Learning Noise Invariant Features Through Transfer Learning for Robust End-to-End Speech Recognition

Citation for published version:
Zhang, S, Do, C-T, Doddipatla, R & Renals, S 2020, Learning Noise Invariant Features Through Transfer Learning for Robust End-to-End Speech Recognition. in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* Institute of Electrical and Electronics Engineers (IEEE), pp. 7024-7028, 2020 IEEE International Conference on Acoustics, Speech, and Signal Processing, Barcelona, Spain, 4/05/20. https://doi.org/10.1109/ICASSP40776.2020.9053169

Digital Object Identifier (DOI):
10.1109/ICASSP40776.2020.9053169

Link:
Link to publication record in Edinburgh Research Explorer

Document Version:
Peer reviewed version

Published In:
ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

LEARNING NOISE INVARIANT FEATURES THROUGH TRANSFER LEARNING FOR ROBUST END-TO-END SPEECH RECOGNITION

Shucong Zhang [1], *Cong-Thanh Do* [2], *Rama Doddipatla* [2] *and Steve Renals* [1]

[1] Centre for Speech Technology Research, University of Edinburgh, Edinburgh, UK
[2] Toshiba Cambridge Research Laboratory, Cambridge, United Kingdom

ABSTRACT

End-to-end models yield impressive speech recognition results on clean datasets while having inferior performance on noisy datasets. To address this, we propose transfer learning from a clean dataset (WSJ) to a noisy dataset (CHiME-4) for connectionist temporal classification models. We argue that the clean classifier (the upper layers of a neural network trained on clean data) can force the feature extractor (the lower layers) to learn the underlying noise invariant patterns in the noisy dataset. While training on the noisy dataset, the clean classifier is either frozen or trained with a small learning rate. The feature extractor is trained with no learning rate re-scaling. The proposed method gives up to 15.5% relative character error rate (CER) reduction compared to models trained only on CHiME-4. Furthermore, we use the test sets of Aurora-4 to perform evaluation on unseen noisy conditions. Our method has significantly lower CERs (11.3% relative on average) on all 14 Aurora-4 test sets compared to the conventional transfer learning method (no learning rate re-scale for any layer), indicating our method enables the model to learn noise invariant features.

Index Terms— end-to-end, robust speech recognition, transfer learning

1. INTRODUCTION

End-to-end speech recognition models simplify the training procedure compared to conventional hybrid systems and have offered impressive performance [1–3]. However, end-to-end models usually have inferior results on noisy data [4, 5].

Several methods have been proposed to help training on noisy data by exploiting clean data, such as teacher-student learning [6–8] and multi-task learning [9–11]. However, in these methods, typically require parallel clean/noisy data, which limits their usefulness. Transfer learning transfers the knowledge learned from the source domain to the target domain and does not require parallel data. In this work, to exploit non-parallel clean and noisy data in training end-to-end speech recognition models, we propose a novel transfer learning from clean speech data to noisy speech data.

The first author performed the work while at Toshiba Cambridge

Transfer learning has been widely employed in training speech recognition models for low-resource languages [12–17]. The low-level features of different languages are generally similar. Thus, a model is usually trained on a well-resourced language and the feature extractor (the lower layers of a neural network) is transferred across languages. However, for transfer learning from clean to noisy data, although the underlying patterns should be invariant to the noise conditions, it is not suitable to transfer the feature extractor owing to the mismatch of acoustic conditions,

We propose to transfer the classifier (the upper layers) from clean to noisy data, rather than transferring the feature extractor, by first training the classifier on the clean dataset. While training on the noisy dataset, the clean classifier is either frozen or tuned using a small learning rate; the feature extractor is trained using the normal learning rate. The feature extractor is constrained to learn features that match the clean classifier. Since the features fit the clean classifier reflect more about the underlying patterns, the clean classifier helps the feature extractor to learn the noise invariant patterns from the noisy data.

We apply the proposed transfer learning method to train connectionist temporal classification (CTC) models transferring from WSJ [18] to CHiME-4 [19, 20]. Compared to models trained directly on CHiME-4, the models trained with the proposed method reduce the character error rate (CER) by up to 15.5% relative. We also tested the performance of our method on unseen noise conditions using the Aurora-4 test sets [21]. Compared to conventional transfer learning in which there is no re-scale of the learning rate for any layer, the proposed method has significantly lower CERs on all 14 test sets with an average relative reduction in CER of 11%. These experiments indicate that the proposed transfer learning method helps the model to learn noise invariant features.

2. RELATED WORK

The core idea of our proposed transfer learning approach is that clean speech features should be similar to features which are invariant to different noise conditions. Thus, making the features extracted from noisy data similar to features extracted from clean data should be helpful.

To achieve this objective, teacher-student learning is applied [6–8]. In general, parallel clean/noisy data is required for teacher-student, with the noisy data often generated by adding noise to the clean data. The teacher model is trained on the clean dataset, and the output distribution of the teacher model of clean utterances is used as soft labels for the parallel noisy utterances. When training the student model on the noisy utterances, we employ an objective which minimizes the KL divergence between the soft labels and the output distribution of the student model. The transcript of the utterance is viewed as a hard label sequence and may be optionally used when training the student model.

Multi-task learning also helps in training robust speech recognition models by exploiting parallel data [9–11]. In multi-task learning, the main training objective is to recognise the noisy utterance and the secondary objective can be to reconstruct the clean utterance. An alternative secondary objective is to minimize the distance of the output of each layer between the model training on the noisy utterance and the model training on the parallel clean utterance.

Although these methods enable models to learn domain invariant features and improve the speech recognition results on noisy data, parallel data is necessary. In contrast, our proposed transfer learning method exploits knowledge learned on non-parallel clean data.

Transfer learning for low-resource languages [12–17] uses feature extractors trained on well-resourced languages, following which the classifier is reinitialized and retrained using the low-resource language. The feature extractor is either jointly trained or kept frozen, following an optional fine-tuning stage. For our proposed method, we transfer the classifier to force the feature extractor to learn noise invariant features.

3. CONNECTIONIST TEMPORAL CLASSIFICATION

Connectionist temporal classification (CTC) models [22] belong to the family of sequence-to-sequence models. They can be applied to end-to-end speech recognition since this model is alignment-free – it considers all the valid alignments. In this work, the inputs for the CTC models are acoustic features and the outputs are characters.

For an input sequence $\boldsymbol{X} = \boldsymbol{x}_1, \cdots, \boldsymbol{x}_t$, a valid output sequence $\boldsymbol{A} = \boldsymbol{a}'_1, \cdots, \boldsymbol{a}'_t$ contains repeated characters and blank symbols $(-)$. The repeated characters between blank symbols will be merged into one character to generate the true output sequence $\boldsymbol{Y} = \boldsymbol{y}_1, \cdots, \boldsymbol{y}_n$. For example, for $\boldsymbol{X} = \boldsymbol{x}_1, \cdots, \boldsymbol{x}_5$, $\boldsymbol{A} = c, c, -, a, t$ is a valid output for $\boldsymbol{Y} = c, a, t$. During training, the model maximizes the probability of the ground truth character sequence, which is the sum of the probability of all valid alignments:

$$P(\boldsymbol{Y}|\boldsymbol{X}) = \sum_{\boldsymbol{A} \in S} P(\boldsymbol{A}|\boldsymbol{X}) \,, \tag{1}$$

where S represents the set of valid alignments. CTC models are usually using by bidirectional recurrent neural networks, and the probability of each valid sequence is given by

$$P(\boldsymbol{A}|\boldsymbol{X}) = \prod_{i=1}^{t} P(\boldsymbol{a}_i|\boldsymbol{X}) \,. \tag{2}$$

The probability $P(\boldsymbol{Y}|\boldsymbol{X})$ can be computed efficiently through a forward-backward algorithm. The decoding can be performed in a greedy way or using beam search.

4. TRANSFER LEARNING

We view the top layers of a model as a classifier and the layers beneath these top layers as a feature extractor. The classifier divides the space, while the feature extractor provides features based on this partition of the space. A well-trained clean model (i.e. a model which has a good speech recognition performance on the clean dataset) gives a “clean classifier”. We assume the features extracted by the well-trained clean model are close to the underlying patterns. Moreover, these features match the clean classifier. If we transfer the clean classifier and train the feature extractor on the noisy data, then the feature extractor is forced to extract features that fit the clean classifier. Since the features that fit the clean classifier should be close to the underlying patterns, the feature extractor is forced to extract noise invariant features from the noisy data.

In our transfer learning method, we do not only consider the output softmax layer as the classifier. We also view the grouping of the softmax layer and several other upper layers as the classifier. With more layers, the classifier is more powerful and may ease the burden of learning features from noisy data for the feature extractor. However, more layers for the classifier also means fewer layers for the feature extractor. It also makes the feature extractor less powerful and less flexible. Thus, with too few layers, the feature extractor may not have the capacity to fit the classifier. In our experiments, we set the number of layers for the classifier using the validation set.

While training on the noisy data, the clean classifier is either frozen or tuned with a small learning rate. The feature extractor is either reinitialized or initialized with the weights of the clean feature extractor, as if the training on the clean data is considered as a pre-training stage. The feature extractor is trained with the normal learning rate without any learning rate re-scaling.

5. EXPERIMENTS

We apply the proposed transfer learning method from WSJ [18] to CHiME-4 [19, 20]. For WSJ, we use si284 as the training set and dev93 as the validation set. For CHiME-4, we use the single channel simulated noisy and the real

	in_channel	out_channel	kernel	stride
conv	1	64	3×3	1
conv	64	64	3×3	1
maxpool			2×2	2
conv	64	128	3×3	1
conv	128	128	3×3	1
maxpool			2×2	2

Table 1. The CNN architecture for the CNN-BLSTM model

noisy data from all channels as the training set. We use dt05_multi_isolated_1ch_track as the development set and et05_real_isolated_1ch_track as the evaluation set. For CHiME-4, the noise conditions in the training and test sets match. To test the performance of the models in unseen noise conditions, we use all 14 test sets in the Aurora-4 corpus [21]. The 14 test sets contain two clean sets which were recorded by a primary closed microphone and a distant secondary microphone. These two sets were corrupted by six different additive noises to create the other 12 test sets, making 14 test sets in total.

We use Kaldi [23] to extract 40-dimension mel-scale filterbank features with three pitch features, and the ESPnet toolkit [24] to build convolutional neutral networks (CNNs) – bidirectional long short-term memory (BLSTM) [25] CTC models. The architecture of the CNNs is in Table 1. There are four BLSTM layers on top of the CNNs. Each BLSTM layer is followed by a linear layer with tanh activation. All the BLSTM layers and linear layers have 320 hidden units. There are 50 output labels in total, correspondingto 26 characters, apostrophe, period, dash, space, noise, sos/eos tokens, and some other special tokens. Adadelta [13, 26] is used as the optimizer. The training stops after 5 epochs if there is no reduction upon the lowest validation loss.

For the transfer learning (TL), firstly a CTC model is trained using the clean WSJ si284 training set. This model is used as the base model for the transfer learning experiments. The top layers of this clean CTC model are viewed as a clean classifiers. While performing TL in CHiME-4, we either freeze or tune the clean classifier with a small learning rate (the learning rate given by Adadelta is reduced by a factor). The bottom layers are optionally reinitialized. The method is illustrated in Figure 1. We compare the performance of the proposed method against models that trained only using the CHiME-4 dataset and also models trained using conventional TL from WSJ to CHiME-4, where all the layers are tuned without learning rate rescaling.

5.1. Random reinitialization of the feature extractor

Table 2 shows the performance of the proposed TL approach to freeze the clean classifier trained in WSJ, reinitialize randomly and retrain the feature extractor using CHiME-4 data. When the top two layers (the softmax layer and the topmost BLSTM layer with its following linear layer) are frozen, the model gives significantly smaller CER compared to the CNN-BLSTM model trained only using CHiME-4. We also notice that if only the softmax layer is frozen, the model does not outperform the baseline, which implies the frozen softmax does not have the capacity to force the feature extractor to learn better features. On the other hand, although freezing three layers surpasses the baseline, it gives inferior results compared to freezing two layers, which indicates that although the three-layered classifier has more capacity, the shallower feature extractor is not powerful/flexible enough to well fit the classifier. The capacity of the classifier and the feature extractor are well balanced when the top two layers are frozen.

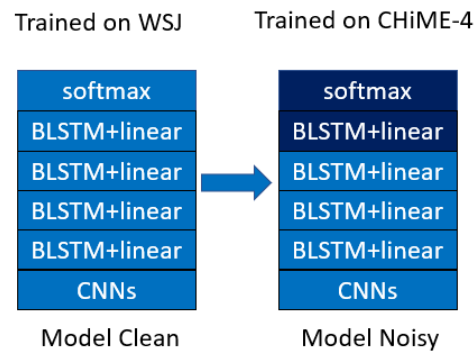

Fig. 1. An illustration of the transfer learning from WSJ to CHiME-4. Model Clean is trained on WSJ. Then, the top two layers (we group each BLSTM layer and its following linear layer as one layer) are either frozen or tuned with a small learning rates on the CHiME-4 noisy data. The bottom layers of Model Noisy are either random initialized or initialized using the weights of Model Clean, and trained on CHiME-4 with no learning rate re-scaling.

Model/CER	dt05_multi	et05_real
Freeze one layer	28.5	38.9
Freeze two layers	**25.7**	**36.0**
Freeze three layers	25.7	37.1
No transfer learning		
CNN-BLSTM CTC	29.0	38.7
BLSTM CTC [4]	/	48.8

Table 2. Character error rate (CER) of different models. No transfer learning means the models are trained only using CHiME-4. The results of BLSTM CTC is from a previous work [4].

5.2. Pre-training of the feature extractor

Here we consider the case of pre-training the feature extractor using WSJ and further training using CHiME-4. That is, the feature extractor is not randomly reinitialized. Instead, it

Model/CER	airport wv1	babble wv1	car wv1	clean wv1	restaurant wv1	street wv1	train wv1	Average
Model A	**11.3**	**12.0**	**9.8**	**8.1**	**13.1**	**12.7**	**13.6**	**11.5**
Model B	12.2	13.1	11.2	9.7	14.1	14.0	14.6	12.7
Model/CER	airport wv2	babble wv2	car wv2	clean wv2	restaurant wv2	street wv2	train wv2	Average
Model A	**26.6**	**27.3**	**24.1**	**22.1**	**27.4**	**27.9**	**28.2**	**26.2**
Model B	30.5	30.8	26.9	25.2	31.5	31.2	32.5	29.8

Table 3. Character error rate on all 14 test sets of Aurora-4. Both model A and model B are initialized using the clean CTC model trained on WSJ. For model A, the learning rate for the top two layers (the softmax layer and the topmost BLSTM layer with its following linear layer) are scaled by a factor of 0.5. For model B, there is no learning rate re-scale. wv1 means the utterances are recorded by the primary microphone and wv2 means the utterances are recorded by the secondary microphone.

Model/CER	dt05_Multi	et05_real
Classifier: One layer		
Frozen	22.0	33.8
LR scaled by 0.1	22.2	33.5
LR scaled by 0.5	22.6	34.2
Classifier: two layers		
Frozen	22.5	33.8
LR scaled by 0.1	22.0	33.3
LR scaled by 0.5 (Model A)	**21.9**	**32.9**
Classifier: three layers		
Frozen	24.6	36.6
LR scaled by 0.1	22.7	34.5
LR scaled by 0.5	21.9	33.4
No LR re-scale (Model B)	21.9	33.3
LR scaled by 0.5 for all layers (Model C)	22.1	33.4
No transfer learning	29.0	38.7

Table 4. Character error rate (CER) for using the clean CTC model to initialize the training on CHiME-4. The classifier is made of the topmost layer(s). During the training on CHiME-4, the classifier is either frozen or the learning rate (LR) is scaled by a small factor. No transfer learning means the model is trained only using CHiME-4.

is initialized by the weights of the clean CTC model. When training on CHiME-4, the classifier is either frozen, or the learning rate is reduced by a small factor. There is no learning rate re-scaling for the feature extractor. We compare the proposed transfer learning method with the conventional transfer learning method. That is, for all layers, there is no learning rate re-scale. Table 4 summarizes the results.

The CERs in Table 4 are lower than the CERs in Table 2, indicating that pre-training on WSJ gives a good initialization for all the layers. Again, the capacity of the classifier and the feature extractor are well balanced when the top two layers are grouped as the classifier. The best CER is achieved by tuning the two-layered classifier using a scaled learning rate (0.5). To ascertain if the performance gains of the best model (Model A) is due to the smaller learning rate, we train a model by scaling the learning rate for all layers by 0.5 (Model C), which gives inferior results. Thus, these experiments imply that the classifier needs to be slightly tuned (compared to the feature extractor) for the noisy data to get the best performance. We also test a two-stage fine tune, i.e., first freeze the softmax layer then unfreeze it and do a fine-tuning. However, it does not give better results compared to the best model.

To show our proposed method forces the models to learn noise invariant features, we test the performance of Model A and Model B in unseen noise conditions by decoding them on all the 14 test sets of Aurora-4 dataset (Model B is trained by the conventional transfer learning, in where all layers are trained jointly with no learning rate re-scale). Table 3 shows that Model A surpasses Model B for all 14 test sets. Compared to Model B, Model A has 11.3% relatively lower CER on average. These results show that Model A is more capable of extracting domain invariant features. In the CHiME-4 dataset, the training set contains the noise conditions of the test set, and the conventional transfer learning method makes Model B biased toward these observed noise conditions. Thus, Model B has a good performance in the test set of CHiME-4 but inferior results in the test sets of Aurora-4. Our transfer learning method only slightly tuned the clean classifier and it forces the feature extractor to extract the noisy invariant underlying patterns. Thus, Model A has the best performance in both seen and unseen noisy condition.

6. CONCLUSION

In this paper, we present a novel transfer learning method from clean data to noisy data for speech recognition. In our proposed method, a clean model is firstly trained on clean data. Then, the clean classifier of the clean model (the top layers) are either frozen or trained with a small learning rate on the noisy data. The feature extractor (the bottom layers) is trained on the noisy data with no learning rate re-scale. Our experiment results on CTC models show our method forces the feature extractor to learn noise invariant features and leads to significant character error rate reductions. Our proposed method is not only constrained to CTC models. Testing our transfer learning methods for other models is left as a further work.

7. REFERENCES

[1] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," *arXiv preprint arXiv:1706.02737*, 2017.

[2] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J Weiss, Kanishka Rao, Ekaterina Gonina, et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[3] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, "Improved training of end-to-end attention models for speech recognition," *arXiv preprint arXiv:1805.03294*, 2018.

[4] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 4835–4839.

[5] Cong-Thanh Do, "Subband temporal envelope features and data augmentation for end-to-end recognition of distant conversational speech," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6251–6255.

[6] Jinyu Li, Michael L Seltzer, Xi Wang, Rui Zhao, and Yifan Gong, "Large-scale domain adaptation via teacher-student learning," *arXiv preprint arXiv:1708.05466*, 2017.

[7] Zhong Meng, Jinyu Li, Yifan Gong, and Biing-Hwang Juang, "Adversarial teacher-student learning for unsupervised domain adaptation," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5949–5953.

[8] Ladislav Mošner, Minhua Wu, Anirudh Raju, Sree Hari Krishnan Parthasarathi, Kenichi Kumatani, Shiva Sundaram, Roland Maas, and Björn Hoffmeister, "Improving noise robustness of automatic speech recognition via parallel data and teacher-student learning," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 6475–6479.

[9] Bin Huang, Dengfeng Ke, Hao Zheng, Bo Xu, Yanyan Xu, and Kaile Su, "Multi-task learning deep neural networks for speech feature denoising," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[10] Masato Mimura, Shinsuke Sakai, and Tatsuya Kawahara, "Joint optimization of denoising autoencoder and dnn acoustic model based on multi-target learning for noisy speech recognition.," in *Seventeenth Annual Conference of the International Speech Communication Association*, 2016, pp. 3803–3807.

[11] Davis Liang, Zhiheng Huang, and Zachary C Lipton, "Learning noise-invariant representations for robust speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 56–63.

[12] Samuel Thomas, Sriram Ganapathy, and Hynek Hermansky, "Multilingual mlp features for low-resource lvcsr systems," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4269–4272.

[13] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Unsupervised cross-lingual knowledge transfer in dnn-based lvcsr," in *2012 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2012, pp. 246–251.

[14] Arnab Ghoshal, Pawel Swietojanski, and Steve Renals, "Multilingual training of deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7319–7323.

[15] Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 7304–7308.

[16] Sibo Tong, Philip N Garner, and Hervé Bourlard, "Multilingual training and cross-lingual adaptation on ctc-based acoustic model," *arXiv preprint arXiv:1711.10025*, 2017.

[17] Jaejin Cho, Murali Karthick Baskar, Ruizhi Li, Matthew Wiesner, Sri Harish Mallidi, Nelson Yalta, Martin Karafiat, Shinji Watanabe, and Takaaki Hori, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 521–527.

[18] Douglas B Paul and Janet M Baker, "The design for the wall street journal-based csr corpus," in *Proceedings of the workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, pp. 357–362.

[19] Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, Jon Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, vol. 46, pp. 535–557, 2017.

[20] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third 'chime' speech separation and recognition challenge: Analysis and outcomes," *Computer Speech and Language*, vol. 46, pp. 605–626, 2017.

[21] David Pearce, "Aurora working group: Dsr front end lvcsr evaluation au/384/02," 2002.

[22] 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.

[23] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number CONF.

[24] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., "Espnet: End-to-end speech processing toolkit," *arXiv preprint arXiv:1804.00015*, 2018.

[25] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[26] Matthew D Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.