Learning When to Trust Which Teacher for Weakly Supervised ASR

Aakriti Agrawal[*] *Milind Rao*[†] *Anit Kumar Sahu*[†] *Gopinath Chennupati*[†] *Andreas Stolcke*[†]

[*]University of Maryland, U.S.A. [†]Amazon Alexa AI, U.S.A.

agrawal5@umd.edu,{milinrao,anitsah,chennug,stolcke}@amazon.com

Abstract

Automatic speech recognition (ASR) training can utilize multiple experts as teacher models, each trained on a specific domain or accent. Teacher models may be opaque in nature since their architecture may be not be known or their training cadence is different from that of the student ASR model. Still, the student models are updated incrementally using the pseudo-labels generated independently by the expert teachers. In this paper, we exploit supervision from multiple domain experts in training student ASR models. This training strategy is especially useful in scenarios where few or no human transcriptions are available. To that end, we propose a *Smart-Weighter* mechanism that selects an appropriate expert based on the input audio, and then trains the student model in an unsupervised setting. We show the efficacy of our approach using LibriSpeech and LibriLight benchmarks and find an improvement of 4 to 25% over baselines that uniformly weight all the experts, use a single expert model, or combine experts using ROVER.

Index terms: ASR, teacher-student training, semi-supervised learning, self-supervised learning, ROVER.

1. Introduction

Self-supervised learning approaches [16, 10, 3] for ASR usually rely on a single expert teacher model to generate pseudo-labels to train the student ASR models. ROVER [5] is a classic technique to generate a single best transcription by aligning alternate teacher hypotheses and using a rule like majority voting. Mixture-of-expert (MoE) approaches for speech [14, 20], on the other hand, make use of the MoE layers in training the student model. Generally, when we have multiple independent experts, each of them is trained to be performant for a specific domain. In practice, these expert models can not be typically deployed on devices with limited compute and storage. Even in a resource-rich, cloud-based setting, these experts are hard to train because the experts are heterogeneous in terms of their structure (e.g., hybrid or deep neural networks based), in terms of their dependence on external language models, and in terms of size. Therefore, we treat these experts as opaque generators of transcripts for a given audio input.

As an alternative to using MoE layers, which involves additional access to the experts beyond output transcripts, we propose *Smart-Weighter*, a method that selects a domain expert among many and with access limited to the generated transcripts. The selection of experts is conditioned on the input audio, i.e., for a given training utterance we select the best expert for generating the teacher transcript.

Work done while the first author was an intern with Amazon Alexa.

In this paper, we use the streaming-compatible recurrent neural network transducer (RNN-T) [7] ASR model, whose training objective is to maximize the probability of the transcript tokens given the audio and the past context. We develop three RNN-T-based domain experts and a separate student model, all of which are of the same size. We train the experts on LibriSpeech; the experts are trained on mutually exclusive data subsets to mimic domain experts. The student model is then trained, along with the *Smart-Weighter* network, on untranscribed audio, with the selected experts producing the transcripts. While this framework is generally applicable to alternate forms of feedback, such as weak supervision, in this work we focus on using transcripts from expert models.

Our main contribution is an unsupervised framework for learning from multiple expert models using a *Smart-Weighter* network that selects domain experts based on the unlabeled input audio.

2. Related Work

ROVER [5] combines multiple transcripts using equal weights or recognition confidences [1] only. Smart-Weighter additionally makes use of utterance audio to determine transcript relevance. Another disadvantage of ROVER is that it asymptotes quickly as the number of experts increases. Furthermore, a ROVER expert can itself be added to the list of experts that the Smart-Weighter weights over. Alternatively, Smart-Weighter could be used to estimate the weights of the different inputs prior to combination with ROVER.

A popular approach to teacher-student learning is knowledge distillation (KD). In KD, the student is trained to match with the teacher's output distribution by minimizing the KL divergence between either their bottleneck layer or output activations. However, KD is not practical with opaque experts that do not provide access to activations. Although KD was originally used to learn from a single model, [11] uses multiple teachers while using uncertainty-based KD. In [22] reinforcement learning (RL) is used to select teachers for KD. The approach relies on feedback/reward based on the performance of the student model to update the policy parameters. Unlike in [22], here we consider an unsupervised setting where we do not have access to ground-truth transcripts to generate rewards, such as word error rate (WER). KD also has been used for multi-level-multi-teacher [12] and methods based on error rate instead of loss [6].

Another line of work utilizing multiple experts is mixture of experts (SpeechMoe2 [21], DeepMoe [19]). A router chooses among experts in each neural network layer during training, given an input acoustic embedding. In this setting, the experts and student are trained end-to-end, leading to a less computationally expensive solution when expert and student are combined. [2] uses an RL-based policy to mask the activation in

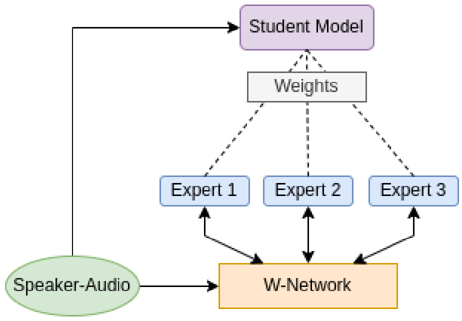

Figure 1: *Visualization of training or updating a student model given unlabeled audio. For a given utterance, we have teacher transcripts from multiple opaque experts of differing quality. A Smart-Weighter (W-network) consumes expert transcriptions and utterance audio to weight their quality, with a larger weight given to experts deemed to be more accurate. The student model is trained using semi-supervised learning with audio and paired expert transcriptions using the determined weights.*

each layer in the mixture of experts. This method is not applicable to our scenario since our domain experts are typically large and reside in the cloud. Also, the experts cannot be trained along with the student model.

Speech enhancement solutions have also been proposed to use multiple experts. Like our work, [8] presents a gating mechanism for expert selection and loss propagation in end-to-end training. The gating mechanism learns to choose appropriate weights while training the experts. [15] uses an ensemble model to train specialized experts based on different speakers, giving significant gains in speech enhancement. Speakers are partitioned based on their characteristics, using k-means clustering. We took inspiration from the clustering-based approach and gating mechanism found in this earlier work.

3. Method

Our focus is on training and updating student ASR models given unlabeled audio and arbitrary opaque teacher ASR models. In this section, we first describe the Smart-Weighter model that either selects an appropriate expert or weights its transcripts for a given utterance. We then describe the setup used to obtain teacher models using LibriSpeech [13], although the ideas developed here are applicable to arbitrary experts. Finally, we describe how student ASR models are trained from scratch given a stream of unlabeled audio (e.g., LibriLight [9]) using transcripts obtained from multiple experts weighted by the Smart-Weighter model. This overall workflow is shown in Figure 1.

3.1. Smart-Weighter

In order to train a student model using these multiple expert transcriptions of differing quality paired with audio, we develop a Smart-Weighter that selects or weights an appropriate expert given the utterance input.

This is done by generating weights for each expert that sum to 1, thereby conforming the weighting to a probability simplex. A larger weight is assigned to experts that are deemed to be more accurate for that utterance.

The Smart-Weighter network shown in Figure 2 takes as input an acoustic signal and transcriptions from the expert models (we use three experts here, but the method is applicable to an arbitrary number). It uses a unidirectional LSTM-based speech encoder trained on the LibriLight dataset to generate acoustic signal embeddings and a pretrained BERT model [4] to generate expert-transcription embeddings. The transcription

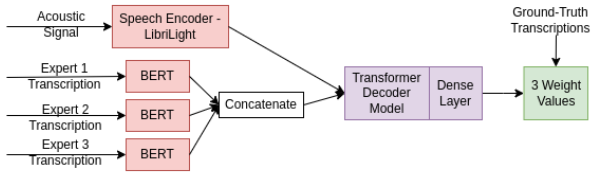

Figure 2: *The Smart-Weighter consists of a speech encoder that produces features from an utterance audio and a BERT language model that produces features from expert transcriptions. A transformer-decoder model consumes the BERT features while cross-attending to audio features. The outputs are processed to determine the weights of the expert models.*

Table 1: *Number of speakers used for training expert ASR models on LibriSpeech partitions. Speakers are partitioned either randomly or clustered by speaker embeddings.*

	Expert 1	Expert 2	Expert 3
Random	779	779	780
Clustered	488	1074	758

embeddings are then joined together using a separator token embedding and used as input for a transformer-decoder model [18] that cross-attends to the acoustic embeddings. We use a 6-layer transformer-decoder model (512 units and 8 attention heads) that uses full self-attention on the input, which is a concatenation of the BERT embeddings of expert transcriptions with full cross-attention to relevant sections of the acoustic embeddings. The output of the transformer-decoder layer is then pooled and passed through feed-forward dense layers with intermediate ReLU activations and a final layer using softmax activation. Finally, we obtain weight values $w_i, i = 1, 2, 3$, corresponding to the three experts.

The Smart-Weighter is trained on a 100-hour subset of the LibriSpeech dataset also using ground-truth transcriptions. For an utterance, we obtain expert weights w_i and we develop target labels $z \in \{0, 1\}^3$ where $z_i = 1$ if expert i has the lowest word error rate (WER) as measured using ground-truth transcriptions, and $z_i = 0$ otherwise. We then apply binary cross entropy loss, i.e., $L = -\sum_i z_i \log w_i + (1 - z_i) \log(1 - w_i)$ on each of the expert weights. Thus the Smart-Weighter is trained to upweight expert transcriptions that show lowest WER and produce lower weights for experts that show poor performance.

3.2. Expert Setup

We treat the expert models as opaque models that may have arbitrary architectures, model sizes, training methodologies or training sets, and may include unspecified domain-specific auxiliary language models. The experts may have comparable performance or be trained on complementary data with minimal overlap. It is unreasonable to expect an expert to be the best performer across all domains and for all utterances. Different expert models may outperform others depending on domain or context, especially when they have similar capacities or sizes.

In order to simulate the variability of real-world expert ASR models, we trained ASR expert models using alternate splits of the 960-hour LibriSpeech dataset. We train three experts using two different speaker partitioning strategies. The first method randomly partitions the speakers of the training set; the second method clusters the training speakers using k-means on their audio features (speaker embeddings). The sizes of the speaker partitions created by these two schemes (Random and Clustered) are shown in Table 1. We expect the experts to perform similarly when trained on random speaker partitions, and to be more complementary when trained on clusters based on speaker similarity.

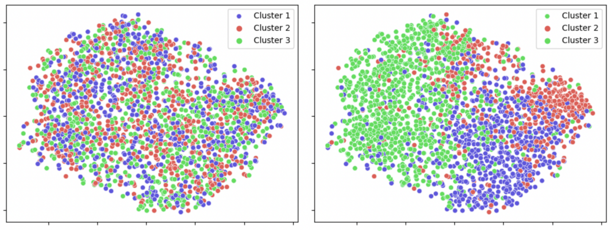

Figure 3: *Speaker cluster assignments for expert ASR training based on random assignment (left) and speaker embedding based clusters (right).*

3.2.1. ASR Model

We use the recurrent neural network-transducer (RNN-T) architecture for all expert teacher models. We believe our method generalizes to different architectures and model sizes, but leave a study beyond RNN-Ts to future work. One advantage of this choice is that all experts can be deployed on resource-constrained devices which may not be feasible if larger architectures are used.

The models have 60M parameters with a 5×1024 LSTM encoder, a 2×1024 LSTM prediction network and a feed-forward joint network with tanh activation. The input embeddings of the prediction network are 512-dimensional. We use a 2500 word-piece tokenizer. SpecAugment is used for the audio features. The audio features comprise 64-dimensional log-Mel filter-bank energy features that are computed over a 25-ms window with a 10-ms shift. The features computed on three consecutive 10-ms frames are stacked and subsampled to result in a 192-dimensional features at a 30-ms frame rate, which form the input to the ASR model. All expert and student models are trained to convergence (30-50K steps on 24 V100 GPUs) with Adam optimizer and learning rate of 10^{-5}.

3.2.2. Speaker Clustering

To partition speakers by similarity, we use a trained speaker identification model trained on LibriSpeech.[1] For each speaker and utterance, we prepare mean- and covariance-normalized features for segments of length 1 second. The speaker embedding for the utterance is the average of 10 such evaluations of the embedding model. For each speaker, we obtain 10 embeddings (for 10 different randomly chosen utterances) that we make use of in the clustering procedure described below.

We applied k-means clustering with a fixed initial state on the speaker embeddings to group them into three disjoint clusters. We use a majority vote to assign a speaker to one of the three clusters based on their 10 embeddings.[2] Figure 3 shows the speaker cluster assignments for the random and clustered partitioning, using a t-SNE [17] visualization to map the 512-dimensional speaker embeddings to two dimensions.

3.3. Student Model

Our student ASR model is an RNN-T similar to the experts, trained from scratch on unlabeled audio using the expert outputs as teacher. We employ a similar architecture for student and experts based on past findings [20] that contrast model complexity and type of students and teachers and shows that most effective training occurs when model architectures are similar. Additionally, student and teacher models can run on resource-constrained devices. However, our methods are generally applicable to settings with differing model complexities for student and teacher models.

Given a pair of audio features x and transcription t, an RNN-T model is trained by minimizing the RNN-T loss $L_{\mathrm{RNNT}}(x,t)$ that maximizes the probability of obtaining transcription t given x. The method used to combine the expert transcriptions t_i impacts student performance. The Smart-Weighter described in Section 3.1 weights the expert transcripts based on inferred accuracy. We contrast this method with two baselines:

- **Baseline 1: Best-Expert** is to train the student model using the transcription from a single best expert t^* as determined from the validation set, i.e., the student model is trained with the loss function $L(x) = L_{\mathrm{RNNT}}(x, t^*)$.
- **Baseline 2: All-Experts** is to train the student model by weighting the transcriptions from all experts equally, i.e., the loss function for each utterance is the sum of loss functions for each of the experts: $L(x) = \frac{1}{3}\sum_i L_{\mathrm{RNNT}}(x, t_i)$.

We compare these baselines and ROVER combined expert transcription with our proposed method:

- **Smart-Weighter** produces weights w_i corresponding to each of the expert transcriptions for an utterance. The weights w_i produced have low entropy, i.e., the weight for one of the experts is close to 1 and others near 0. In order to make use of all available information, we flatten this distribution by renormalizing the weights using softmax with a temperature parameter ($T = 1$), giving normalized weights $\hat{w}_i = \frac{e^{w_i/T}}{\sum_j e^{w_j/T}}$. Finally, the loss function for training the student model using the expert transcriptions weighted by the Smart-Weighter is $L(x) = \sum_i \hat{w}_i L_{\mathrm{RNNT}}(x, t_i)$.

Table 2: *WER evaluation of experts on LibriSpeech data splits. The results for the choice of "Best-Expert" are highlighted.*

	Random Expert			Clustered Expert		
	1	2	3	1	2	3
TestClean	10.91	11.81	10.82	13.66	11.27	15.94
TestOther	25.56	26.53	25.27	30.5	28.57	35.18
DevClean	10.77	11.45	**10.78**	13.65	**10.99**	16.07
DevOther	25.18	25.29	**24.27**	29.75	**27.02**	33.16
Train	9.74	9.93	9.24	12.08	8.48	16.15
Train split						
Expert-1	0.34	14.28	13.29	0.14	15.65	20.1
Expert-2	14.32	0.74	14.14	18.09	0.13	20.68
Expert-3	14.25	14.97	0.38	17.18	15.36	0.18

4. Results

4.1. Evaluating the Experts

Table 2 shows the ASR performance of the experts on various LibriSpeech data splits. We can observe overfitting on the specific data split an expert is trained on (e.g., Expert 1 on Expert 1's training split), but not on the other splits. We observe some variation in the performance of the experts obtained by speaker clustering, since, unlike for the random splits, these experts had differing numbers of utterances and speakers for training.

4.2. Student Model Evaluation

The student model is trained on 10K-hour subset of the Libri-Light dataset and evaluated on LibriSpeech train, dev and test splits. For the Best-Expert baseline we chose Expert 3 among the random experts, and Expert 2 for the clustered experts, based on the highlighted results shown in Table 2 and train a student model. The All-Experts baseline model is trained by giving all experts equal weight, as described in Section 3.3. As another baseline, we also evaluate using ROVER [5] to combine teacher transcripts.

[1] Specifically, we use the ResCNN model trained with triplet loss, as available on https://github.com/philipperemy/deep-speaker.

[2] As anecdotal validation, when using just two clusters, we obtained a partition that strongly correlated with gender annotations.

Table 3: *WER results with* **student model trained on LibriLight and experts trained on LibriSpeech**, *using different teacher configurations and training speaker partitioning methods (Random and Clustered).*

Teacher	TestClean	TestOther	Train	DevClean	DevOther
	Random				
ROVER Baseline	9.32	22.84	**7.62**	9.52	21.90
Best-Expert	10.03	23.69	11.80	9.89	22.83
All-Experts	7.79	19.93	8.56	7.72	19.46
Smart-Weighter	**7.47**	**19.37**	7.95	**7.34**	**18.85**
	Clustered				
ROVER Baseline	11.08	27.55	**9.06**	10.97	26.34
Best-Expert	9.40	24.09	11.07	9.23	23.62
All-Experts	8.55	22.33	9.81	8.42	21.53
Smart-Weighter	**8.21**	**21.22**	9.32	**7.90**	**20.86**

Table 4: *WER results with* **student and experts trained on LibriSpeech**, *using different teacher and oracle configurations and training speaker partitioning methods (Random and Clustered).*

Teacher	TestClean	TestOther	Train	DevClean	DevOther
	Random				
ROVER Baseline	9.32	22.84	7.62	9.52	21.90
Best-Expert	10.15	23.45	9.99	10.19	22.70
All-Experts	7.94	18.86	6.83	7.67	18.42
Smart-Weighter	**7.53**	**18.93**	**5.30**	**7.22**	**18.17**
Oracle	7.41	18.24	1.80	7.06	17.68
	Clustered				
ROVER Baseline	11.08	27.55	9.06	10.97	26.34
Best-Expert	9.08	23.05	7.78	8.89	21.97
All-Experts	8.33	20.92	6.95	8.16	20.35
Smart-Weighter	**7.95**	**19.85**	**5.09**	**7.70**	**19.29**
Oracle	7.08	17.88	1.82	6.83	17.48

Table 3 shows all results when training the student on LibriLight data. ROVER outperforms the best teacher model comparing with Table 6. Distilling a student on a large dataset makes it better than the corresponding teacher model (comparing Best-Expert student model to the best performing teacher). All-Experts performs better than Best-Expert and ROVER, showing the value of multiple experts. For random experts, Smart-Weighter shows an improvement of 4% and 3% compared to All-Experts, 25% and 18% compared to Best-Expert, and 20% and 15% compared to ROVER baseline on test-clean and test-other splits. For clustered experts, we see 4% and 5% improvement compared to All-Experts, 13% and 12% compared to Best-Expert, 26% and 23% on test-clean and test-other. Best-Expert is more competitive with clustered experts, possibly because the best expert also has the largest training set.

Table 4 shows results with a student model when trained on LibriSpeech data, but without using ground-truth transcriptions. Here we can also study what an "oracle" expert could achieve, i.e., choosing the expert with the most accurate output for each training utterance, giving us a bound on the performance of an expert-weighting model. With such an oracle, we see larger improvements for clustered experts than for random experts; this could be simply because clustered experts have more variation in their output quality.

Notably, for different datasets as well as different configurations of expert models, using a smart weighter produces statistically significant improvement compared to the baselines. We also note the counter-intuitive finding that the end-to-end student ASR evaluation with random experts is better than with clustered experts. We suppose this is because of the unequal cluster sizes in the clustered case. We defer the investigation of optimal cluster assignment for expert training to future work.

4.3. Smart-Weighter Evaluation

We evaluate our Smart-Weighter using two metrics. The first, **accuracy**, is based on the percentage of transcriptions selected from the best teacher, i.e., 100% implies that the Smart-Weighter always assigned highest weight to the expert with lowest WER. Another metric, **weighted WER**, uses Smart-Weighter output to compute a weighted average of WERs $\sum_{i=1}^{3} w_i \cdot \mathrm{WER}(t^*, t_i)$. The intuition behind this metric is that, as the model learns, the weights should increase for lower-WER transcriptions and weighted WER will decrease. As shown in Table 5, the Smart-Weighter performs marginally better at associating acoustic profiles with relevant experts using clustered experts than with random experts .

Table 5: *Smart-Weighter evaluation results on LibriSpeech*

	Random		Clustered	
	Accuracy	Weighted WER	Accuracy	Weighted WER
TestClean	60%	0.118	63%	0.134
TestOther	49%	0.280	59%	0.315
DevClean	61%	0.110	64%	0.130
DevOther	50%	0.254	58%	0.286

4.3.1. Smart-Weighter with ASR entropy

As a possible variant for expert weighting, we study the effect of adding a simple form of ASR confidence as side information to the W-network. This makes additional assumptions about the expert models, i.e., that n-best hypotheses and their scores are available. We compute the posterior hypothesis probabilities p_i from their score values s_i, by normalization: $p_i = \frac{s_i}{\sum_j s_j}$. The entropy is then computed as $H = -\sum_i p_i \log p_i$. We use the $n = 10$ best hypotheses.

This entropy measure is low when the 1-best hypothesis is assigned a score vastly higher than the other hypotheses, and higher when the ASR model is less confident in its best hypothesis. This entropy measure is injected into the Smart-Weighter model in the feed-forward layers before the final weights are obtained. Results seen in Table 6 show a 3 to 4% improvement in accuracy as compared to Table 5 for random experts.

Table 6: *Smart-Weighter evaluation results on LibriSpeech splits after including ASR entropy information*

	Random		Clustered	
	Accuracy	Weighted WER	Accuracy	Weighted WER
TestClean	63%	0.113	64%	0.133
TestOther	53%	0.274	60%	0.314
DevClean	64%	0.109	64%	0.130
DevOther	54%	0.244	57%	0.287

5. Conclusions

We have shown how to train student ASR models given unlabeled audio using teacher output from multiple opaque expert ASR models. An application of this framework is in continual adaptation of deployed ASR systems, using unlabeled audio and domain-specific experts. We have developed a Smart-Weighter that consumes audio features and expert transcriptions and upweights experts that are deemed to be more accurate for a given training utterance. We simulated opaque ASR experts, with or without complementarity, using multiple speaker partitioning strategies. The student model trained with weighted expert teacher transcriptions showed a 4 to 25% improvement over baselines that weight all experts uniformly, choose a single expert, or combine transcriptions with ROVER. We also observed an improvement in the Smart-Weighter by using ASR confidence or entropy as an additional feature.

Acknowledgments: We thank Anirudh Raju, Gautam Tiwari, Guruprasad Ramesh, Bach Bui, and Sri Subramaniam among many others at Alexa AI for helpful discussions.

6. References

[1] Kartik Audhkhasi, Andreas M Zavou, Panayiotis G. Georgiou, and Shrikanth S. Narayanan. Empirical link between hypothesis diversity and fusion performance in an ensemble of automatic speech recognition systems. In *Proc. Interspeech*, pages 3082–3086, 2013.

[2] Emmanuel Bengio, Pierre-Luc Bacon, Joelle Pineau, and Doina Precup. Conditional computation in neural networks for faster models. *arXiv preprint arXiv:1511.06297*, 2015.

[3] Gopinath Chennupati, Milind Rao, Gurpreet Chadha, Aaron Eakin, Anirudh Raju, Gautam Tiwari, Anit Kumar Sahu, Ariya Rastrow, Jasha Droppo, Andy Oberlin, Prahalad Venkataramanan, Zheng Wu, and Pankaj Sitpure. ILASR: privacy-preserving incremental learning for automatic speech recognition at production scale. In *Proc. 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2780–2788, 2022.

[4] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[5] Jonathan G. Fiscus. A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER). In *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354, 1997. https://github.com/usnistgov/SCTK.

[6] Yan Gao, Titouan Parcollet, and Nicholas D. Lane. Distilling knowledge from ensembles of acoustic models for joint CTC-attention end-to-end speech recognition. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pages 138–145, 2021.

[7] Alex Graves. Sequence transduction with recurrent neural networks. *arXiv preprint arXiv:1211.3711*, 2012.

[8] Robert A. Jacobs, Michael I. Jordan, Steven J. Nowlan, and Geoffrey E. Hinton. Adaptive mixtures of local experts. *Neural computation*, 3(1):79–87, 1991.

[9] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. Libri-Light: A benchmark for ASR with limited or no supervision. In *Proc. IEEE ICASSP*, pages 7669–7673, 2020.

[10] Jacob Kahn, Ann Lee, and Awni Hannun. Self-training for end-to-end speech recognition. In *Proc. IEEE ICASSP*, pages 7084–7088, 2020.

[11] Ho-Gyeong Kim, Min-Joong Lee, Hoshik Lee, Tae Gyoon Kang, Jihyun Lee, Eunho Yang, and Sung Ju Hwang. Multi-domain knowledge distillation via uncertainty-matching for end-to-end ASR models. In *Proc. Interspeech*, pages 1311–1315. International Speech Communication Association, 2021.

[12] Yuang Liu, Wei Zhang, and Jun Wang. Adaptive multi-teacher multi-level knowledge distillation. *Neurocomputing*, 415:106–113, 2020.

[13] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. LibriSpeech: an ASR corpus based on public domain audio books. In *Proc. IEEE ICASSP*, pages 5206–5210, 2015.

[14] Felipe Cruz Salinas, Kenichi Kumatani, Robert Gmyr, Linquan Liu, and Yu Shi. Knowledge distillation for mixture of experts models in speech recognition. Technical Report MSR-TR-2022-6, Microsoft Research, May 2022. https://www.microsoft.com/en-us/research/publication/knowledge-distillation-for-mixture-of-experts-models-in-speech-recognition/.

[15] Aswin Sivaraman and Minje Kim. Zero-shot personalized speech enhancement through speaker-informed model selection. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 171–175, 2021.

[16] Samuel Thomas, Michael L Seltzer, Kenneth Church, and Hynek Hermansky. Deep neural network features and semi-supervised training for low resource speech recognition. In *Proc. IEEE ICASSP*, pages 6704–6708, 2013.

[17] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(11), 2008.

[18] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in Neural Information Processing Systems*, 30, 2017.

[19] Xin Wang, Fisher Yu, Lisa Dunlap, Yi-An Ma, Ruth Wang, Azalia Mirhoseini, Trevor Darrell, and Joseph E. Gonzalez. Deep mixture of experts via shallow embedding. In *Uncertainty in artificial intelligence*, pages 552–562. PMLR, 2020.

[20] Jeremy H. M. Wong, Mark J. F. Gales, and Yu Wang. Learning between different teacher and student models in ASR. In *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, pages 93–99, 2019.

[21] Zhao You, Shulin Feng, Dan Su, and Dong Yu. SpeechMoE2: Mixture-of-experts model with improved routing. In *Proc. IEEE ICASSP*, pages 7217–7221, 2022.

[22] Fei Yuan, Linjun Shou, Jian Pei, Wutao Lin, Ming Gong, Yan Fu, and Daxin Jiang. Reinforced multi-teacher selection for knowledge distillation. In *Proc. AAAI Conference on Artificial Intelligence*, volume 35, pages 14284–14291, 2021.