merging.

3. Language model fusion

Let $\mathbf{X}$ denote the input acoustic features and $\mathbf{y}$ denote the text tokens in the transcript. An end-to-end ASR model estimates $p_{\text{asr}}(\mathbf{y}|\mathbf{X})$, directly mapping $\mathbf{X}$ to $\mathbf{y}$. LM fusion methods leverage an external LM $p_{\text{lm}}(\mathbf{y})$ during ASR inference by interpolating the output probabilities $p_{\text{asr}}(\mathbf{y}|\mathbf{X})$ and $p_{\text{lm}}(\mathbf{y})$.

3.1. Shallow fusion

Shallow fusion is a widely adopted approach for interpolating $p_{\text{asr}}(\mathbf{y}|\mathbf{X})$ and $p_{\text{lm}}(\mathbf{y})$ [4]. The end-to-end ASR model searches for the hypothesis that maximizes the log-linear interpolated score of the ASR model and the LM, denoted as:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} [\log p_{\text{asr}}(\mathbf{y}|\mathbf{X}) + \alpha \log p_{\text{lm},\text{tgt}}(\mathbf{y})]. \quad (1)$$

For domain adaptation, a target-domain LM trained on text data of the target domain, denoted as $p_{\text{lm},\text{tgt}}(\mathbf{y})$, is used. α in Eq. (1) is a hyperparameter.

3.2. Density ratio

Density ratio [5] is an extension of shallow fusion, which uses Bayes' theorem to define $p_{\text{asr}}(\mathbf{y}|\mathbf{X})$ using $p_{\text{lm}}(\mathbf{y})$. The acoustic likelihood for the source and target domains are represented via Bayes' theorem as

$$p_{\text{src}}(\mathbf{X}|\mathbf{y}) = p_{\text{src}}(\mathbf{X})p_{\text{src}}(\mathbf{y}|\mathbf{X})/p_{\text{src}}(\mathbf{y}), \quad (2)$$

$$p_{\text{tgt}}(\mathbf{X}|\mathbf{y}) = p_{\text{tgt}}(\mathbf{X})p_{\text{tgt}}(\mathbf{y}|\mathbf{X})/p_{\text{tgt}}(\mathbf{y}). \quad (3)$$

Give an assumption that $p_{\text{src}}(\mathbf{X}|\mathbf{y}) = p_{\text{tgt}}(\mathbf{X}|\mathbf{y})$, the posterior $p_{\text{tgt}}(\mathbf{y}|\mathbf{X})$ is represented as

$$p_{\text{tgt}}(\mathbf{y}|\mathbf{X}) \propto \frac{p_{\text{tgt}}(\mathbf{y})}{p_{\text{src}}(\mathbf{y})} p_{\text{src}}(\mathbf{y}|\mathbf{X}). \quad (4)$$

We assume $p(\mathbf{y}|\mathbf{X})$ and $p(\mathbf{y})$ are modeled by the end-to-end ASR model and LM, respectively. The following score is used in beam search:

$$\hat{\mathbf{y}} = \arg \max_{\mathbf{y}} [\log p_{\text{asr}}(\mathbf{y}|\mathbf{X}) + \alpha \log p_{\text{lm},\text{tgt}}(\mathbf{y}) - \beta \log p_{\text{lm},\text{src}}(\mathbf{y})], \quad (5)$$

where $p_{\text{lm},\text{src}}(\mathbf{y})$ denotes a source-domain LM trained on the transcripts of the ASR training data.

Both LM fusion methods will improve the ASR performance on the target domain using LMs. However, they require inference with the target-domain LM (as well as the source-domain LM in density ratio) at every step in the beam search, which increases computational cost during ASR inference.

4. Language model merging

We consider an LLM-based ASR model built by attaching a speech encoder to a pre-trained LLM. The LLM parameters are adapted via the Low-Rank Adaptation (LoRA) for the ASR task, following [8, 9, 10]. For each adapted layer l , the parameters $\theta^{(l)}$ are represented as:

$$\theta_{\text{asr},\text{src}}^{(l)} = \theta_{\text{pre}}^{(l)} + \theta_{\Delta\text{asr},\text{src}}^{(l)}, \quad (6)$$

where $\theta_{\text{pre}}^{(l)}$ denotes the parameters of the base LLM and $\theta_{\Delta\text{asr},\text{src}}^{(l)}$ denotes the LoRA adapter parameters for ASR. Similarly, a target-domain LM is fine-tuned from the same pre-trained LLM, denoted as:

$$\theta_{\text{lm},\text{tgt}}^{(l)} = \theta_{\text{pre}}^{(l)} + \theta_{\Delta\text{lm},\text{tgt}}^{(l)}, \quad (7)$$

where $\theta_{\Delta\text{lm},\text{tgt}}^{(l)}$ denotes the LoRA adapter parameters for the target-domain LM.

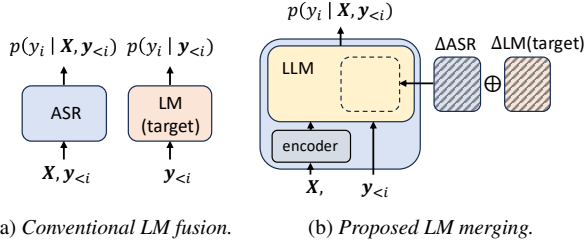


Figure 1: Illustration of (a) conventional LM fusion and (b) proposed LM merging for domain extension. LM merging integrates ASR and target-domain LM within the model, which does not add any computational costs at inference.

4.1. Domain extension merging

First, we propose extending the ASR model to the target domain by leveraging the knowledge of the target-domain LM. This corresponds to shallow fusion described in Section 3.1, which integrates the ASR model and the LM at the output log-probability level. In contrast, we integrate the knowledge of them within the parameter space, as illustrated in Figure 1. We define domain extension merging as:

$$\theta^{(l)} = \theta_{\text{pre}}^{(l)} + \theta_{\Delta\text{ASR},\text{src}}^{(l)} + \lambda_{\alpha} \theta_{\Delta\text{lm},\text{tgt}}^{(l)}, \quad (8)$$

where the element-wise addition of the parameters is computed with a coefficient hyperparameter λ_{α} .

Eq. (8) can be further enhanced by advanced merging methods. Specifically, we consider TIES-merging¹ in this study. First, we trim the parameters to retain the top- $p\%$ values by magnitude:

$$\hat{\theta} = \text{trim}(\theta, p). \quad (9)$$

Next, we elect a consensus sign $s^*[j]$ for each parameter index:

$$s^*[j] = \text{sgn}(\hat{\theta}_{\Delta\text{ASR},\text{src}}[j] + \lambda_{\alpha} \hat{\theta}_{\Delta\text{lm},\text{tgt}}[j]). \quad (10)$$

Finally, we merge only the parameters whose signs matched the elected signs $s^*[j]$:

$$\theta[j] = \theta_{\text{pre}}[j] + \delta(\hat{\theta}_{\Delta\text{ASR},\text{src}}, s^*[j]) + \lambda_{\alpha} \delta(\hat{\theta}_{\Delta\text{lm},\text{tgt}}, s^*[j]), \quad (11)$$

where $\delta(x, s) = x$ if $\text{sign}(x) = s$ and 0 otherwise.

This approach is derived from task vector addition in task arithmetic and its extension to parameter-efficient modules [16, 33]. We build a multi-task model capable of performing both ASR and target-domain LM, by adding the corresponding task vectors $\theta_{\Delta\text{ASR},\text{src}}^{(l)}$ and $\theta_{\Delta\text{lm},\text{tgt}}^{(l)}$. Unlike [16, 33], which explore model merging within a single modality, this study explores cross-modal model merging. We aim to improve target-domain speech-to-text (i.e. ASR) performance without target-domain paired data, by merging a speech-to-text model and a target-domain LM.

4.2. Domain transfer merging

Secondly, we propose transferring the ASR model to the target domain, which corresponds to density ratio described in Section 3.2. We define domain transfer merging, by introducing a

source-domain LM and its LoRA adapter $\theta_{\Delta\text{lm},\text{src}}^{(l)}$, as:

$$\theta^{(l)} = \theta_{\text{pre}}^{(l)} + \theta_{\Delta\text{ASR},\text{src}}^{(l)} + \lambda_{\alpha} \theta_{\Delta\text{lm},\text{tgt}}^{(l)} - \lambda_{\beta} \theta_{\Delta\text{lm},\text{src}}^{(l)}. \quad (12)$$

Eq. (12) can be enhanced by TIES-merging, as in Section 4.1.

This approach is derived from task analogies in task arithmetic [16, 33]. We identify the relationship that “source-domain LM is to target-domain LM and source-domain ASR is to target-domain ASR”. The task vector for target-domain ASR can be derived from the remaining three task vectors [16]. Domain transfer merging offers better performance in target domains. However, it requires access to the source-domain text used for ASR training, which is not always available. In such cases, only domain extension merging is applicable.

The proposed LM merging offers a clear advantage over the conventional LM fusion explained in Section 3. While LM fusion requires running one or two LMs in addition to the ASR model during inference, LM merging allows for running only an ASR model into which the parameters of the LM have been merged. Therefore, it introduces no additional computational overhead at inference time, which is particularly beneficial when using recent large-scale LLMs. In addition, LM merging retains a single-model forward pass, which can be easily deployed and optimized with existing inference engines (e.g. vLLM).

5. Experimental evaluations

We conducted experimental evaluations on ASR domain adaptation covering two adaptation scenarios in Japanese and English. First, we used the Corpus of Spontaneous Japanese (CSJ) [22]. CSJ consists of 520 hours of Japanese oral presentations, including CSJ-APS (240h) of academic presentations and CSJ-SPS (280h) of simulated public speaking on everyday topics. In the CSJ experiments, we treated CSJ-SPS as a source domain, where paired data were available to train an ASR model. Then, CSJ-APS was treated as a target domain, where only text data were available, following [39]. In addition to CSJ, we used LibriSpeech [23] and SPGISpeech [24]. LibriSpeech [23] comprises 960 hours of English public-domain book readings. SPGISpeech [24] is based on corporate earnings calls, representing the financial domain. In the experiments, we tested domain adaptation from LibriSpeech to SPGISpeech³, where only text data were used for SPGISpeech, following [40].

We built an LLM-based ASR model by fine-tuning LLM-jp-3-980M [21] on CSJ-SPS. We added a Conformer-based [41] speech encoder to the LLM, comprising 512 dimensions, 8 attention heads, and 12 layers. The encoder parameters were fully fine-tuned, while the LLM parameters were updated via LoRA [19], following [9]. We applied LoRA to the key, query, value and output layers of the self-attention modules with a rank $R = 8$ and $\alpha = 16$. The model was trained using Adam optimizer of max learning rate 0.0005 with 25k warmup steps then decay. We applied SpecAugment [42] as well as speed perturbation [43] of $\times 0.9$ and $\times 1.1$. We used an auxiliary CTC loss of weight 0.3. Decoding was performed using beam search of a beam size 5 and with joint CTC decoding of weight 0.3 [44]. We built source-domain and target-domain LMs by LoRA fine-tuning from LLM-jp-3-980M, on CSJ-APS and CSJ-SPS, respectively. We also applied LoRA with the same configuration as ASR, trained with Adam optimizer of learning rate 0.00001 with decay scheduling⁴. The implementation are based on ES-Pnet [45] for ASR with Huggingface’s Transformers [46].

³We used randomly selected 10k of 39k samples for evaluation.

⁴The mean absolute change in the parameters was 0.0001, while it was 0.0037 for ASR.

¹We follow the algorithm [18] <https://github.com/arcee-ai/mergekitt>

²Task vector is defined as the difference between the task-specific and pre-trained model, corresponding to the LoRA adapter parameters.

Table 1: *LM merging for domain adaptation from CSJ-SPS (eval3) to CSJ-APS (eval1).*

	CER(%) ↓	
	eval1	eval3
ASR	13.9	4.8
+Merge-E	13.6	4.8
+TIESmerge-E	13.4	4.8
+Merge-T	13.3	5.1
+TIESmerge-T	13.3	4.9

Table 2: *Comparison and combination with conventional LM adaptation methods on CSJ.*

	CER↓	Params↓	RTF↓
ASR	13.9	1.1B	0.45
+TIESmerge-T	13.3	1.1B	0.45
+SF	12.8	2.1B	0.52
+DR	12.5	3.1B	0.57
+Rescore	13.3	2.1B	0.45
+Rescore (DR)	13.2	3.1B	0.46
+TIESmerge-T+SF	12.4	2.1B	0.53
+TIESmerge-T+DR	12.4	3.1B	0.58
+TIESmerge-T+Resc	12.7	2.1B	0.45
+TIESmerge-T+Resc(DR)	12.8	3.1B	0.46

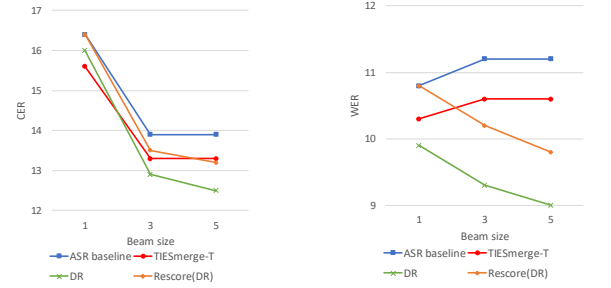
Table 1 shows the results of our proposed LM merging on CSJ. We applied domain extension merging (Merge-E) as defined in Eq. (8) and domain transfer merging (Merge-T) in Eq. (12). In addition to simple arithmetic operations (i.e., element-wise addition or subtraction), we further applied TIES-merging [20] (TIESmerge-T/E) as defined in Eq. (11) with $p = 50\%$. We found that the proposed LM merging methods improved the ASR performance over the baseline in the target domain (eval1). Domain transfer merging outperformed domain extension merging. For the source domain (eval3), domain extension merging maintained baseline performance, while domain transfer merging degraded the performance. These results align with the objective of our methods for domain extension and transfer. Also, we observed that TIES-merging yielded further gains.

Table 2 presents a comparison and combination of our LM merging with other LM adaption methods. We compared TIESmerge-T with shallow fusion (SF), density ratio (DR), and 5-best rescoring (Rescore). We also considered rescoring using the density ratio (Rescore-DR), where the score of the source-domain LM is subtracted as in Eq. (5). We report the total number of model parameters (Params) and the real time factor (RTF) on an NVIDIA RTX A6000 GPU. We found that LM merging achieved a CER almost competitive with rescoring. However, LM merging did not perform as well as SF and DR. This can be explained by that SF and DR explicitly incorporate the target LM knowledge at every decoding step, while LM merging does not. Also, we observed a cross-modal alignment problem: increasing the LM coefficient (λ_α in Eq. (8)) leads to a collapse in ASR, making it difficult to sufficiently transfer the LM knowledge. Note that LM merging has an advantage in model size, RTF and deployment flexibility. We investigated the combination of LM merging with other methods, where we found that these combinations yielded further performance gains.

Table 3 presents the results of ASR domain adaptation from LibriSpeech to SPGISpeech. We built an LLM-based ASR model using LLaMA3.2-1B [7]. Different from CSJ experi-

Table 3: *LM merging for LibriSpeech-to-SPGISpeech adaptation.*

	WER↓	Params↓	RTF↓
ASR	11.2	1.4B	0.40
+TIESmerge-E	10.6	1.4B	0.40
+TIESmerge-T	10.6	1.4B	0.40
+SF	9.1	2.4B	0.49
+DR	9.0	3.4B	0.56
+Rescore	9.8	2.4B	0.41
+Rescore(DR)	9.8	3.4B	0.41
+TIESmerge-T+SF	8.9	2.4B	0.49
+TIESmerge-T+DR	8.8	3.4B	0.55
+TIESmerge-T+Resc	9.3	2.4B	0.40
+TIESmerge-T+Resc(DR)	9.3	3.4B	0.41



(a) CSJ-SPS to CSJ-APS

(b) LibriSpeech to SPGISpeech

Figure 2: *LM adaptation methods on different beam sizes.*

ments, we utilized the WavLM-base-plus SSL model [47] as a feature extractor frontend, followed by an E-Branchformer [48] encoder with 256 dimensions, 4 attention heads, and 12 layers. Both the E-Branchformer encoder parameters and the LoRA adapters in the LLM were updated during training. As shown in Table 3, we observed that the proposed LM merging also improved the ASR performance in the target domain (SPGISpeech). Although the WER reduction was limited compared to other methods, the proposed LM merging offers the advantages of not increasing parameters and RTF over the baseline ASR. Furthermore, we demonstrated that combining LM merging with other methods yields additional performance gains.

Figure 2 compares TIES-merging (TIESmerge-T) against DR and DR-based rescoring with different beam sizes. Our LM merging improved ASR even for greedy decoding (beam size of 1), where rescoring cannot be applied. The improvements obtained by DR are dependent on the beam size; the gains at beam sizes 1 and 3 were not as significant as beam size 5. For greedy decoding on CSJ, TIES-merging even outperformed DR.

6. Conclusions

In this study, we have explored improving the ASR performance in a specific domain, leveraging a target-domain LM. To this end, we propose integrating LMs into ASR via model merging. We introduce two merging methods, domain extension and domain transfer, defined by parameter arithmetic between ASR and LMs. We experimentally demonstrated that LM merging yielded consistent improvement in Japanese and English. Although the performance improvement was modest compared to SF and DR, LM merging does not increase memory footprint or latency and is executed within a single-model forward pass. As future work, we will explore more flexible LM merging between different model architectures [49].

7. Generative AI Use Disclosure

Generative AI was used only for proofreading the sentences in the manuscript.

8. References

- [1] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *ICML*, 2023.
- [2] R. Prabhavalkar, T. Hori, T. N. Sainath, R. Schluter, and S. Watanabe, "End-to-end speech recognition: A survey," *TASLP*, vol. 32, pp. 325–351, 2023.
- [3] Y. Peng, M. Shakeel, Y. Sudo, W. Chen, J. Tian, C.-J. Lin, and S. Watanabe, "OWSM v4: Improving open whisper-style speech models via data scaling and cleaning," in *Interspeech*, 2025, pp. 2225–2229.
- [4] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *ICASSP*, 2018, pp. 1–5828.
- [5] E. McDermott, H. Sak, and E. Variani, "A density ratio approach to language model fusion in end-to-end automatic speech recognition," in *ASRU*, 2019, pp. 434–441.
- [6] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altschmidt, S. Altman, S. Anadkat *et al.*, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.
- [7] A. Grattafiori, A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Vaughan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.
- [8] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu *et al.*, "On decoder-only architecture for speech-to-text and large language model integration," in *ASRU*, 2023, pp. 1–8.
- [9] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli *et al.*, "Prompting large language models with speech recognition abilities," in *ICASSP*, 2024, pp. 13 351–13 355.
- [10] Z. Chen, H. Huang, A. Andrusenko, O. Hrinchuk, K. C. Puvvada, J. Li, S. Ghosh, J. Balam, and B. Ginsburg, "SALM: Speech-augmented language model with in-context learning for speech recognition and translation," in *ICASSP*, 2024, pp. 13 521–13 525.
- [11] G. Saon, A. Dekel, A. Brooks, T. Nagano, A. Daniels, A. Satt, A. Mittal, B. Kingsbury, D. Haws, E. Morais *et al.*, "Granite-speech: open-source speech-aware llms with strong english asr capabilities," *arXiv preprint arXiv:2505.08699*, 2025.
- [12] D. Ding, Z. Ju, Y. Leng, S. Liu, T. Liu, Z. Shang, K. Shen, W. Song, X. Tan, H. Tang *et al.*, "Kimi-Audio technical report," *arXiv preprint arXiv:2504.18425*, 2025.
- [13] J. Xu, Z. Guo, H. Hu, Y. Chu, X. Wang, J. He, Y. Wang, X. Shi, T. He, X. Zhu *et al.*, "Qwen3-Omni technical report," *arXiv preprint arXiv:2509.17765*, 2025.
- [14] T. Hori, M. Kocour, A. Haider, E. McDermott, and X. Zhuang, "Delayed fusion: Integrating large language models into first-pass decoding in end-to-end speech recognition," in *ICASSP*, 2025, pp. 1–5.
- [15] K. Hu, T. N. Sainath, B. Li, N. Du, Y. Huang, A. M. Dai, Y. Zhang, R. Cabrera, Z. Chen, and T. Strohman, "Massively multilingual shallow fusion with large language models," in *ICASSP*, 2023, pp. 1–5.
- [16] G. Ilharco, M. T. Ribeiro, M. Wortsman, L. Schmidt, H. Hajishirzi, and A. Farhadi, "Editing models with task arithmetic," in *ICLR*, 2023. [Online]. Available: <https://openreview.net/forum?id=6t0Kwf8-jrj>
- [17] E. Yang, L. Shen, G. Guo, X. Wang, X. Cao, J. Zhang, and D. Tao, "Model merging in llms, mllms, and beyond: Methods, theories, applications, and opportunities," *ACM Computing Surveys*, vol. 58, no. 8, pp. 1–41, 2026.
- [18] C. Goddard, S. Siriwardhana, M. Ehghaghi, L. Meyers, V. Karpukhin, B. Benedict, M. McQuade, and J. Solawetz, "Arcee's MergeKit: A toolkit for merging large language models," in *EMNLP*, 2024, pp. 477–485.
- [19] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *ICLR*, 2022. [Online]. Available: <https://openreview.net/forum?id=nZeVKeeFYf9>
- [20] P. Yadav, D. Tam, L. Choshen, C. Raffel, and M. Bansal, "TIES-Merging: Resolving interference when merging models," *NeurIPS*, vol. 36, pp. 7093–7115, 2023.
- [21] A. Aizawa, E. Aramaki, B. Chen, F. Cheng, H. Deguchi, R. Enomoto, K. Fujii, K. Fukumoto, T. Fukushima, N. Han *et al.*, "LLM-jp: A cross-organizational project for the research and development of fully open japanese LLMs," *arXiv preprint arXiv:2407.03963*, 2024.
- [22] K. Maekawa, "Corpus of Spontaneous Japanese : its design and evaluation," in *SSPR*, 2003, p. paper MMO2.
- [23] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015, pp. 5206–5210.
- [24] P. K. O'Neill, V. Lavrukhin, S. Majumdar, V. Noroozi, Y. Zhang, O. Kuchaiev, J. Balam, Y. Dovzhenko, K. Freyberg, M. D. Shulman, B. Ginsburg, S. Watanabe, and G. Kucsko, "SPGISpeech: 5, 000 hours of transcribed financial audio for fully formatted end-to-end speech recognition," in *Interspeech*, 2021, pp. 1434–1438.
- [25] T. Chen, C. Allauzen, Y. Huang, D. S. Park, D. Rybach, W. R. Huang, R. Cabrera, K. Audhkhasi, B. Ramabhadran, P. J. Moreno, and M. Riley, "Large-scale language model rescoring on long-form data," *ICASSP*, pp. 1–5, 2023.
- [26] A. Ogawa, N. Kamo, K. Matsuura, T. Ashihara, T. Moriya, T. Kano, N. Tawara, and M. Delcroix, "Applying LLMs for rescoring n-best ASR hypotheses of casual conversations: Effects of domain adaptation and context carry-over," *arXiv preprint arXiv:2406.18972*, 2024.
- [27] Z. Meng, S. Parthasarathy, E. Sun, Y. Gaur, N. Kanda, L. Lu, X. Chen, R. Zhao, J. Li, and Y. Gong, "Internal language model estimation for domain-adaptive end-to-end speech recognition," in *SLT*, 2021, pp. 243–250.
- [28] 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 *ASRU*, 2023, pp. 1–8.
- [29] 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," *NeurIPS*, vol. 36, pp. 31 665–31 688, 2023.
- [30] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Distilling the knowledge of BERT for sequence-to-sequence ASR," in *Interspeech*, 2020, pp. 3635–3639.
- [31] J. Lee and H. Seo, "Online knowledge distillation of decoder-only large language models for efficient speech recognition," in *Interspeech*, 2024, pp. 2890–2894.
- [32] L. Yu, Y. Bowen, H. Yu, F. Huang, and Y. Li, "Language models are super mario: Absorbing abilities from homologous models as a free lunch," in *ICML*, 2024.
- [33] J. Zhang, S. Chen, J. Liu, and J. He, "Composing parameter-efficient modules with arithmetic operation," in *NeurIPS*, 2023. [Online]. Available: <https://openreview.net/forum?id=5r3e27I9Gy>
- [34] Z. Zhao, T. Shen, D. Zhu, Z. Li, J. Su, X. Wang, and F. Wu, "Merging loRAs like playing LEGO: Pushing the modularity of loRA to extremes through rank-wise clustering," in *ICLR*, 2025. [Online]. Available: <https://openreview.net/forum?id=j6fsbpAllN>

- [35] G. Ramesh, K. Audhkhasi, and B. Ramabhadran, "Task vector algebra for ASR models," in *ICASSP*, 2024, pp. 12 256–12 260.
- [36] H. Su, H. Farn, F.-Y. Sun, S.-T. Chen, and H.-y. Lee, "Task arithmetic can mitigate synthetic-to-real gap in automatic speech recognition," in *EMNLP*, 2024, pp. 8905–8915.
- [37] Y.-F. Cheng, H. Futami, Y. Kashiwagi, E. Tsunoo, W. S. Teo, S. Arora, and S. Watanabe, "Task arithmetic for language expansion in speech translation," *arXiv preprint arXiv:2409.11274*, 2024.
- [38] R. Jing, C. Gong, Y. Jiang, B. Zhu, S. Liu, C. Zhang, X.-L. Zhang, and X. Li, "Rare word recognition and translation without fine-tuning via task vector in speech models," *arXiv preprint arXiv:2512.21894*, 2025.
- [39] H. Futami, H. Inaguma, S. Ueno, M. Mimura, S. Sakai, and T. Kawahara, "Non-autoregressive error correction for CTC-based ASR with phone-conditioned masked lm," in *Interspeech*, 2022, pp. 3889–3893.
- [40] Y. Li, Y. Wu, J. Li, and S. Liu, "Prompting large language models for zero-shot domain adaptation in speech recognition," in *ASRU*, 2023, pp. 1–8.
- [41] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech*, 2020, pp. 5036–5040.
- [42] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019, pp. 2613–2617.
- [43] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Interspeech*, 2015, pp. 3586–3589.
- [44] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *ACL*, Jul. 2017, pp. 518–529.
- [45] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Yalta, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.
- [46] T. Wolf, L. Debut, V. Sanh, J. Chaumond, C. Delangue, A. Moi, P. Cistac, T. Rault, R. Louf, M. Funtowicz *et al.*, "HuggingFace's Transformers: State-of-the-art natural language processing," *arXiv preprint arXiv:1910.03771*, 2019.
- [47] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1505–1518, 2022.
- [48] K. Kim, F. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-Branchformer: Branchformer with enhanced merging for speech recognition," in *SLT*, 2022, pp. 84–91.
- [49] C. Cui, B. Yang, F. Shen, Y. Chen, J. Zheng, X. Wang, A. Zhang, and T.-S. Chua, "Transport and merge: Cross-architecture merging for large language models," *arXiv preprint arXiv:2602.05495*, 2026.