SRI-B ASR System Description for OpenASR21 Challenge

Team name: SRIB_ASR
Industry name: Samsung R&D Institute (SRI), Bangalore, India

November 2021

1 Experimental Setups

1.1 Data augmentations for constrained case

- All wave files are downsampled to 8 kHz.
- Speed perturbations were applied with factors 0.9 and 1.1 for experiments in kaldi and Espnet for constrained case.
- For Espnet experiments, SpecAugment [1] is also applied.

1.2 Data augmentations for unconstrained case

- We only train Espnet models for unconstrained case after actual challenge deadline.
- Publicly available external speech and text data are used for unconstrained case as shown in Table 1.
- Microsoft Research (MSR) Tamil dataset consists of 40 hours of training and 5 hours of dev set.
- IITM Tamil datasets consists of 120 hours of training and 5 hours of dev set.
- AI4Bharat Tamil text corpus has 0.6 million utterances

Table 1: Details of publicaly available datasets

Dataset	Resource	Link
MSR	Audio, Text	https://github.com/Open-Speech-EkStep/ULCA-asr-dataset-corpus
IITM	Audio, Text	https://github.com/Open-Speech-EkStep/ULCA-asr-dataset-corpus
AI4Bharat	Text	https://indicnlp.ai4bharat.org/corpora/#downloads

1.3 Audio segmentation

- For training and development sets, segments and STM files are created using transcription files provided with the dataset.
- For evaluation set, Kaldi Voice Activity Detection (VAD) is applied to segment the evaluation audio data set.

1.4 Feature extraction

- GMM-HMM systems are trained using 39-D Mel Frequency Cepstral Coefficients (MFCC) including delta features.
- Neural networks in kaldi are trained using 40-D high resolution MFCC features (without delta features).
- 83-D Mel filterbank features with pitch are using for Espnet transformer models.

2 Experiments with Hybrid ASR

2.1 Kaldi Chain Models

Hybrid DNN-HMM systems using various TDNN-based models are trained using Kaldi framework. Forced-aligned labels are created using triphone-based GMM-HMM systems with 2750 senone labels. The details Kaldi hybrid systems are shown in Table 2. All the hybrid systems are trained using LF-MMI criteria with CE loss and MMI loss as multitasking system.

Table 2: Kaldi TDNN-based system details

Sr. No.	Hidden Layers	Hidden Units
1	6 TDNN	512
2	6 TDNN, 3 LSTM	512,320
3	2 TDNN, 4 CNN	512, (40, 20)

2.2 Mixture of Experts (MoE) Models

We employ a Mixture of Experts (MoE) based architecture, referred as *MixNet*, for acoustic modeling. This work is based on our earlier proposed work using MoE for accented ASR [2]. MoEs are essentially region-dependent processing of the features using an ensemble of experts, which could be either classifiers or regressors Different experts are specialized to operate on specific regions in the input space. Outputs of the experts are linearly combined using data dependent weights generated by an additional auxiliary classifier. The role of this classifier is to "select" (soft or hard) the best expert that is akin to the location of the feature vector in the input space [2].

The block diagram of MixNet is shown in Figure 1.

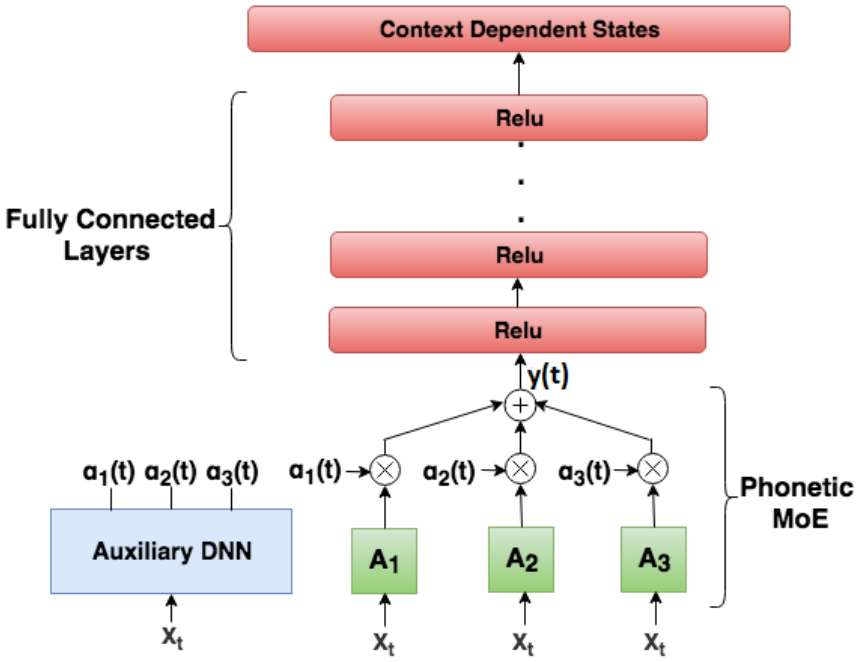

Figure 1: Schematic diagram of MixNet: Mixture of Experts based Acoustic Model

The experts, denoted by $\mathbf{A}_i$ matrices in the Figure, are learned on top of input features belonging to different acoustic regions. The regions may be predefined to be broad phonetic classes. An auxiliary classifier is trained to classify input features into these classes. The outputs from the experts are then combined linearly using posterior probabilities generated by a classifier as weights. Mathematically,

$\mathrm{y(t)} = \sum_{i=1}^{C_p} \alpha_i(t)\mathbf{A}(i)\mathbf{x}(t)$

where C_p is the number of the broad phonetic classes, $\mathbf{x}_t$ and $\mathbf{y}_t$ are the input and output features of the MoE network. $\alpha_i(t)$ is the posterior probability (or the gating signal) of class i pertaining to frame at time t, which is generated by the classifier.

We apply Mixture of Experts on different domains. With the available information with the data, we employ MoE on gender, environment and dialect.

- **Gender-based Mixture of Experts**: We apply Mixture of Experts to account for gender variability in the data. One linear layer is used for genders, male and female, which are then combined using the output of an auxiliary gender classifier. We call this *MoE-2Gen*.
- **Environment-based Mixture of Experts**: On similar idea, we also try

MoE to account for variations in the environment where each environment type has its own smaller layer. We use 6 environemnt types available in the data. The outputs of which are combined using an auxiliary environment classifer. We call this *MoE-7Env*.

- **Dialect-based Mixture of Experts**: Similarly, to handle variations in dialects, we use separate layers to process individual dialects whose outputs are combined using the weights of the auxiliary dialect classifier. We have 5 dialects in our data and the classifier is trained on those. We call this *MoE-5Dia*.

3 Experiments with End-to-end ASR

Transformer [3] based encoder-decoder architecture is utilized for all of our experiments. Hybrid CTC and Attention training and decoding method is applied as introduced in [4] with a CTC weight of 0.3. Transformer model has 12 encoder layers and 6 decoder layers. Both encoder and decoder layers consist of 2048 hidden units. The output targets are subwords units generated by training a unigram language model on the text using SentencePiece model. For constrained category, 200 subwords are used and for unconstrained case 1000 subword units are used.

Decoding is performed using the beam search algorithm with a beam size of 60 and a CTC weight of 0.4. We do not use any language model for rescoring. All the ASR models are trained using ESPNet [5] toolkit and we follow the default Librispeech recipe. For all the experiments, 80-dim Mel filterbank features are used with window size as 512 and hop length of 256 samples. Model is trained using Adam optimizer [6] and a warmup learning rate schedule with 25000 warmup steps. The SpecAugment [1] data augmentation is applied by default during training time.

All the E2E ASR experiments were performed using 2 GPUs and it took around 12 hours to complete constrained experiment and 3 days 12 hours for unconstrained case. Decoding 10 hours of dev set with single GPU takes around 5 hours due to large beam size.

For unconstrained experiments, we also trained large LSTM-LM model using external Tamil text data. LSTM-LM has 2 hidden layers with 2048 hidden units. During decoding with LM, weight of 0.3 is used along with CTC and attention model weights.

4 Experimental Results

4.1 Development set results

- Experimental results for the dev set are shown in Table 3 for TDNN, TDNN-LSTM and TDNN-CNN models in Kaldi.

- Best results are achieved using TDNN model alone compared to adding LSTM and CNN layers.
- System combination using MBR for theses models results in significant reduction in WER 72.73 % compared to best TDNN model with WER 76.59 %.
- The experimental results using MoE model with TDNN layers are also shown in Table 3. Due to low resource data, MoE with 2 classes as gender performed well compared to 5 and 7 classes in MoE framework, respectively.
- MBR system combination with this 2 class MoE modeol and 3 TDNN-based systems described above gave WER of 72.25 % that is slightly better than TDNN system combination.

Table 3: Experimental results on dev set.

Sr. No.	Model	Dev Set
1	TDNN	76.59
2	TDNN-LSTM	77.36
3	TDNN-CNN	76.63
4	MBR comb (1+2+3)	72.73
5	MoE-7Env (7 classes)	78.55
6	MoE-5Dia (5 classes)	78.48
7	MoE-2Gen (2 classes)	77.57
8	MBR comb (1+2+3+7)	**72.25**

4.2 Evaluation set results

- Two Kaldi systems, TDNN and 2 class MoE were submitted on evaluation data during challenge submission period.
- Best result of 79.41 % are obtained using system combination approach, however, it was after submission portal closes.

Table 4: Experimental results on dev set.

Sr. No.	Model	Eval Set
1	TDNN	82.52
2	MBR comb (3 TDNN system+ 1 MoE system)	79.41

4.3 Results after Challenge Deadline

- The end-to-end transformer model did not perform well due to very small amount of data. There was a huge overfitting issue while training this model even after speed perturbation and specaugment activated during training.
- Currently we are exploring E2E systems for possible improvements in unconstrained case. As mentioned before, we are using two additional Tamil datasets to improve the performance of acoustic model. NIST scores obtained for dev and eval set for this unconstrained case are shown in Table 5. From the experimental results we can see that there is huge gap between dev and eval set errors which may be due to difference in speakers, acoustic environment and possible OOVs.

Table 5: Experimental results using end-to-end ASR.

Sr. No.	Model	Condition	Dev	Eval
1	Transformer	Constrained	89.5	-
2	Transformer+LSTM-LM	Unconstrained	64.70	77.24

References

[1] 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," in *Proc. Interspeech 2019*, 2019, pp. 2613–2617.

[2] Abhinav Jain, Vishwanath P. Singh, and Shakti P. Rath, "A Multi-Accent Acoustic Model Using Mixture of Experts for Speech Recognition," in *Proc. Interspeech 2019*, 2019, pp. 779–783.

[3] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. 2017, vol. 30, Curran Associates, Inc.

[4] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[5] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew

Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai, "ESPnet: End-to-end speech processing toolkit," in *Proceedings of Interspeech*, 2018, pp. 2207–2211.

[6] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Yoshua Bengio and Yann LeCun, Eds., 2015.