Odyssey 2020 The Speaker and Language Recognition Workshop
1-5 November 2020, Tokyo, Japan

Joint Training End-to-End Speech Recognition Systems with Speaker Attributes

Sheng Li, Xugang Lu, Raj Dabre, Peng Shen, and Hisashi Kawai

National Institute of Information and Communications Technology, Kyoto, Japan

Abstract

The end-to-end (E2E) model allows for simplifying the conventional automatic speech recognition (ASR) systems. It integrates the acoustic model, lexicon, and language model into one neural network. In this paper, we focus on improving the performance of the state-of-the-art transformer-based E2E ASR system (ASR-Transformer). We propose to jointly train the compressed ASR-Transformer with speaker recognition (SR) tasks. As a common practice, speaker-ids are used for joint training the ASR and SR tasks. However, this leads to no significant improvement. To address this problem, we propose to augment the labels with meta-tags such as speaker attributes instead of simple speaker-ids. Experiments show that the proposed method can effectively improve the performance of compressed ASR-Transformer on CSJ corpus. Moreover, the proposed bags-of-attributes method has the potential to be used for building a highly customized ASR system.

1. Introduction

Conventional GMM-HMM [1] and DNN-HMM [2] based automatic speech recognition (ASR) systems require independently optimized components: acoustic model, lexicon and language model. The end-to-end (E2E) model integrates these components into a single neural network. It simplifies ASR system construction, solves the sequence labeling problem between variable-length speech frame inputs and label outputs (phone, character, syllable, word, etc.) and has achieved promising results on ASR tasks. Various types of E2E model have been studied in recent years: connectionist temporal classification (CTC) [3, 4], attention-based encoder-decoder (Attention) E2E models [5, 6], E2E lattice-free maximum mutual information (LFMMI) [7], and E2E models jointly trained with CTC and attention-based objectives (CTC/Attention) [8, 9, 10, 11].

Recently, the transformer [12] has been applied to E2E speech recognition systems [13, 14, 15, 16] and has achieved promising results. This transformer-based E2E ASR model (ASR-Transformer) entirely relies on attentional and feedforward components [12] to draw the contextual dependencies; this is a more aggressive technique compared with a time-delay neural network [17, 18]. Therefore, it can be trained faster with more parallelization, which is exactly what is required by the E2E models in ASR. However, this simple block-by-block stacking structure leads to severe problems. A deeper stacking structure may achieve better recognition results, but it increases the size of the model by a significant amount and also increases decoding latency.

In this paper, we propose a novel enhancement to substantially reduce the transformer model parameters by sharing parameters across layers. Reducing the number of parameters naturally leads to a drop in performance because the model might not be able to learn complex features. To overcome this we supply the model with meta information tokens during training that have been proven to be useful in the related field of natural language processing (NLP) [19]. The meta information we provide are speaker attributes (speaker information and speech utterance properties) that augment the training data and thereby enhance the performance of the compressed transformer-based E2E model as an extension of [20, 14]. All the experiments were performed on publicly available Japanese datasets (CSJ [21]).

The remainder of this paper is organized as follows. Section 2 introduces the compressed ASR-Transformer and its performance. Section 3 describes our proposed methods and provides experimental evaluations. Conclusions and future works are given in Section 4.

2. Previous Work on Compressed Transformer-based E2E ASR systems

2.1. Compressing Model Size of ASR-Transformer

The ASR-Transformer maps an input sequence, that is, the log-Mel filterbank feature, to a sequence of intermediate representations by the encoder. The decoder generates an output sequence of symbols (phones, syllables, characters, sub-words, or words) given the intermediate representations. The big difference between the ASR-Transformer and commonly used E2E models [5, 6] is that the ASR-Transformer completely relies on attention and feedforward components [12] as shown in Figure 1: multi-head self-attention (MHA), positional-encoding (PE), and position-wise feed-forward networks (PFFN). The blocks in the encoders and decoders are defined as follows:

1. The encoder-block has MHA and PFFN layers consecutively. Residual connections are used around each of the MHA and PFFN layers. Residual dropout [22] is introduced to each residual connection.
2. The decoder-block is similar to the encoder-block except for the addition of another MHA layer to perform attention over the output of the encoder-block stack.
3. PEs are added to the input at the bottom of these encoder-block and decoder-block stacks, providing information about the relative or absolute position of the tokens in the sequence as well as length.

The most prominent approach for reducing the size of a neural model is knowledge distillation [23], which requires training a parent model, which can be a time-consuming task. Work on zero-shot NMT [24] demonstrated that it is possible for multiple language pairs to share a single encoder and decoder without an appreciable loss in translation performance. However, this work did not consider sharing the parameters across

10.21437/Odyssey.2020-54

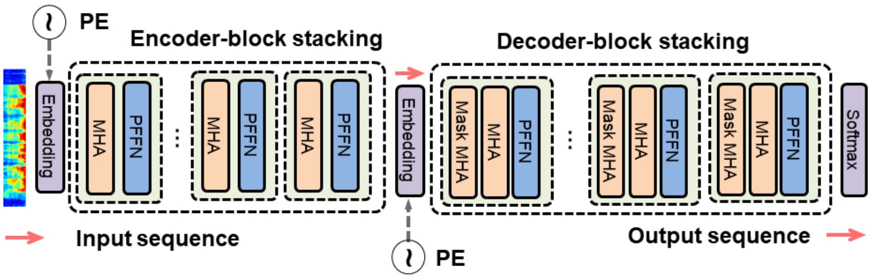

Figure 1: ASR-Transformer structure.

the stacked layers in the encoder or decoder. The work on universal transformer [25] demonstrated that feeding the output of the multi-layer encoder (and decoder) to itself repeatedly leads to an improvement for English-German translation. However, this work did not consider the accelerating speed of training and testing. Dabre et al. [26] proposed a novel modification to the NMT architecture, where parameters are shared across layers.

In the previous work [27], we share the parameters in the stacked structure to compress the model parameters as shown in Figure 2.

2.2. Data Descriptions and Uncompressed Baseline

In this work, we experiment on the "Corpus of Spontaneous Japanese (CSJ)" [21]. We used approximately 577 hours of lecture recordings as the training set (**CSJ-Train**) according to [28, 29, 11, 30]. Three official evaluation sets (**CSJ-Eval01**, **CSJ-Eval02**, and **CSJ-Eval03**), each containing ten lecture recordings [30], were used to evaluate the speech recognition results. Ten lecture recordings were selected for development (**CSJ-Dev**).

We used the implementation version-1.2.0 of the Transformer-based neural machine translation (NMT-Transformer) [12] in tensor2tensor [1] for all our experiments. The training and testing settings listed in Table 1 were similar to those in [16].

Table 1: Major Experimental Settings

Model structure			
Attention-heads	8	Decoder-blocks	6
Hidden-units	512	Residual-drop	0.3
Encoder-blocks	6	Attention-drop	0.0
Training settings			
Max-length	5000	GPUs (K40m)	4
Tokens/batch	10000	Warmup-steps	12000
Epochs	30	Steps	300000
Label-smooth	0.1	Optimizer	Adam
Testing settings			
Ave. chkpoints	last 20	Batch-size	100
Length-penalty	0.6	Beam-size	13
Max-length	50	GPUs (K40m)	4

Table 2: ASR performance (CER%) of the ASR-Transformer models trained with different units

Network	#unit	CER%			
		E01	E02	E03	Ave.
char	3178	8.2	5.9	6.6	6.9
word	98245	10.2	8.6	9.7	9.5
WPM	3000	8.4	**6.1**	6.3	6.9
	8000	**7.8**	**6.0**	**6.1**	**6.6**

Table 3: ASR performance (CER%) of the ASR-Transformer (WPM 8000) compared with other baselines

Network	#Para.	CER%			
		E01	E02	E03	Ave.
DNN-HMM-CE	38M	9.7	7.8	8.4	8.6
DNN-HMM-sMBR	38M	8.8	7.1	7.4	7.8
TDNN-LFMMI [33]	11M	8.3	6.6	6.5	7.1
BLSTM-CTC [34]	11M	9.4	7.3	7.5	8.1
Attention #w34330	22M	9.6	8.0	8.9	8.8
Attention #w3260	12M	9.4	7.3	7.5	8.1
CTC/Attention [8]	10M	8.4	**6.1**	6.9	7.1
ASR-Transformer	220M	**7.8**	**6.0**	**6.1**	**6.6**

We used 72-dim filterbank features (24-dim static $+\Delta$ $+\Delta\Delta$), which were mean and variance normalized per speaker, and four frames were spliced (four left, one current and zero right). Speed-perturbation [31] was not used to save training time. We trained the baseline ASR-Transformer models using CSJ-Train. For testing, we decoded the speech from test sets (CSJ-E01/02/03) and evaluated our models using the character error rate (CER%). Several modeling units were compared on Japanese ASR tasks as shown in Table 2, including words, word-piece-model (WPM)[32] and characters. We used the sentence-piece toolkit [2] as the sub-word segmenter. We used separate 3000 and 8000 sub-word vocabularies. The model trained with 8000 WPM sub-word vocabulary is statistically significantly better than the other models and will henceforth be considered as the baseline (the two-tailed t-test at p-value $<$ 0.05).

We also compared a set of other state-of-the-art systems with the ASR-Transformer (WPM 8000) as shown in Table 3.

The first three models are the hybrid DNN-HMM model and TDNN model. **DNN-HMM-CE** and **DNN-HMM-sMBR** use the same seven-layer DNN-HMM model (approximately 8500 senones) trained with cross-entropy (CE) and then state-

[1] https://github.com/tensorflow/tensor2tensor

[2] https://github.com/google/sentencepiece

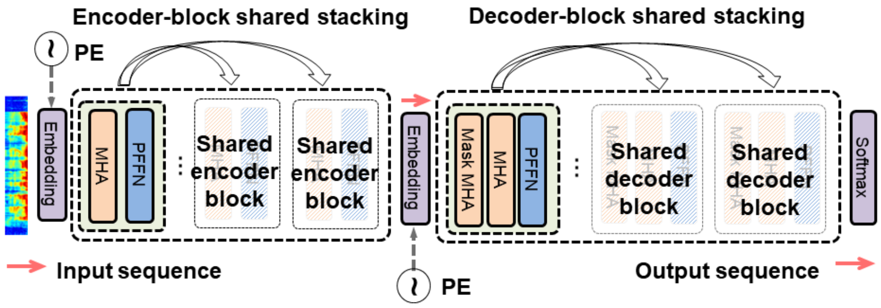

Figure 2: ASR-Transformer with parameter sharing.

level minimum Bayes risk (sMBR) criteria according to [34]. **TDNN-LFMMI** is the TDNN model (TDNN-a with approximately 3000 re-clustered senones) trained with LFMMI objective [33].

The second group are E2E models. **BLSTM-CTC** is the CTC model with five bi-directional LSTM (BLSTM) layers trained with 266 syllable-level labels given in [34]. **Attention** is the reproduction of the attention encoder-decoder E2E models implemented in [10] using a deeper (five-layer) encoder and two different word-level vocabulary sizes. **CTC/Attention** is the CTC and attention jointly trained model with 3315 characters as basic units implemented in [8] (MTL-large), and the structure is five BLSTM layers as an encoder and one LSTM layer as a decoder.

The DNN-HMM, TDNN and CTC models use 4-gram word language models in the WFST decoding framework. Other E2E models use one-pass decoding without language models. Detailed settings are available in related papers.

From Table 3, the ASR-Transformer significantly (the two-tailed t-test at p-value < 0.05) output-performs other models. This result is consistent with [16]. However, the size of this model is almost ten times larger than the other models. This is the biggest obstacle for model deployment, especially in memory-constrained devices.

2.3. Performances of Compressed Model

For an ASR-Transformer model with n-block stacking encoder and n-block stacking decoder ($n > 1$), we first create the parameters of the first encoder-block and first decoder-block. For other encoder-blocks and decoder-blocks, we reuse the parameters of the first encoder-block and decoder-block. To understand what each depth of stacking brings about, we trained and evaluated the following models:

1. Full-model: 1, 2, 3, 4, 5, 6 and 9-block (for both encoder and decoder) models without any shared parameters across layers.
2. Shared-model: 1, 2, 3, 4, 5, 6 and 9-block (for both encoder and decoder) models with parameters shared across all layers. These are referred to as 1-1, 2-2, 3-3, 4-4, 5-5, 6-6 and 9-9 by indicating the number of blocks in the encoder and decoder.

We trained all models with the default ADAM optimizer with learning rate warm-up and decay. The training procedure is the same regardless of whether we use layer sharing or not. It should be noted that in the case of full models, different sets of layer parameters receive a different number of updates but in the shared models a single set of layer parameters receive a large number of updates. It is possible that the large number of feedback signals received by the shared models causes them to approach the performance of the full models. We plan to verify this in the future.

Table 4: ASR performance (CER%) and real-time factor (RTF) of shared-models compared with full-models

Model (#Para.)	Blocks (n-n)	RTF (1 K40m)	CER%			
			E01	E02	E03	Ave.
Full (36M×n)	1-1	0.025	17.3	14.3	17.1	16.2
	2-2	0.043	10.7	8.5	8.9	9.4
	3-3	0.060	8.4	6.8	7.2	7.5
	4-4	0.078	8.5	**6.3**	6.7	7.2
	5-5	0.091	8.4	**6.1**	**6.4**	7.0
	6-6	0.115	**7.8**	**6.0**	**6.1**	**6.6**
	9-9	0.169	**7.7**	**5.7**	**6.1**	**6.5**
Shared (36M)	2-2	0.028	11.5	9.0	9.9	10.1
	3-3	0.034	10.5	7.8	**8.5**	8.9
	4-4	0.037	9.9	**7.5**	**8.5**	**8.6**
	5-5	0.046	**9.4**	**7.1**	**8.1**	**8.2**
	6-6	0.054	**9.5**	**7.3**	**8.3**	**8.4**
	9-9	0.074	9.9	**7.6**	**8.0**	**8.5**

The results compared with the 6-6 model in their own catergory (full or shared) without statistical significance (the two-tailed t-test at p-value < 0.05) are in bold font.

From Table 4, we found that both the full and shared-models can benefit from deeper structures. For the shared-model, the number of parameters was 36M (w/o output layer) regardless of how many blocks were in the stack. Additionally, the 6-6 shared-model corresponds to a six-times reduction in the number of parameters compared with the best vanilla 6-6 full-model (216M w/o output block). With the sacrifice of 1.6% averaged CER%, it decoded twice as fast with a single GPU. Using more than six blocks (for both encoder and decoder) did not achieve significant improvement and even had a negative influence on both types of models.

We observed approximately 10% acceleration in parallel training (8 K40m) of a shared-model compared with the full-model of the same depth. We expect that the shared-model with fewer parameters will converge faster than the full-model in small-data tasks such as WSJ (si84) or CHiME4.

Technically, sharing parameters does not reduce the amount of computation. Our hypothesis about faster decoding speed is that the weight matrix has a high chance of being cached into the GPU L1/L2 caches. In an uncompressed transformer, there are multiple weight matrices for each layer and will undoubt-

Table 5: ASR results (CER%) of 6-6 shared-models with different speaker attributes combinations on E01/E02/E03 (The results compared with the 6-6 full-model without significance difference at p-value < 0.05 are shown in bold font.)

Model	Combination of	ASR (CER%)			
	Speaker Attributes	E01	E02	E03	Ave.
Shared-model (6-6)	w/o attribute (same one in Table 4)	9.5	7.3	8.3	8.4
	SPK (traditional method)	11.2	8.8	9.3	9.8
	DIA	9.6	7.3	8.3	8.4
	DUR	8.4	**6.3**	6.8	7.2
	TOP	8.4	6.5	6.6	7.2
	SEX	**8.1**	**6.2**	6.6	7.0
	AGE	8.3	**6.3**	6.5	7.0
	EDU	8.5	6.4	6.7	7.2
	SEX+AGE	8.2	**6.3**	6.5	7.0
	SEX+DUR	**8.0**	**6.0**	6.6	**6.9**
	DUR+AGE	8.1	6.4	6.7	7.1
	DUR+TOP+SEX	8.3	6.4	6.6	7.1
	DUR+SEX+AGE	**7.9**	**6.3**	6.5	**6.9**
	DUR+TOP+SEX+AGE	**8.1**	**6.3**	**6.3**	**6.9**
	TOP+SEX+AGE+EDU	8.4	6.5	7.0	7.3
	DUR+SEX+AGE+EDU	8.3	6.7	6.5	7.2
	DUR+TOP+SEX+AGE+EDU	**8.1**	6.6	6.7	7.1
Full-model (6-6)	w/o attribute (same one in Table 4)	**7.8**	**6.0**	**6.1**	**6.6**
	SEX	**7.9**	**6.2**	**6.2**	**6.8**
	DUR+SEX+AGE	**7.6**	**6.1**	**6.3**	**6.7**

edly lead to a larger number of cache swaps which will negatively affect the speed. We verified the hypothesis that using the same parameters across all layers can speed up the training and decoding processes, which Dabre et al. [26] didn't report.

3. Improving the Compressed ASR-Transformer with Jointly Trained Speaker Recognition Tasks

3.1. Speaker Attributes Instead of Speaker IDs

Since there is a performance gap between the shared-model and the full-model, we propose to introduce a joint trained speaker recognition task to enhance the compressed transformer-based E2E model following [20, 14]. We use 1550 speaker IDs (**SPK**) for every sentence as a common practice, however, this leads to no significant improvement in both tasks as shown in Table 5.

Instead of using speaker IDs, we use speaker attributes (speaker information and speech utterance properties) and augment the training data to enhance the transformer-based E2E model as an extension of [20, 14]. The speaker attributes are defined as follows.

1. The dialect of speaker (**DIA**): Tokyo dialect (around Tokyo), Kansai dialect (around Osaka), Kyushu dialect (around Fukuoka), Tohoku dialect (around Aomori), San-yo dialect (around Okayama and Hiroshima, west of Osaka), and Unknown
2. Duration of the utterance (**DUR**): Short (up to 3 seconds), Long (more than 3 seconds)
3. Topic of the lecture (**TOP**): Academic, Simulated, Dialogue, Read, Misc, and Unknown
4. Sex of the speaker (**SEX**): Male, Female, and Unknown
5. Age of the speaker (**AGE**): Young (10-20s), Middle-age (30-50s), Old (60-80s), and Unknown
6. Education of the speaker (**EDU**): Middle-school, High-school, Bachelor, Master-Doctor, and Unknown

We feed these speaker attribute labels as a ground-truth in training, and the combinations of speaker attributes (e.g., <Male> <Long> <Master-Doctor>) are inserted to the beginning of the label of the training utterances (**label-based method**). The training labels are organized as "<S> <Male> <Long> <Master-Doctor> labels </S>". **The network is trained to output them at the beginning of decoding automatically, so we do not have to prepare classifiers for these attributes**.

We train both the 6-6 shared-model and 6-6 full-model with speaker attribute augmentation. As shown in Table 5, we first select the best single attributes, then use their combinations to get the best result. We find that the best single attribute is **SEX**. The **DUR+TOP+SEX+AGE** combination gives the best results on all of the test sets. However, considering the easiness of system construction, the **DUR+SEX+AGE** is adopted. We discovered that the decoding speed would not be influenced by using more attributes.

We also used the **DUR+SEX+AGE** and **SEX** as two kinds of attributes for training the 6-6 full-model. The proposed speaker attributes augmentation training was not effective for the 6-6 full model (bottom of Table 5). The full-sized model has a vast number of parameters to learn the speaker attributes by itself without explicitly telling them in the label. For the shared-model of much smaller size, feeding these speaker attribute labels as a ground-truth is effective in improving performance during decoding.

3.2. Extensive Investigations

Changing the Order of Attributes.

The order of the attributes are also significantly influence the ASR performances. Although these are bag-of-word features that are fed to the Transformer in sequence form.

Table 6: ASR results (CER%) of 6-6 shared-models with different speaker attributes combinations with different orders on E01/E02/E03

Model	Combination of	ASR (CER%)			
	Speaker Attributes with Different Orders	E01	E02	E03	Ave.
Shared-model (6-6)	w/o attribute (same one in Table 4)	9.5	7.3	8.3	8.4
	SEX+AGE	8.2	6.3	6.5	7.0
	AGE+SEX	8.2	6.4	6.5	7.0
	SEX+DUR	8.0	6.0	6.6	6.9
	DUR+SEX	8.3	6.4	6.3	7.0
	DUR+AGE	8.1	6.4	6.7	7.1
	AGE+DUR	8.1	6.3	6.5	7.0
	DUR+SEX+AGE	7.9	6.3	6.5	6.9
	DUR+AGE+SEX	8.0	6.3	6.7	7.0
	SEX+DUR+AGE	8.2	6.3	6.7	7.1
	SEX+AGE+DUR	8.3	6.5	6.7	7.2
	AGE+SEX+DUR	8.2	6.5	6.7	7.1
	AGE+DUR+SEX	8.0	6.1	6.4	6.8

Consequently, the order in which these features are provided do tend to have an impact on the final quality. Often, this impact is statistically significant (rows in Table 6). Although we do not have a clear understanding of why this happens, research in the related field of neural machine translation (NMT) [35] has shown that the position of meta-tags (such as AGE, SEX, etc.) impacts the encoder and decoder representations which then affects the final translation performance. This aspect has been discussed in [36]. In the future, we will consider merging these meta-tag embeddings with the embeddings of the actual text so as to make the decoder strongly aware of the meta-tag information.

Extension to feature-based method.

We inserted these attributes as one-hot values (e.g., 010 for <Male>, 100 for <Female>, 10010 for <Female> <Long>) at the beginning of the feature of the training utterances (**feature-based method**). This method can achieve close performance on average compared with inserting attributes in labels (**label-based method**) as shown in Table 7. This discovery means the proposed bags-of-attributes method has potential to be used for highly customized ASR system.

Table 7: ASR performance (CER%) of shared-models trained with attributes by label-based and feature-based methods.

Network	RTF	CER%			
	(1 K40m)	E01	E02	E03	Ave.
Full-model (w/o attributes)	0.115	7.8	6.0	6.1	6.6
Label-based (SEX+DUR)	0.057	8.0	6.0	6.6	6.9
Feature-based (SEX+DUR)	0.053	8.3	6.2	6.4	7.0

4. Conclusions and Future Work

In this paper, we proposed a novel method to efficiently train the ASR-Transformer by parameter sharing and introducing speaker attributes as meta-tags to augment the training data. The compressed and attributes-augmented trained model achieved results that are comparable to the full (uncompressed) model without any parameter sharing. In the future, we will perform an in-depth analysis of the proposed method, in addition to combining our methods with knowledge distillation approaches for high-performance compact modeling. We will also investigate the inner workings of parameter sharing so as to explain why it works despite having a substantially lower number of parameters.

5. Acknowledgements

We thank Dr. Naoyuki Kanda of Microsoft and Mr. Sei Ueno of Kyoto University for kindly providing results of the LFMMI and Attention models. This study is partially supported by JSPS KAKENHI Grant No. 19K24376 and NICT tenure-track startup fund "Research of advanced automatic speech recognition technologies", Japan.

6. References

[1] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1988.

[2] G. Dahl, D. Yu, L. Deng, and A. Acero, "Context dependent pre-trained deep neural networks for large vocabulary speech recognition," *IEEE Trans. ASLP*, vol. 20, no. 1, pp. 30–42, 2012.

[3] A. Graves and N. Jaitly, "Towards End-to-End speech recognition with recurrent neural networks," in *Proc. ICML*, 2014.

[4] Y. Miao, M. Gowayyed, and F. Metze, "EESEN: End-to-End speech recognition using deep RNN models and WFST-based decoding," in *Proc. IEEE-ASRU*, 2015, pp. 167–174.

[5] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NIPS*, 2015.

[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *Proc. IEEE-ICASSP*, 2016.

[7] H. Hadian, H. Sameti, D. Povey, and S. Khudanpur, "End-to-end speech recognition using lattice-free mmi," in *Proc. INTERSPEECH*, 2018.

[8] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. 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.

[9] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Proc. INTERSPEECH*, 2018.

[10] S.Ueno, H.Inaguma, M.Mimura, and T.Kawahara, "Acoustic-to-word attention-based model complemented with character-level ctc-based model," in *Proc. IEEE-ICASSP*, 2018.

[11] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in joint CTC-Attention based End-to-End speech recognition with a deep CNN Encoder and RNN-LM," in *Proc. INTERSPEECH*, 2017.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *CoRR abs/1706.03762*, 2017.

[13] L. Dong, S. Xu, and B. Xu, "Speech-transformer: A no-recurrence sequence-to-sequence model for speech recognition," in *Proc. IEEE-ICASSP*, 2018.

[14] S. Zhou, S. Xu, and B. Xu, "Multilingual end-to-end speech recognition with a single transformer on low-resource languages," in *CoRR abs/1806.05059*, 2018.

[15] S. Zhou, L. Dong, S. Xu, and B. Xu, "A comparison of modeling units in sequence-to-sequence speech recognition with the transformer on mandarin chinese," in *CoRR abs/1805.06239*, 2018.

[16] S.Zhou, L.Dong, S.Xu, and B.Xu, "Syllable-based sequence-to-sequence speech recognition with the transformer in mandarin chinese," in *Proc. INTERSPEECH*, 2018.

[17] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang, "Phoneme recognition using time-delay neural networks," *IEEE/ACM Trans. ASLP*, vol. 37, no. 3, pp. 328–339, 1989.

[18] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proc. INTERSPEECH*, 2015.

[19] Rico Sennrich, Barry Haddow, and Alexandra Birch, "Controlling politeness in neural machine translation via side constraints," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, San Diego, California, June 2016, pp. 35–40, Association for Computational Linguistics.

[20] B. Li and et al., "Multi-dialect speech recognition with a single sequence-to-sequence model," in *CoRR abs/1806.05059*, 2018.

[21] K. Maekawa, "Corpus of spontaneous japanese: Its design and evaluation," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[22] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Weinberger, "Deep networks with stochastic depth," in *Proc. ECCV*, 2016.

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NIPS*, 2015.

[24] M. Johnson and et al., "Google's multilingual neural machine translation system: Enabling zero-shot translation," in *CoRR abs/1611.04558*, 2016.

[25] M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser, "Universal transformers," in *CoRR abs/1807.03819*, 2018.

[26] R. Dabre and A. Fujita, "Recurrent stacking of layers for compact neural machine translation models," in *CoRR abs/1807.05353*, 2018.

[27] S. Li, R. Dabre, X. Lu, P. Shen, T. Kawahara, and H. Kawai, "Improving transformer-based speech recognition systems with compressed structure and speech attributes augmentation," in *Proc. INTERSPEECH*, 2019.

[28] T. Moriya, T. Shinozaki, and S. Watanabe, "Kaldi recipe for Japanese spontaneous speech recognition and its evaluation," in *Autumn Meeting of ASJ*, 2015, number 3-Q-7.

[29] N. Kanda, X. Lu, and H. Kawai, "Maximum a posteriori based decoding for CTC acoustic models," in *Proc. INTERSPEECH*, 2016, pp. 1868–1872.

[30] T. Kawahara, H. Nanjo, T. Shinozaki, and S. Furui, "Benchmark test for speech recognition using the corpus of spontaneous japanese," in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.

[31] T. Ko, V. Peddinti, D. Povey, and S. Khudanpur, "Audio augmentation for speech recognition," in *Proc. INTERSPEECH*, 2015.

[32] T. Kudo, "Subword regularization: Improving neural network translation models with multiple subword candidates," in *CoRR abs/1804.10959*, 2018.

[33] N. Kanda, Y. Fujita, and K. Nagamatsu, "Investigation of lattice-free maximum mutual information-based acoustic models with sequence-level kullback-leibler divergence," in *Proc. IEEE-ASRU*, 2017.

[34] N. Kanda, X. Lu, and H. Kawai, "Maximum-a-Posteriori-based decoding for End-to-End acoustic models," *IEEE/ACM Trans. ASLP*, vol. 25, no. 5, pp. 1023–1034, 2017.

[35] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, "https://arxiv.org/abs/1409.0473Neural Machine Translation by Jointly Learning to Align and Translate," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, USA, May 2015.

[36] Raj Dabre, Chenhui Chu, and Anoop Kunchukuttan, "A comprehensive survey of multilingual neural machine translation," 2020.