Context Shuffling and Lexicographic Sorting for Marathi Accented Speech Recognition

Vinit Unni[1], *Preethi Jyothi*[1]

[1]Indian Institute of Technology Bombay

vinit@cse.iitb.ac.in, pjyothi@cse.iitb.ac.in

Abstract

Automatic speech recognition (ASR), powered by deep neural networks, has made large strides in the recent past. This has largely been aided by increased access to compute power as well as data. However, this abundance of data is skewed towards only a few languages. Even within the languages, one sees a bias towards particular accents or other demographic artefacts. This invariably leads to trained models that are biased towards such skews in the datasets. In this paper, we propose two training curricula that take advantage of redundancy in the training data and encourage learning representations that are more robust to varying accents. We show significant improvements on a Marathi ASR task consisting of speakers from different demographics and accents.

Index Terms: accented speech recognition, curriculum, Marathi ASR

1. Introduction

Recent state-of-the-art ASR systems employ end-to-end architectures [1] such as encoder-decoder models [2], CTC and RNN-Transducer alignment-based models [3, 4], Transformer-based models [5] or a hybrid combination of attention and CTC-based models [6]. In datasets containing speech samples from speakers with different accents, these models do not explicitly exploit redundancies in text when different speakers are speaking the same words or sentences. This is our main question of interest: Are there any training modifications we can adopt that exploit redundancies in text across speakers to (hopefully) learn more accent-invariant representations and improve recognition performance on speakers from all the training accents?

In this work, we focus on a specific setting where we assume that the training data consists of speech samples from multiple accents or demographics corresponding to the same underlying text. We propose two simple training tricks, *context shuffling* and *lexicographic sorting*, that help learn representations that are most robust to varying speaker characteristics such as accents.

2. Related Work

There has been a lot of prior work on accented speech recognition. In early work [7], synthetic data was generated by modifying the pronunciation models to include various pronunciations of vowels. In [8], a combination of speaker adaptation techniques (like maximum likelihood linear regression, MLLR) was used to improve performance across accents. [9] used i-vectors [10] and an accent-dependent layer while keeping rest of the network same. In [11], a different top layer for different accents was explored. Model interpolation methods were explored in [12]. [13] showed adaptive acoustic models that are conditioned on the fly using dialect representations.

Multi-task architectures have also been explored for accented ASR. In [14], two parallel acoustic models for different accents are trained which share a common feature extractor. In [15], an Accent Identification (AID) task is used as an auxiliary loss in a multi-task architecture. [16] explored the use of a mixture of feature extractors (Mixture of Experts, MoE) to generate acoustic models where each individual extractor focuses on a particular phone or accent class. In [17], a hierarchical multi-task approach is used where a phoneme CTC loss is used in the lower layers along with a grapheme CTC at the top layer. As a variant of multi-task, [18] uses an adversarial training method to extract features which aid in transcription but do not encode information to predict accents.

[19] demonstrated how large error rate reductions can be obtained by fine-tuning only the initial encoder layers. [20] showed that a multi-task architecture with accent labels/embeddings fed back into the decoder layer also provide improvement in accented speech recognition. In [21], a teacher-student method was explored where a multi-accent teacher model is distilled into individual single-accent models. In [22], accented speech recognition is modelled as a Model Agnostic Meta Learning (MAML) problem where each accent is treated as a different task. In [23], AID task was explored not only to augment the feature vector, but also as an output in the same task. It was observed that delaying *AID* to the end of an utterance was better than attempting to incorporate it within the feature representation.

3. Methodology

3.1. Coupled Loss

We use a hybrid CTC-attention based model as our base ASR end-to-end model [6]. The hybrid model optimizes a linear combination of an attention-based loss ($\mathcal{L}_{\text{att}}$) and a CTC-based loss ($\mathcal{L}_{\text{ctc}}$):

$$\mathcal{L}_{\text{hyb}} = \beta\mathcal{L}_{\text{att}} + (1-\beta)\mathcal{L}_{\text{ctc}} \quad (1)$$

where $\beta \in [0,1]$ is a scaling hyperparameter. Consider an encoder network within the hybrid model that transforms the input $\mathbf{x}$ into a sequence of hidden representations $\mathbf{h} = \{h_1, \ldots, h_K\}$. The attention-based loss, $\mathcal{L}_{\text{att}}$, is defined using an attention distribution $\{\alpha_{i1}, \ldots, \alpha_{iK}\}$ that linearly interpolates $\{h_1, \ldots, h_K\}$ to form a context vector c_i at the i^{th}decoder time-step:

$$c_i = \sum_{j=1}^{K} \alpha_{ij} h_j$$

$\mathcal{L}_{\text{att}}$ is also defined using decoder states that are estimated by conditioning on the context vectors, along with the decoder states and predictions at the previous time-step.

To define the coupled loss [24], we assume we have pairs of utterances $\mathbf{x}$ and $\mathbf{x}'$ that map to the same output word sequence $\mathbf{y}$. Let the context vectors corresponding to $\mathbf{x}$ and $\mathbf{x}'$ at each decoder time-step i be c_i and c'_i, respectively. Since the underlying text corresponding to $\mathbf{x}$ and $\mathbf{x}'$ are identical, we hypothesize that the context vectors at each decoder time-step should be close to one another via a regularization loss,

$$\mathcal{L}_{\text{coup}} = \frac{1}{K}\sum_i \text{dist}(c_i - c'_i) \tag{2}$$

which we refer to as a *coupled loss* ($\mathcal{L}_{\text{coup}}$). We define dist to be $1-\text{cosine}(c_i, c'_i)$ where cosine is the cosine similarity between the two vectors. This can be added as an additional (scaled) loss term to $\mathcal{L}_{\text{att}}$ in Eqn.(1) with a scaling parameter λ.

3.2. Context Shuffling

In *context shuffling*, instead of imposing a distance metric on the context vectors as in coupled loss, we delegate the task of making the context vectors more accent invariant to the network itself. We achieve this by merely swapping context vectors at a decoder time-step between two utterances that map to the same output sequence.

Similar to the setup for coupled loss, we consider training batches consisting of pairs of inputs $\mathbf{x}$ and $\mathbf{x}'$ which map to the same output $\mathbf{y}$ (similar to the setup for coupled loss). The context vectors generated by the attention layer for these inputs are denoted by $\mathbf{c} = \{c_1, c_2, \ldots, c_N\}$ and $\mathbf{c}' = \{c'_1, c'_2, \ldots, c'_N\}$, respectively. (Note that the lengths of $\mathbf{c}$ and $\mathbf{c}'$ are the same since they correspond to the same underlying text.) With a swapping probability set as a hyperparameter, we swap c_i with c'_i within the two context vector sequences.

That is, the context vectors before swapping would look like:

$$\mathbf{c} = \{c_1, c_2, ...c_N\}$$
$$\mathbf{c}' = \{c'_1, c'_2, ...c'_N\}$$

After shuffling the context vectors with a swapping probability of $1-\eta$, the context vectors might look like:

$$\mathbf{c}_{CSF} = \{c_1, c'_2, c_3, c_4, c'_5 ... c_N\}$$
$$\mathbf{c}'_{CSF} = \{c'_1, c_2, c'_3, c'_4, c_5 ... c'_N\}$$

Such a swapping would have the effect of bringing c_i and c'_i closer together and potentially strengthening the decoder to be robust enough to such variations in the context vectors. An illustration of the architecture is shown in Fig 1.

3.3. Lexicographic Curriculum with Context Shuffling

Rather than use context shuffling with pairs of inputs, we extend the idea further by creating batches that contain utterances with a high degree of overlapping text and shuffle context vectors across utterances within a batch. We first lexicographically sort the transcriptions and select random batches of contiguous samples from the sorted list.

Extending the above idea further, we create batches such that each batch contains as many inputs as possible mapping to the same output. This is done in our dataset by lexicographically sorting our outputs and selecting random batches of contiguous samples from the same. We consider context vectors corresponding to overlapping Ngrams within the text. We also set a left context a and a right context b such that $a + b = N - 1$. Let c_i^j be the i^{th} context vector of the j^{th} sample in a batch that maps to an output label denoted by y_i^j. We define a function $f(c_i^j) = \mathbf{w}$ that maps c_i^j to an N-gram $\mathbf{w}$ containing y_i^j, along with an additional a labels to the left and b labels to the right. To illustrate, if we have an output $y = \{y_1, \ldots, y_k\}$ with context vectors $c = \{c_1, \ldots, c_k\}$, then $f(c_i) = \{y_{j-a}, \ldots, y_{j-1}, y_j, y_{j+1}, \ldots, y_{j+b}\}$.

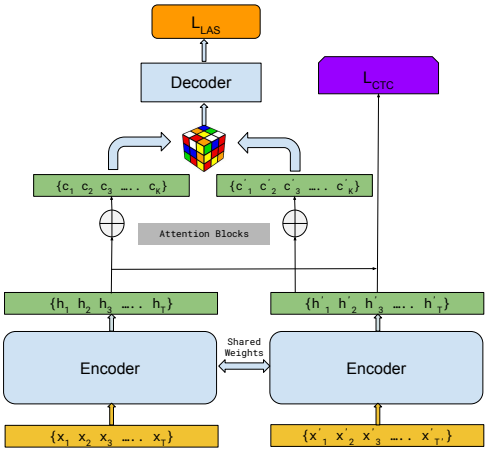

Figure 1: *Context Shuffling in a hybrid* CTC/LAS *model for paired inputs.*

For every N-gram in our batch, we maintain a list of the context vectors mapping to the same N-gram across all the examples in the batch. Since our batches are sorted lexicographically, and given the redundancy in our dataset with respect to the output text, we always have more than one context vector mapping to the same N-gram. With a fixed probability $1-\eta$, we overwrite the original context vector c_i^j with one of the context vectors in the list that matches in the underlying N-gram.

To illustrate, assume that a batch has four utterances where the first three map to the same underlying text. Let us denote the text strings as:

$$\mathbf{y}^1 = \{p, q, c, d, e, f\}$$
$$\mathbf{y}^2 = \{p, q, c, d, e, f\}$$
$$\mathbf{y}^3 = \{p, q, c, d, e, f\}$$
$$\mathbf{y}^4 = \{c, d, z, p, q\}$$

where the symbols p, q, etc. correspond to N-grams. Let the corresponding context vectors be:

$$\mathbf{c}^1 = \{c_1^1, c_2^1, c_3^1, c_4^1, c_5^1, c_6^1\}$$
$$\mathbf{c}^2 = \{c_1^2, c_2^2, c_3^2, c_4^2, c_5^2, c_6^2\}$$
$$\mathbf{c}^3 = \{c_1^3, c_2^3, c_3^3, c_4^3, c_5^3, c_6^3\}$$
$$\mathbf{c}^4 = \{c_1^4, c_2^4, c_3^4, c_4^4, c_5^4\}$$

where $\mathbf{c}^k$ denotes the context vectors for utterance k. Let $a = 1$ and $b = 0$, which means that we are considering bigrams. Although the first 3 sentences are dissimilar from the 4$^{\text{th}}$ sentence, the 1$^{\text{st}}$ bigram of the first three sentences map to the 4$^{\text{th}}$ bigram of the 4$^{\text{th}}$ sentence based on an N-gram similarity. Similarly, the 3$^{\text{rd}}$ bigram of the first three utterances maps with the 1$^{\text{st}}$ bigram of the 4$^{\text{th}}$ sentence. Thus, after context shuffling, the resulting context vectors could become:

$$\mathbf{c}^1_{CSF} = \{c_1^2, c_2^3, c_3^3, c_2^4, c_5^1, c_6^3\}$$
$$\mathbf{c}^2_{CSF} = \{c_1^3, c_2^1, c_3^2, c_4^3, c_5^1, c_6^2\}$$
$$\mathbf{c}^3_{CSF} = \{c_1^3, c_2^2, c_3^1, c_2^4, c_5^3, c_6^1\}$$
$$\mathbf{c}^4_{CSF} = \{c_1^4, c_2^4, c_3^4, c_4^4, c_2^1\}$$

Data-Split	#utts	#sent.	#spks	Duration(hrs)
College-Train	20587	2497	7	22.89
College-Dev	88	86	1	0.255
College-Test	290	284	1	0.098
Urban-Train	26394	2497	10	34.34
Urban-Dev	71	59	1	0.060
Urban-Test	330	283	1	0.382
Rural-Train	18029	2497	8	25.253
Rural-Dev	87	85	1	0.286
Rural-Test	291	284	1	0.130

Table 1: *Statistics of the Marathi datasets.*

In the above example, as the first 3 utterances map to the same sentence, there is a higher probability their context vectors will shuffle amongst themselves for a particular time step. Such shuffling operations directly inform the decoder about different context vectors that map to the same output labels.

4. Experiments

4.1. Datasets

All our experiments are performed on a Marathi dataset [25] consisting of labeled speech utterances in Marathi from speakers in three different demographics: *Urban-Poor, Rural-Poor and College*. We create two kinds of training datasets using these utterances.

- UCR: Combine data from all the three demographics.
- UC: Combine data from *Urban-poor* and *College*.

Table 1 lists the dataset details, along with the train, developement and test set splits, for both UCR and UC. (The UC dataset allows us to evaluate performance on Rural-Test without having any access to training data from speakers in the Rural-Poor category.)

4.2. Implementation details

All the ASR models were implemented using the ESPNet toolkit [26]. Our base model is a hybrid attention-CTC model with a mixing coefficient of $\beta = 0.4$. Our encoder has 2 VGG-ish convolution layers followed by 3 *Bi-LSTM* layers with 1024 units each. The last two layers of the encoder are pyramidal in nature skipping every other input. For the *CTC* part of the hybrid loss, the final layer of the encoder is fully connected to an output layer, followed by a softmax distribution over the output tokens. For the attention-based part of the hybrid loss, the encoder outputs feed into a location based attention (1024 units) which feeds into a 2 layer LSTM Decoder of 1024 units each. We use an ADADELTA [27] optimizer with starting learning rate of 1. We regularize using a *dropout* rate of 0.5 and perform scheduled-sampling during training with a probability of 0.3. Our vocabulary consists of 150 sub-words. For context shuffling with lexicographic curriculum, we use 4-grams ($a = 3, b = 1$).

5. Results

5.1. Urban College Rural

Table 2 shows results by training on the UCR dataset and testing on Rural-Test (denoted by Test_R) and a combination of Urban-Test and College-Test (denoted by Test_UC). We show

Model	Test_R(CER/WER)	Test_UC(CER/WER)
BASELINE	0.1666/0.2896	0.1752/0.3065
$\text{BASELINE}_\text{cur}$	0.1781/0.3056	0.1763/0.3098
$\text{BASELINE}_\text{SG}$	0.1610/0.2790	0.1681/0.2920
COU ($\lambda = 0.0001$)	**0.1459/0.2572**	0.1527/0.2762
COU ($\lambda = 0.00001$)	0.1450/0.2649	0.1491/0.2771
SHUF ($\eta = 0.3$)	0.1343/0.2606	**0.1436/0.2722**
SHUF ($\eta = 0.5$)	0.1352/0.2601	0.1444/0.2743
SORT	0.1382/0.2461	0.1413/0.2675
SORT ($\eta = 0.3$)	0.1331/0.2509	0.1309/0.2626
SORT ($\eta = 0.35$)	0.1324/0.2601	0.1315/0.2643
SORT ($\eta = 0.4$)	**0.1230/0.2403**	**0.1332/0.2612**
SORT ($\eta = 0.45$)	0.1340/0.2630	0.1391/0.2731
SORT ($\eta = 0.5$)	0.1314/0.2524	0.1386/0.2699

Table 2: *Experiments on* UCR *dataset. We compare three baseline methods against coupled loss, context shuffling and lexicographic sorting.*

results for both context shuffling and the lexicographic curriculum, along with comparisons against three baseline systems including 1) BASELINE, a standard hybrid model 2) $\text{BASELINE}_\text{cur}$, a curriculum learning baseline hybrid model where batches are fed in the order of increasing input length, and 3) $\text{BASELINE}_\text{SG}$, a SortaGrad baseline where we perform curriculum for only the first epoch [28]. We show three systems with our proposed technique: 1) COU ($\lambda = N$) that refers to using the coupled loss with λ set to different values 2) SHUF ($\eta = N$) that refers to using context shuffling with paired inputs and different values of η, and 3) SORT ($\eta = N$) that refers to using lexicographic sorting followed by context shuffling with different values of η. Note that the system SORT (without any η) refers to batching after a lexicographic sort without any context shuffling. Table 2 shows that while coupled loss and context shuffling over paired inputs provide significant gains over the baseline systems, the SORT system without any changes in context vectors or loss functions provides a further boost in performance. Context shuffling with SORT provides further improvements with SORT ($\eta = 0.4$) performing the best compared to all other systems on both the development and test sets. With our proposed techniques, the test performance on both Test_R and Test_R improve showing that both types of speech samples benefit from our training techniques.

Similarly on the UC dataset, as shown in Table 3, we see significant reductions in WER over the baseline using both context shuffling and lexicographic sorting (with and without context shuffling). We note that adding an additional 25 hours of Rural-Train data hurt performance on Test_UC (comparing BASELINE numbers in Table 3 and Table 2), thus motivating the need for techniques such as ours that help make additional

Model	Test_R(CER/WER)	Test_UC(CER/WER)
BASELINE	0.1806/0.3027	0.1683/0.3011
SHUF ($\eta = 0.3$)	**0.1450/0.2654**	**0.1420/0.2738**
SORT	0.1542/0.2886	0.1424/0.2752
SORT ($\eta = 0.3$)	**0.1450/0.2736**	**0.1385/0.2736**
SORT ($\eta = 0.4$)	0.1496/0.2877	0.1368/0.2790
SORT ($\eta = 0.5$)	0.1491/0.2848	0.1347/0.2689

Table 3: *Experiments on* UC *dataset. We compare baseline against context shuffling and lexicographic sorting.*

training data in the same language useful even when it comes from a different demographic or accent.

6. Conclusion

We have proposed two simple but effective training techniques that take advantage of the redundancy in training transcriptions to improve the performance of a Marathi ASR system across different accented speech samples. Future work includes replicating these results across datasets in other languages exhibiting different levels of redundancy in transcriptions.

7. References

[1] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson, and N. Jaitly, "A Comparison of sequence-to-sequence models for speech recognition," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2017-Augus, pp. 939–943, 2017.

[2] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, Attend and Spell," aug 2015. [Online]. Available: http://arxiv.org/abs/1508.01211

[3] A. Graves, S. Fernandez, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification : Labelling Unsegmented Sequence Data with Recurrent Neural Networks," *Proceedings of the 23rd international conference on Machine Learning*, pp. 369–376, 2006.

[4] A. Graves, "Sequence Transduction with Recurrent Neural Networks," 2012. [Online]. Available: http://arxiv.org/abs/1211.3711

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention Is All You Need," jun 2017. [Online]. Available: http://arxiv.org/abs/1706.03762https://arxiv.org/abs/1706.03762

[6] T. Hori, S. Watanabe, and J. Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, Jul. 2017, pp. 518–529. [Online]. Available: https://www.aclweb.org/anthology/P17-1048

[7] J. J. Humphries and P. C. Woodland, "Using Accent-Specific Pronunciation Modelling for Improved Large Vocabulary Continuous Speech Recognition," *5th European Conference on Speech Communication and Technology*, pp. 1–4, 1997.

[8] Y. Zheng, R. Sproat, L. Gu, I. Shafran, H. Zhou, Y. Su, D. Jurafsky, R. Starr, and S.-Y. Yoon, "Accent detection and speech recognition for Shanghai-accented Mandarin." *Interspeech*, pp. 7–10, 2005. [Online]. Available: http://t3-1.yum2.net/index/www-nlp.Stanford.EDU/pubs/p1304.pdf

[9] M. Chen, Z. Yang, J. Liang, Y. Li, and W. Liu, "Improving Deep Neural Networks Based Multi-Accent Mandarin Speech Recognition Using I-Vectors and Accent-Specific Top layer Electric Power Research Institute of ShanXi Electric Power Company , China State Grid Corp," *InterSpeech*, pp. 3620–3624, 2015.

[10] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[11] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the KLD-regularized model adaptation," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, no. September, pp. 2977–2981, 2014.

[12] T. Fraga-Silva, J.-L. Gauvain, and L. Lamel, "Speech recognition of multiple accented english data using acoustic model interpolation," in *2014 22nd European Signal Processing Conference (EUSIPCO)*, 2014, pp. 1781–1785.

[13] S. Yoo, I. Song, and Y. Bengio, "A Highly Adaptive Acoustic Model for Accurate Multi-dialect Speech Recognition," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2019, pp. 5716–5720. [Online]. Available: https://ieeexplore.ieee.org/document/8683705/

[14] X. Yang, K. Audhkhasi, A. Rosenberg, S. Thomas, B. Ramabhadran, and M. Hasegawa-Johnson, "Joint Modeling of Accents and Acoustics for Multi-Accent Speech Recognition," 2018. [Online]. Available: http://arxiv.org/abs/1802.02656

[15] A. Jain, M. Upreti, and P. Jyothi, "Improved Accented Speech Recognition Using Accent Embeddings and Multi-task Learning," no. September, pp. 2454–2458, 2018.

[16] A. Jain, V. P. Singh, and S. P. Rath, "A Multi-Accent Acoustic Model Using Mixture of Experts for Speech Recognition," in *Interspeech 2019*. ISCA: ISCA, sep 2019, pp. 779–783. [Online]. Available: http://www.isca-speech.org/archive/Interspeech{_}2019/abstracts/1667.html

[17] K. Rao and H. Sak, "Multi-accent speech recognition with hierarchical grapheme based models," *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, pp. 4815–4819, 2017.

[18] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain Adversarial Training for Accented Speech Recognition," 2018. [Online]. Available: http://arxiv.org/abs/1806.02786

[19] J. Shor, D. Emanuel, O. Lang, O. Tuval, M. Brenner, J. Cattiau, F. Vieira, M. McNally, T. Charbonneau, M. Nollstadt, A. Hassidim, and Y. Matias, "Personalizing ASR for Dysarthric and Accented Speech with Limited Data," *Interspeech 2019*, pp. 784–788, jul 2019. [Online]. Available: http://www.isca-speech.org/archive/Interspeech{_}2019/abstracts/1427.htmlhttp://arxiv.org/abs/1907.13511

[20] T. Viglino, P. Motlicek, and M. Cernak, "End-to-End Accented Speech Recognition," ISCA, p. jkj, sep 2019. [Online]. Available: http://www.isca-speech.org/archive/Interspeech{_}2019/abstracts/2122.html

[21] S. Ghorbani, A. E. Bulut, and J. H. L. Hansen, "Advancing multi-accented lstm-ctc speech recognition using a domain specific student-teacher learning paradigm," 2019.

[22] G. I. Winata, S. Cahyawijaya, Z. Liu, Z. Lin, A. Madotto, P. Xu, and P. Fung, "Learning Fast Adaptation on Cross-Accented Speech Recognition," mar 2020. [Online]. Available: http://arxiv.org/abs/2003.01901

[23] B. Li, T. Sainath, K. Sim, M. Bacchiani, E. Weinstein, P. Nguyen, Z. Chen, Y. Wu, and K. Rao, "Multi-dialect speech recognition with a single sequence-to-sequence model," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4749–4753, 2018.

[24] V. Unni, N. Joshi, and P. Jyothi, "Coupled Training of Sequence-to-Sequence Models for Accented Speech Recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, may 2020, pp. 8254–8258. [Online]. Available: https://ieeexplore.ieee.org/document/9052912/

[25] B. Abraham, D. Goel, D. Siddarth, K. Bali, M. Chopra, M. Choudhury, P. Joshi, P. Jyoti, S. Sitaram, and V. Seshadri, "Crowdsourcing speech data for low-resource languages from low-income workers," in *Proceedings of The 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 2819–2826. [Online]. Available: https://www.aclweb.org/anthology/2020.lrec-1.343

[26] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. E. Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "ESPNet: End-to-end speech processing toolkit," *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, vol. 2018-Septe, pp. 2207–2211, 2018.

[27] M. D. Zeiler, "ADADELTA: An Adaptive Learning Rate Method," dec 2012. [Online]. Available: http://arxiv.org/abs/1212.5701

[28] D. Amodei, R. Anubhai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, J. Chen, M. Chrzanowski, A. Coates, G. Diamos, E. Elsen, J. Engel, L. Fan, C. Fougner, T. Han, A. Hannun, B. Jun, P. LeGresley, L. Lin, S. Narang, A. Ng, S. Ozair, R. Prenger, J. Raiman, S. Satheesh, D. Seetapun, S. Sengupta, Y. Wang, Z. Wang, C. Wang, B. Xiao, D. Yogatama, J. Zhan, and Z. Zhu, "Deep speech 2: End-to-end speech recognition in english and mandarin," 2015.