INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Multi-domain Knowledge Distillation via Uncertainty-Matching for End-to-End ASR Models

Ho-Gyeong Kim[1], *Min-Joong Lee*[1], *Hoshik Lee*[1], *Tae Gyoon Kang*[1], *Jihyun Lee*[1], *Eunho Yang*[2], *and Sung Ju Hwang*[2]

[1]Samsung Advanced Institute of Technology, Samsung Electronics, South Korea
[2]KAIST, South Korea

hogyeong.kim@samsung.com

Abstract

Knowledge Distillation basically matches predictive distributions of student and teacher networks to improve performance in an environment with model capacity and/or data constraints. However, it is well known that predictive distribution of neural networks not only tends to be overly confident, but also cannot directly model various factors properly that contribute to uncertainty. Recently, deep learning studies based on uncertainty have been successful in various fields, especially in several computer vision tasks. The prediction probability can implicitly show the information about how confident the network is, however, we can explicitly utilize confidence of the output by modeling the uncertainty of the network. In this paper, we propose a novel knowledge distillation method for automatic speech recognition that directly models and transfers the uncertainty inherent in data observation such as speaker variations or confusing pronunciations. Moreover, we investigate an effect of transferring knowledge more effectively using multiple teachers learned from various domains. Evaluated on WSJ which is the standard benchmark dataset with limited instances, the proposed knowledge distillation method achieves significant improvements over student baseline models.

Index Terms: aleatoric uncertainty, end-to-end speech recognition, multi-domain, knowledge distillation

1. Introduction

Knowledge distillation (KD) aims to improve the performance of a target network with limited capacity or data, by transferring the knowledge of a teacher network, possibly larger or trained with more data. KD is originally proposed in [1] to minimize the performance degradation even under resource constraints by matching the softmax prediction probability of student network with that of teacher network. In recent years, beyond the information on softmax layer, the knowledge of intermediate layers [2] or attention maps [3] is used for matching to further improve performance. As a general purpose compression technique, this KD has been applied for various fields including not only computer vision tasks [4, 5, 6, 7] but automatic speech recognition (ASR) [8, 9, 10, 11].

The core principle of KD proposed in [1] is that the softmax probability generated by teacher network contains much more useful information than a single point of output label, to guide the student network. That is, by matching predictive distributions, the student network can *implicitly* utilize the information about how confident the teacher network is about the given input data. We go one step further and ask the following question: Can we *explicitly* utilize the information about the *uncertainty* of teacher network?

In fact, the task of measuring the uncertainty of the given model itself is challenging (especially for deep models), and several studies have been conducted. As a seminal work in this field, [12] has shown that deep neural networks trained with dropout regularization is basically a variational approximation of the posterior of a deep Gaussian process, and that a dropout-regularized network can output uncertainty by applying dropout at test time. However, as in revealed in [13], it is helpful to model the noise inherent in observations called aleatoric uncertainty (uncertainty that cannot be explained away even with sufficient data) in situations where data is sufficient.

In this paper, we study a method of knowledge distillation where the student network is restricted to learn with little data due to issues such as privacy. Specifically, we directly transfer uncertainty information by modeling token-wise uncertainty which type is heteroscedastic or aleatoric. To further boost the performance, we also propose to distill the knowledge from several teacher networks from multiple domains via dynamic weighting. We show that the proposed KD method achieves significant improvements over the student baseline on Wall Street Journal (WSJ) [14] when the teacher models are trained on several standard benchmark datasets for ASR such as LibriSpeech [15] and Tedlium2 [16].

2. Knowledge distillation for E2E ASR

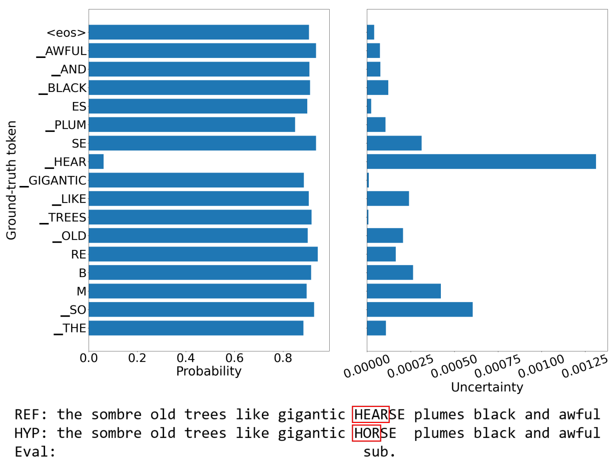
As a sequence labeling task, the end-to-end (E2E) ASR takes a sequence of input tokens $\mathbf{s}$ and makes predictions on output text $\mathbf{t}$ by minimizing the negative log-likelihood (NLL),

$$\mathcal{L}_{\text{NLL}} = -\sum_{i}\sum_{c=1}^{|C|}\delta\left(y_i, c\right)\log p\left(t_i = c|\mathbf{s};\theta\right) \tag{1}$$

where y_i is the i-th ground truth in the target text and in $p(t_i = c|\mathbf{s};\theta)$ we suppress the dependency on predictions or ground truth for previous tokens. Here, C indicates the output vocabulary and the symbol c is the class index. θ is the model parameter, and δ indicates Kronecker delta which is a function of two variables where the function is 1 if the variables are equal, and 0 otherwise. This objective can be seen as minimizing the cross-entropy between the target and model output probability distribution.

The goal of KD is to effectively transfer the knowledge of a teacher model to a student model. An original teacher model, which performs the task well enough, guides a student model with limited data or capacity. As a consequence, the student model retains the distilled knowledge of the teacher model. The

http://dx.doi.org/10.21437/Interspeech.2021-1169



```
REF: the sombre old trees like gigantic HEARSE plumes black and awful
HYP: the sombre old trees like gigantic HORSE  plumes black and awful
Eval:                                   sub.
```

Figure 1: *Example of a recognition result, probability and uncertainty for the "5683-32866-0028" utterance in a test-clean set of Librispeech. The error occurs at the confusing pronunciation between* HEAR *and* HOR.

standard form of token-level KD [8, 9, 10] for compressing E2E ASR models is as follows:

$$\mathcal{L}_{\text{tok-KD}} = -\sum_{i}\sum_{c=1}^{|C|} \hat{p}\left(t_i = c|\mathbf{s};\theta_T\right)\log p\left(t_i = c|\mathbf{s};\theta\right), \quad (2)$$

where $\hat{p}_i$ (instead of δ in NLL loss) is the i-th token output distribution from the teacher model parameterized by θ_T which is pre-learned model parameters of the teacher model. Minimizing this loss function can be viewed as minimizing the KL-divergence between teacher and student's probability distributions.

3. Uncertainty-matching knowledge distillation

Uncertainty is additional information to the predictive distribution, which shows how much confident predictive probabilities are. In [13], they proposed to capture uncertainty in both the model and data using MC-dropout and input-dependent variance modeling, and applied it to semantic segmentation and depth estimation. In this paper, we utilize uncertainty in KD experiments for sequence-to-sequence models.

3.1. Uncertainty modeling for E2E ASR

According to [13], uncertainty in prediction can be categorized into 1) aleatoric uncertainty, that comes from inherent ambiguity in data, and 2) epistemic uncertainty, that comes from the model due to lack of data. Especially, the aleatoric uncertainty in speech recognition usually comes from labeling noise, variations among speakers, or confusing pronunciations. As an example, Fig. 1 shows an example of a recognition result, token-level class probabilities and aleatoric uncertainty given ground-truth tokens. When there is an ambiguous pronunciation in the input utterance, the recognition result shows the error at confusion pronunciation between HEAR and HOR. Moreover, the model outputs a low value of the class probability and a high value of uncertainty at the HEAR token.

The aleatoric type of uncertainty can be modeled explicitly with the network parameters so that we utilize this uncertainty to transfer additional knowledge of teachers. Specifically, following [13], the model predicts a logit vector $\mathbf{f}_i$ for each token, which forms a probability vector $\mathbf{p}_i$ after a softmax operation. In order to model token-wise aleatoric uncertainty, our network generates a variance vector $\boldsymbol{\sigma}_i^2$ whose dimension is the same as that of the logit vector $\mathbf{f}_i$. With this variance vector, the final logit vector is corrupted by the independent Gaussian noise:

$$\boldsymbol{\sigma}_i^2 = f\left(\mathrm{W}\mathbf{h}_i + b\right), \quad (3)$$

$$\mathbf{g}_i = \mathbf{f}_i + \boldsymbol{\sigma}_i^2 \cdot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I) \quad (4)$$

where $\mathbf{h}_i$ is the representation of token i at the previous layer of logit and $\boldsymbol{\sigma}_i^2$ is linearly projected given $\mathbf{h}_i$. For a nonlinear function f, we can use any function as long as it satisfies the constraints of $\boldsymbol{\sigma}_i^2 > 0$ like a ReLU. In practice, instead of (3), the network generates $\log\boldsymbol{\sigma}_i^2$ itself without using a nonlinear function. The corrupted vector $\mathbf{g}_i$ is then passed through the softmax function, as in the usual way:

$$\mathbf{p}_i = \mathrm{softmax}\left(\mathbf{g}_i\right). \quad (5)$$

3.2. Uncertainty-matching KD

We now provide our uncertainty based knowledge distillation approach when transferring knowledge from the teacher network to the student network. Finally, our *uncertainty-matching* KD transfers the knowledge present in the token-level noise vectors $\hat{\boldsymbol{\sigma}}_i$ and $\boldsymbol{\sigma}_i$ from the teacher and student networks using the normalized MSE function:

$$\mathcal{L}_{\text{unc-KD}} = \sum_i \left\| \frac{\hat{\boldsymbol{\sigma}}_i}{\|\hat{\boldsymbol{\sigma}}_i\|_2} - \frac{\boldsymbol{\sigma}_i}{\|\boldsymbol{\sigma}_i\|_2} \right\|^2 \quad (6)$$

The final distillation loss of the student model is the combination of all pieces above:

$$\mathcal{L} = \mathcal{L}_{\text{NLL}} + \alpha\mathcal{L}_{\text{KD}} \quad (7)$$

$$\text{where} \quad \mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{tok-KD}} + \beta\mathcal{L}_{\text{unc-KD}}. \quad (8)$$

Here, the hyperparameter α controls the relative importance between NLL and KD loss functions while β does that between tok-KD and unc-KD loss functions. In all our experiments, we fix $\beta = 1$ to simply the hyperparameter tuning process.

When multiple teacher networks are given, the student network is then trained to minimize the following loss:

$$\mathcal{L}_{\text{KD}} = \sum_{k=1}^{K} \gamma_k \mathcal{L}_{\text{KD}}^k. \quad (9)$$

Here, γ_k is a relative weight for the k-th teacher model. Typically, γ_k in (9) is set to a equal value without any a prior knowledge on the teacher model's domain. In our experiments, we set the same weight values for all k when KD training.

4. Experiments

In this section, we describe our experiments on LibriSpeech [15], Tedlium2 [16], and WSJ [14] corpus as multi-domain datasets. LibriSpeech is primarily composed of 960 hours of audio reading public domain books. And the Tedlium2 dataset is extracted from TED talks, which consists of 207 hours of audio. These two speech corpus were used for training the teacher networks. For WSJ, we use si-284 with approximately 81 hours

Table 1: *Transformer architectures of teacher and student networks. The number of hidden H, filter F and head h sizes are described.*

Model		Layers	$F/H/h$	Param.
Teacher	Libri	12/6	512/2048/8	70.7M
	Ted	12/6	256/2048/4	27.1M
Student	WSJ-base	12/6	256/2048/4	27.1M
	WSJ-small	8/4	256/2048/4	18.7M
	WSJ-tiny	6/2	256/1024/4	8.7M

Table 2: *Baseline performance in WERs [%] on nov93 dev and nov92 test sets of WSJ.*

Model	nov93	nov92
Libri	4.7	2.9
Ted	6.5	4.5
WSJ-base	7.8	4.9
WSJ-small	8.3	5.5
WSJ-tiny	8.4	5.6

of speech as the training set, nov93 as our development set and nov92 as our test set. The student networks were trained using WSJ corpus. In tables, we denote the LibriSpeech and Tedlium2 as Libri and Ted, respectively.

For all speech corpora, we represent input signals as a sequence of 80-dim log-Mel filter bank with 3-dim pitch features [17]. SentencePiece[18] is employed as the tokenizer, and we select 31 characters including `blank`, `unk` and `space` for output unit modeling. All configurations for student and teacher networks are described in Table 1. For a LM of WSJ, we adopted 12-layer Transformer LM with 512 filters, 2048 hidden units and 8 heads when decoding for all KD experiments. For training ASR models, we employed label smoothing of value as 0.15, and scheduled sampling [19] after 100k gradient updates. We used the Adam [20] optimizer with $\beta_1 = 0.9, \beta_2 = 0.98$ and $\epsilon = 10^{-9}$. The hyperparameters of SpecAugment [21] follow [22] for all ASR experiments. The speed perturbation is applied only for training the Tedlium2 ASR teacher model. We used the baseline models as the pre-trained model for KD experiments with the initial learning rate of 1.0. The weight parameter α for KD is set to 2.5 both for two ASR teachers. We average the 5 checkpoints yielding best recognition accuracies in token-level on validation sets as the final model. All the experiments are implemented based on ESPnet [22]. Overall baseline setup on WSJ experiments follows that of ESPnet. All models are jointly trained with a Connectionist Temporal Classification (CTC) [24] loss function with a weight 0.3. In joint decoding for KD experiments, we integrated the neural network-based LM with E2E ASR models by shallow fusion [23] which is a log-linear interpolation between token-level probabilities during beam search in addition to the CTC probabilities as in [22]. We set beam width, LM and CTC weights to 16, 0.85, and 0.3, respectively. We observed word error rates (WERs) as the evaluation metric.

4.1. Baseline performance

Table 2 shows the baseline performance of the teacher and student models in WERs. As the student model size decreased, the recognition performance was degraded more. When the compression ratio is 0.3 from WSJ-base to WSJ-tiny, the WER increased from 4.9% to 5.6% on nov92. The recognition performance of two teacher models outperforms the student baseline models. The Librispeech ASR teacher model shows the best WER of 2.9% on nov92. Moreover, the Tedlium2 ASR teacher model shows degraded but still competitive performance compared to the performance of the Librispeech teacher since the Tedlium2 teacher network is trained using a smaller size of speech corpus.

Table 3: *WERs [%] of proposed knowledge distillation methods on WSJ.*

Model	Teacher	Method	nov93	nov92
WSJ-base	-	-	7.8	4.9
	Libri	tok-KD	5.5	3.5
	Ted		5.7	3.7
	Libri, Ted		5.2	3.1
	Libri	+ unc-KD	5.2	3.0
	Ted		5.3	3.3
	Libri, Ted		5.0	3.1

Table 4: *Model compression results in WERs [%] on WSJ for different student model sizes.*

Model	Teacher	Method	nov93	nov92
WSJ-small	-	-	8.3	5.5
	Libri	tok-KD	5.7	4.0
		+ unc-KD	5.4	3.8
	Libri, Ted		5.1	3.8
WSJ-tiny	-	-	8.4	5.6
	Libri	tok-KD	7.4	5.1
		+ unc-KD	7.1	4.7
	Libri, Ted		6.3	4.3

4.2. Multi-domain uncertainty-matching KD

The effectiveness of the proposed knowledge distillation methods is shown in Table 3 and 4. Overall knowledge distillation results outperformed the student baseline models which are trained from scratch. Moreover, the proposed unc-KD method works well for overall KD experiments with teachers from different domains. In Table 3, the WER of tok-KD on nov92 was 3.1% when using both Librispeech and Tedlium2 teachers, yielding a relatively 36.7% reduction over the student baseline. Moreover, WERs of 3.0% and 3.3% on nov92 were achieved using unc-KD, yielding a relative improvements of 14.3% and 10.8% over the results of the KD method only with tok-KD using Librispeech and Tedlium2 teachers, respectively. This indicates that the effectiveness of transferred knowledge differs from teachers which are trained from different domains. When distilling both two teacher networks to transfer the output probability and uncertainty information, the final WERs of 5.0% and 3.1% on nov93 and nov92, yielding a relative improvements of 35.9% and 36.7% over the student baseline. However, the relative WER improvements over models with the tok-KD method are less than that over the baseline models.

Table 4 shows the overall recognition performance of student models which have smaller network sizes than the WSJ-base model. Despite the reduction in model size, WERs of the student models exceeded the baseline performance of WSJ-base in all cases after KD was performed. The final WERs of WSJ-small and WSJ-tiny models by the proposed unc-KD method

Table 5: *Effectiveness of uncertainty modeling for ASR and language models. WERs [%] on test-clean and test-other sets of Librispeech. We denote the Transformer and uncertainty as Trf. and Unc., respectively.*

Model		test-clean	test-other
Trf. ASR	-	3.5	8.3
+ Unc.	-	3.4	8.0
Trf. ASR	Trf. LM	2.5	5.6
+ Unc.	Trf. LM	2.5	5.6
Trf. ASR	+ Unc.	2.5	5.6
+ Unc.	+ Unc.	2.4	5.6

Table 6: *Comparison of WERs [%] on WSJ for different ASR systems. We denote the Transformer as Trf.*

Model	Params	LM	nov93	nov92
Deep Speech 2 [25]	52M	5-gram	-	3.6
Fully Conv. [28]	-	Conv	6.8	3.5
QuartzNet 5x3 [26]	6.4M	Trf.XL	7.0	4.5
15x5-transfer	18M	Trf.XL	**4.8**	**3.0**
Trf. Self-distil [11]	27.1M	Trf.	6.9	4.2
Proposed KD w/ multi-domain teachers				
Trf. WSJ-base	27.1M	Trf.	**5.0**	**3.1**
Trf. WSJ-small	18.7M	Trf.	5.1	3.8
Trf. WSJ-tiny	8.7M	Trf.	6.3	4.3

are 3.8% and 4.3% on nov92, yielding the relative WER improvements of 29.6% and 21.8% over each baseline, repectively. Compared to the method of tok-KD only, additional distilled knowledge from uncertainty also helps to improve the recognition performance, yielding the relative WER improvements of 5.0% and 7.8% on the WSJ-small and WSJ-tiny model when the single Librispeech teacher model is used. Finally, we emphasize that the final KD results on nov92 set for all student models outperformed the teacher baseline of Tedlium2.

4.3. Effect of uncertainty modeling

Table 5 shows the effect of uncertainty modeling on the ASR and language model. Overall results when the models are trained via modeling uncertainty outperformed the models without uncertainty. Especially, the Transformer ASR model with uncertainty shows the WER relative improvements of 2.8% and 3.6% on test-clean and test-other set, respectively, when decoding with no LM. Uncertainty modeling for both ASR and language models gives slightly better WER improvements. In Fig. 2, the probability and uncertainty distribution for each token are shown for Librispeech and Tedlium2 teacher models. The sub-figures in Fig. 2 is drawn using the "01fc020k" utterance in a train-si284 set of WSJ. The `space` and `eos` token in x-axis of each figure are denoted as _ and ., respectively. Interestingly, the output probability distribution of two teacher models has a similar shape, but the uncertainty distributions between two teacher models are quite different for each token.

4.4. Comparison

Table 6 summarizes WERs of prior works and our implementations on WSJ. Despite of the small-sized model such as WSJ-small and WSJ-tiny, the proposed KD method has competitive performance to the performance of previous works on WSJ. Deep Speech 2 [25] reports a WER of 3.6%, whose ASR model

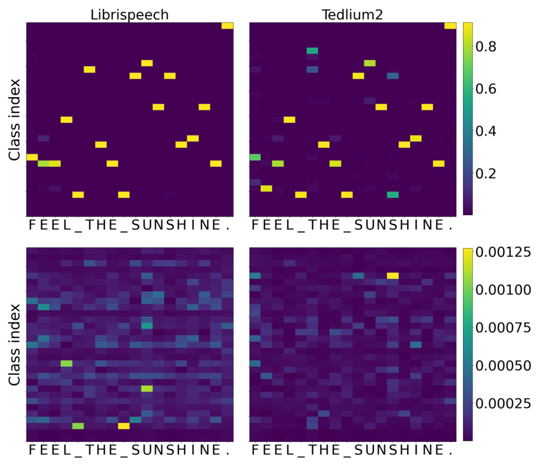


Figure 2: *Comparison of probability and uncertainty distributions between two ASR teachers. The top and bottom sub-figures show the probability and uncertainty distributions, respectively. The left and right sub-figures are drawn using Librispeech and Tedlium2 ASR teachers, respectively.*

is trained using more training data (150 times) for the ASR model, and abundant texts for LM training. And QuartzNet 15x5 with transfer learning [26] shows the best character-based performance on WSJ for a supervised learning task, by a WER 3.0% on nov92, which is pre-trained of 960-hours Librispeech and Mozillas Common Voice datasets, and then fine-tuned on WSJ. Our student model is purely trained using the 81-hour WSJ training set only, compared to other works, but utilizes the plentiful information from the teacher models which are trained using more training data such as Librispeech and Tedlium2 corpus. Note that Transformer-XL [27] has more network parameters compared to our Transformer LM. For the comparison of other KD works for ASR, our final model outperforms self-distillation [11], which utilizing the Transformer outputs and attention weights for making pseudo-targets to improve the encoding ability of speech frames. The WSJ-tiny model still has competitive performance of self-distillation [11] despite a small-size model of 8.7 million (M) parameters by showing a WER difference of 0.1% on nov92.

5. Conclusion

In this paper, we investigated uncertainty-based knowledge distillation method for the self-attention based E2E model to overcome the degraded performance of models with small amount of target training data and limited sizes. We matched the aleatoric uncertainty of the output sequences of the teacher model based on the ground-truth target sequences to the KD loss function. Moreover, we showed the effectiveness of multiple teachers from different domain and type. We demonstrated that the WERs were improved over the baseline on WSJ, clearly showing that the recognition performance was meaningful using the additional loss function with uncertainty and multiple teachers. Finally, we emphasize that the proposed KD method can be applied on other recognition systems that are capable of modeling uncertainty.

6. References

[1] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *NIPS workshop*, 2014.

[2] A. Romero, N. Ballas, S. E. Kahou, A. Chassang, C. Gatta, and Y. Bengio, "Fitnets: Hints for thin deep nets," in *ICLR*, 2015.

[3] S. Zagoruyko and N. Komodakis, "Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer," in *ICLR*, 2017.

[4] J. Yim, D. Joo, J. Bae, and J. Kim, "A gift from knowledge distillation: Fast optimization, network minimization and transfer learning," in *CVPR*, 2017.

[5] J. Kim, S. Park, and N. kwak, "Paraphrasing complex network: Network compression via factor transfer," in *In Advances in neural information processing systems*, 2018.

[6] B. Heo, J. Kim, S. Yun, H. Park, N. Kwak, and J. Y. Choi, "A comprehensive overhaul of feature distillation," in *ICCV*, 2019.

[7] H. Lee, S. J. Hwang, and J. Shin, "Rethinking data augmentation: Self-supervision and self-distillation," *arXiv:1910.05872*, 2019.

[8] R. Pang, T. N. Sainath, R. Prabhavalkar, S. Gupta, Y. Wu, S. Zhang, and C.-C. Chiu, "Compression of end-to-end models," in *INTERSPEECH*, 2018.

[9] M. Huang, Y. You, Z. Chen, Y. Qian, and K. Yu, "Knowledge distillation for sequence model," in *INTERSPEECH*, 2018.

[10] R. Takashima, S. Li, and H. Kawai, "An investigation of a knowledge distillation method for CTC acoustic models," in *ICASSP*, 2018.

[11] T. Moriya, T. Ochiai, S. Karita, H. Sato, T. Tanaka, T. Ashihara, and M. Delcroix, "Self-distillation for improving CTC-Transformer-based ASR systems," in *INTERSPEECH*, 2020.

[12] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *ICML*, 2016.

[13] A. Kendall and Y. Gal, "What uncertainties do we need in bayesian deep learning for computer vision?" in *In Advances in neural information processing systems*, 2016.

[14] D. B. Paul and J. M. Baker, "The design for the wall street journal-based CSR corpus," in *workshop on Speech and Natural Language*. Association for Computational Linguistics, 1992, p. 357–362.

[15] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[16] A. Rousseau, P. Delglise, and Y. Estve, "Enhancing the TED-LIUM corpus with selected data for language modeling and more TED talks," in *LREC*, 2014.

[17] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A pitch extraction algorithm tuned for automatic speech recognition," in *ICASSP*, 2014.

[18] T. Kudo and J. Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," in *ACL*, 2018.

[19] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled sampling for sequence prediction with recurrent neural networks," in *In Advances in neural information processing systems*, 2015.

[20] D. P. D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *ICL*, 2015.

[21] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," in *ICASSP*, 2014.

[22] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, and A. Renduchintala, "Espnet: End-to-end speech processing toolkit," in *INTERSPEECH*, 2018.

[23] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv:1503.03535*, 2015.

[24] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks." in *ICML*, 2006.

[25] D. Amodei, S. Ananthanarayanan, R. Anubhai, J. Bai, E. Battenberg, C. Case, J. Casper, B. Catanzaro, Q. Cheng, and G. Chen, "Deep speech 2: End-to-end speech recognition in english and mandarin," in *ICML*, 2016, p. 173–182.

[26] S. Kriman, S. Beliaev, B. Ginsburg, J. Huang, O. Kuchaiev, V. Lavrukhin, and Y. Zhang, "Quartznet: Deep automatic speech recognition with 1d time-channel separable convolutions," in *ICASSP*, 2020.

[27] Z. Dai, Z. Yang, Y. Yang, J. Carbonell, Q. V. Le, and R. Salakhutdinov, "Transformer-XL: Attentive language models beyond a fixed-length context," in *ACL*, 2019.

[28] N. Zeghidour, Q. Xu, V. Liptchinsky, N. Usunier, G. Synnaeve, and R. Collobert, "Fully convolutional speech recognition," *arXiv:1812.06864*, 2018.