SPEECHMOE2: MIXTURE-OF-EXPERTS MODEL WITH IMPROVED ROUTING

Zhao You^{*1}*, Shulin Feng*^{*1}*, Dan Su*^{1}*, Dong Yu*^{2}

^{1}Tencent AI Lab, Shenzhen, China
^{2}Tencent AI Lab, Bellevue, WA, USA
{dennisyou, shulinfeng, dansu, dyu}@tencent.com

ABSTRACT

Mixture-of-experts based acoustic models with dynamic routing mechanisms have proved promising results for speech recognition. The design principle of router architecture is important for the large model capacity and high computational efficiency. Our previous work SpeechMoE only uses local grapheme embedding to help routers to make route decisions. To further improve speech recognition performance against varying domains and accents, we propose a new router architecture which integrates additional global domain and accent embedding into router input to promote adaptability. Experimental results show that the proposed SpeechMoE2 can achieve lower character error rate (CER) with comparable parameters than SpeechMoE on both multi-domain and multi-accent task. Primarily, the proposed method provides up to 1.6% $\sim$ 4.8% relative CER improvement for the multi-domain task and 1.9% $\sim$ 17.7% relative CER improvement for the multi-accent task respectively. Besides, increasing the number of experts also achieves consistent performance improvement and keeps the computational cost constant.

***Index Terms*—** mixture of experts, router architecture, domain embedding, accent embedding, global information

1. INTRODUCTION

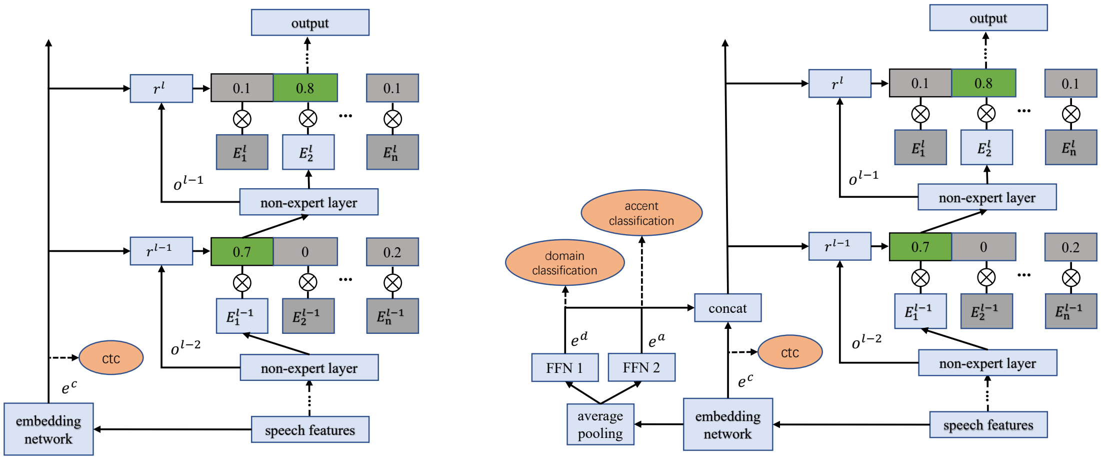
Recently, dynamic mixture-of-experts (MoE) based approaches have been investigated and applied in different tasks such as language modeling [1, 2] and image classification [3, 4, 5, 6]. Compared with traditional static models, dynamic models can be more efficient in training and inference by selectively activating model components(e.g. layers [7], channels [8], sub-networks [9]) on conditioned inputs. With this property, they are more adaptable to various inputs, and can also scale up to very large models to better perform while keeping computation cost constant.

our previous work investigated one type of dynamic model in speech recognition, named SpeechMoE [10]. It has multiple mixture-of-experts layers, using a dynamic routing mechanism to dispatch speech frames among experts networks in each layer. In SpeechMoE, a shared embedding network trained with CTC [11] criterion is used to provide a high-level representation for routers, helping routers to get a better selective effect. However, speech recognition systems need to be robust with different input conditions such as accent, recording channels and acoustic environments in real-world applications. Using the grapheme information from the embedding network for routers, SpeechMoE is not sensitive enough to handle different accents or domains inputs. Explicitly utilizing corresponding information may solve the problem. [12] has shown that incorporating global dialect-specific information(e.g., one-hot vector of the target dialect) is effective in modeling dialect variations. [13] adds global domain information to the end-to-end models and shows that feeding the domain-id improves the model performance.

In this study, we continue to explore a mixture-of-experts approach for speech recognition, augmenting SpeechMoE with extra global knowledge to deal with varying domains and accents. The previous SpeechMoE only provides frame-level grapheme embedding for routers. Here, we propose a new router architecture which takes extra domain embedding and accent embedding as router input. To constrain the computation cost of the embedding network, we use multi-task training for the original embedding network to simultaneously produce grapheme, domain, and accent embedding. We hypothesize that explicitly providing domain and accent embedding for routers should help improve the route decisions, making the experts specialized in processing distinct input utterances. Besides, we use FastMoE [14] in our system to support scaling up the MoE model size by training across multiple GPUs on multiple nodes. With comparable computation costs, more experts also help achieve better performance.

The rest of the paper is organized as follows. Section 2 reviews the previous work of SpeechMoE, and Section 3 presents our proposed method SpeechMoE2. The experimental setup is described in Section 4, and the experimental results are reported in Section 5. Finally, we conclude this paper in Section 6.

[*] Equal contribution.



(a) (b)

Fig. 1. (a) and (b) illustrate the architecture of SpeechMoE and SpeechMoE2 respectively. In SpeechMoE2, multi-task training is conducted on the embedding network to learning grapheme embedding, domain embedding, and accent embedding simultaneously, which will be concatenated as extra knowledge to feed routers.

2. PREVIOUS WORKS

In this section, we mainly describe our previous work of SpeechMoE.

2.1. SpeechMoE

The SpeechMoE is proposed in [10], and it can achieve higher prediction accuracy than traditional static models with comparable computational costs of inference.

The architecture of SpeechMoE is described in Fig.1(a), which comprises multiple MoE layers, non-expert layers and a shared embedding network. Each MoE layer consists of n experts and a router layer. It takes the output of the previous layer and the shared embedding as input and routes each speech frame to the top-1 expert with the largest route probability. Let W_r^l, e^c and o^{l-1} be the router weights of the l-th layer, shared embedding and the output of the previous layer, then the router probability can be defined as follows:

$$r^l = W_r^l \cdot Concat(e^c; o^{l-1}) \tag{1}$$

$$p_i^l = \frac{exp^{r_i^l}}{\sum_{j=1}^{n} exp^{r_j^l}} \tag{2}$$

Then, the selected expert's output is also gated by router probability to get the output of the MoE layer,

$$y^l = p_i^l E_i^l \tag{3}$$

Since only one expert is active in each layer, the SpeechMoE can keep the computational cost constant while scaling up to a very large model. To achieve better sparsity and balance among different experts, the sparsity L1 loss L_s and mean importance loss L_m are added into the loss function:

$$L_s = \frac{1}{k}\sum_{j=1}^{k} \parallel \hat{p}_j \parallel_1 \tag{4}$$

$$L_m = n\sum_{i=1}^{n}(\frac{1}{k}\sum_{j=1}^{k} p_{ij})^2 \tag{5}$$

where $\hat{p}_j$ stands for the unit-normalized router probability distribution of frame j, and k is the number of frames in this mini-batch. p_{ij} stands for the router probability on expert i of frame j.

The loss function for training SpeechMoE is defined as:

$$L(x;y) = L_c(x;y) + \alpha L_s(x) + \beta L_m(x) + \gamma L_e(x;y) \tag{6}$$

where x and y are the input speech features and target label, respectively. L_c is the CTC loss for speech recognition, while L_s and L_m are the mentioned sparsity $L1$ loss and mean importance loss, used to encourage sparsity and diversity of the model. The embedding loss L_e is also CTC loss. It shares the same goal with our SpeechMoE model and provides reliable embedding for the routers. α, β, and γ are the scale for L_s, L_m and L_e respectively.

3. SPEECHMOE2

3.1. Model structure

Fig.1(b) illustrates the architecture of the proposed SpeechMoE2. Its main structure is the same as the SpeechMoE while the difference is that the embedding network is revised to output domain and accent embedding besides grapheme embedding. In order not to significantly increase the computation

Table 1. *Configurations for different types of test sets*

Data type	vp	rir	aec	noise
CL	×	×	×	×
NVR	×	✓	×	×
VR	✓	✓	×	×
VRA	✓	✓	✓	×
VRN	✓	✓	×	✓
VRNA	✓	✓	✓	✓

cost, we simply add an average pooling layer and two feed-forward layers over the embedding network to do multi-task training on domain and accent classification. Then, embedding from three aspects of the domain, accent and CTC will be concatenated with the previous layer's output as the input of routers, which can be defined as:

$$r^l = W_r^l \cdot Concat(e^c; e^a; e^d; o^{l-1}) \tag{7}$$

$$e^a = W_a \cdot \frac{1}{T} \sum_{t=1}^{T} e_t^c \tag{8}$$

$$e^d = W_d \cdot \frac{1}{T} \sum_{t=1}^{T} e_t^c \tag{9}$$

Where W_a and W_d represent projection weight for accent and domain respectively. T is the total frames number for a certain utterance.

Given the input x, grapheme target y, domain target y_d and accent target y_a, the full loss function of our method is defined as

$$\begin{aligned} L(x;y) &= L_c(x;y) + \alpha L_s(x) + \beta L_m(x) \\ &+ \gamma L_e(x;y) + \eta L_a(x;y_a) + \theta L_d(x;y_d) \end{aligned} \tag{10}$$

Among these items, L_c, L_s, L_m and L_e are the same as mentioned above. L_a and L_d are the accent and domain classification loss. α, β, γ, η, and θ are the scale for L_s, L_m, L_e, L_a and L_d respectively.

3.2. Toward Large models

We use FastMoE [14] to support scaling up the model size by training across multiple GPUs on multiple nodes, where experts are placed on different devices while non-expert layers and routers are copied. Different workers process different batches of data and speech frames in them will communicate across experts by the route decisions. We have introduced the mentioned L_s and L_m loss to make the router probabilities sparse and diverse. Since large batch size will lead to large route decision space, we collect all router probabilities from different workers and compute the complete loss and gradients, which would be equivalent to dealing with a larger data batch and may help improve model performance.

Table 2. *Total length (approx.hours) of each domain for training and testing (k for thousand).*

Model	CL	NVR	VR	VRA	VRN	VRNA
Train	89k	7k	30k	8k	8k	8k
Test	2.1	2.1	2.1	2.1	2.1	2.1

Table 3. *Total length (approx.hours) of each accent (k for thousand).*

Accent	sichuan	hunan	fujian	guangdong
Train	3.2k	4.5k	4.3k	4.5k
Test	4.1	4.3	4.4	4.0

4. EXPERIMENTAL SETUP

4.1. Datasets

Our training corpus is mixed data sets collected from several different application domains and accents. It comes to a 150k hours training corpus. In this work, we use the term 'domain' to mean a logical group of utterances that share similar characteristics such as background noise.

To evaluate the performance of augmenting with domain embedding, we have simulated six types of domain test sets. They are constructed with different mixing conditions like voice processing(vp), acoustic echo cancellation(aec), reverberation(rir) and signal-to-noise-ratio(SNR), etc. The SNR is set between 15 and 30 dB, while the reverberation time is set between 0 and 900 milliseconds. We refer these test sets as clean(CL), nonvp-rir(NVR), vp-rir(VR), vp-rir-noise(VRN), vp-rir-aec(VRA), vp-rir-aec-noise(VRAN) respectively. The detailed configurations for them are shown in Table 1 and the amount of domain-specific training and test data can be found in Table 2.

The training utterances include four different accents, namely sichuan, hunan, fujian and guangdong. The amount of training and test data is shown in Table 3.

4.2. Training setup

All the experiments use 40-dimension log-mel filterbank features appended with the first-order and the second-order derivatives. Log-mel filterbank features are computed with a 25ms window and shifted every 10ms. These features are stacked with 8 consecutive frames and downsampled to a 30ms frame rate. Each frame uses a global mean and variance normalization. All networks are trained with the CTC criterion. We use the context-independent-syllable-based acoustic modeling method [15] for CTC learning. The target labels of CTC learning are defined to include 1394 Mandarin syllables, 39 English phones, and a blank. Character error rate results are measured on the test sets, and the floating-point operations (FLOPs) for a one-second example are used to evaluate the computational cost of inference. We use a pruned, first

Table 4. *Results of augmenting with domain embedding.*

Model	Params	FLOPs	Test set					
			CL	NVR	VR	VRA	VRN	VRNA
MoE1	105M	2.3B	14.09	17.64	18.22	26.68	22.43	33.67
MoE2-2e	107M	2.3B	**13.55**	**17.34**	**17.87**	**26.07**	**21.36**	**33.13**

Table 5. *Results of augmenting with accent embedding.*

Model	Params	FLOPs	Test set			
			sichuan	guangdong	fujian	hunan
MoE1	105M	2.3B	12.81	7.99	6.59	34.25
MoE2-2e	107M	2.3B	**10.54**	**7.84**	**6.39**	**31.00**

pass, 5-gram language model for decoding. All the systems use a vocabulary that consists of millions of words. Decoding is performed with a beam search algorithm by using the weighted finite-state transducers (WFSTs).

4.3. Acoustic Model

The backbone of our model consists of 30 MoE layers, 30 sequential memory layers and 3 self-attention layers. Each MoE layer is followed by one sequential memory layer, and a self-attention layer is inserted after each 10 consecutive MoE and sequential memory layers. The detailed configuration of each layer is consistent with [10]. In our experiments, we vary the number of experts of MoE layers to be 2, 4 and 16, which are marked as MoE-2e, MoE-4e and MoE-16e respectively. The shared embedding network is a static model without MoE layers but a similar structure to the backbone. Additionally, accent and domain projection layers are added to the shared embedding network. For the accent and domain projection layers, we set the embedding dimension to be 512.

In our study, we built the baseline SpeechMoE system (MoE1) for evaluating the performance of our proposed method. The baseline model has 2 experts in each MoE layer, not using the domain and accent embedding. For all experiments on MoE models, we set the hyper-parameters $\alpha = 0.05$, $\beta = 0.05$, $\gamma = 0.01$, $\eta = 0.1$ and $\theta = 0.1$.

5. EXPERIMENTAL RESULTS

5.1. Augmenting with extra knowledge

In this section, we evaluate the performance of augmenting with domain and accent embedding. As shown in Table 4, the MoE2-2e, which uses extra domain embedding and accent embedding, achieves lower character error rate than the baseline model for the multi-domain task. Similar results can be seen in Table 5, MoE2-2e can also achieve lower character error than the baseline model for the multi-accent task.

The experimental results indicate that providing a high-level representation for routers to make them aware of the varying domains and accents would be helpful to get a better route strategy and improve the performance. But, more importantly, the proposed architecture may prove to be a flexible framework to integrate diverse global information to adapt different tasks.

Table 6. *Results of increasing the number of experts for multi-domain task.*

Model	Params	FLOPs	Test set					
			CL	NVR	VR	VRA	VRN	VRNA
MoE2-2e	107M	2.3B	13.55	17.34	17.87	26.07	21.36	33.13
MoE2-4e	171M	2.3B	13.21	17.07	17.41	25.41	21.05	32.54
MoE2-16e	500M	2.3B	**12.79**	**16.82**	**16.96**	**24.45**	**20.12**	**31.46**

Table 7. *Results of increasing the number of experts for multi-accent task.*

Model	Params	FLOPs	Test set			
			sichuan	guangdong	fujian	hunan
MoE2-2e	107M	2.3B	10.54	7.84	6.39	31.00
MoE2-4e	171M	2.3B	9.16	7.73	6.31	29.63
MoE2-16e	500M	2.3B	**8.92**	**7.59**	**6.24**	**28.49**

5.2. Increasing the number of experts

In this section, we investigate the effect of increasing the number of experts. Table 6 and Table 7 show the performance of SpeechMoE2 models with different number of experts on multi-domain and multi-accent tasks. The results show that the performance of SpeechMoE2 gets better consistently as the number of experts increases. From 2 experts to 16 experts, the relative gain is averaged 5.1% for domain-specific test sets and 7.7% for accent-specific test sets. Specially, as we increase the number of experts, our models keep the FLOPs per frame constant.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we extend our previous work to improve the performance of multi-domain and multi-accent speech recognition. The improvement comes from the new router architecture, which takes extra domain embedding and accent embedding as input. The idea is intuitive that providing extra knowledge can make the router more specialized for varying input utterances and experimental results prove the effectiveness. For multi-domain task, the proposed method provides up to 1.6% $\sim$ 4.8% relative CER improvement. For the multi-accent task, the relative gain is between 1.9% $\sim$ 17.7%. Besides, increasing the number of experts also achieves consistent performance improvement. In the future, we plan to explore new MoE layer with more complicated model such as conformer.

7. REFERENCES

[1] Dmitry Lepikhin, HyoukJoong Lee, Yuanzhong Xu, Dehao Chen, Orhan Firat, Yanping Huang, Maxim Krikun, Noam Shazeer, and Zhifeng Chen, “Gshard: Scaling giant models with conditional computation and automatic sharding,” *arXiv preprint arXiv:2006.16668*, 2020.

[2] William Fedus, Barret Zoph, and Noam Shazeer, “Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity,” *arXiv preprint arXiv:2101.03961*, 2021.

[3] Sam Gross, Marc’Aurelio Ranzato, and Arthur Szlam, “Hard mixtures of experts for large scale weakly supervised vision,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6865–6873.

[4] Karim Ahmed, Mohammad Haris Baig, and Lorenzo Torresani, “Network of experts for large-scale image categorization,” in *European Conference on Computer Vision*. Springer, 2016, pp. 516–532.

[5] 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*. PMLR, 2020, pp. 552–562.

[6] Shaofeng Cai, Yao Shu, and Wei Wang, “Dynamic routing networks,” in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2021, pp. 3588–3597.

[7] Gao Huang, Danlu Chen, Tianhong Li, Felix Wu, Laurens van der Maaten, and Kilian Q Weinberger, “Multi-scale dense networks for resource efficient image classification,” *arXiv preprint arXiv:1703.09844*, 2017.

[8] Ji Lin, Yongming Rao, Jiwen Lu, and Jie Zhou, “Runtime neural pruning,” in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 2178–2188.

[9] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean, “Outrageously large neural networks: The sparsely-gated mixture-of-experts layer,” *arXiv preprint arXiv:1701.06538*, 2017.

[10] Zhao You, Shulin Feng, Dan Su, and Dong Yu, “Speechmoe: Scaling to large acoustic models with dynamic routing mixture of experts,” *arXiv preprint arXiv:2105.03036*, 2021.

[11] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

[12] Bo Li, Tara N Sainath, Khe Chai Sim, Michiel Bacchiani, Eugene Weinstein, Patrick Nguyen, Zhifeng Chen, Yanghui Wu, and Kanishka Rao, “Multi-dialect speech recognition with a single sequence-to-sequence model,” in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 4749–4753.

[13] Tara N Sainath, Yanzhang He, Bo Li, Arun Narayanan, Ruoming Pang, Antoine Bruguier, Shuo-yiin Chang, Wei Li, Raziel Alvarez, Zhifeng Chen, et al., “A streaming on-device end-to-end model surpassing server-side conventional model quality and latency,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6059–6063.

[14] Jiaao He, Jiezhong Qiu, Aohan Zeng, Zhilin Yang, Jidong Zhai, and Jie Tang, “Fastmoe: A fast mixture-of-expert training system,” *arXiv preprint arXiv:2103.13262*, 2021.

[15] Zhongdi Qu, Parisa Haghani, Eugene Weinstein, and Pedro Moreno, “Syllable-based acoustic modeling with ctc-smbr-lstm,” in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 173–177.