

# TRANSFER LEARNING OF LANGUAGE-INDEPENDENT END-TO-END ASR WITH LANGUAGE MODEL FUSION

Hirofumi Inaguma<sup>1\*</sup>, Jaejin Cho<sup>2</sup>, Murali Karthick Baskar<sup>3</sup>, Tatsuya Kawahara<sup>1</sup>, Shinji Watanabe<sup>2</sup>

<sup>1</sup>Graduate School of Informatics, Kyoto University, Kyoto, Japan

<sup>2</sup>Johns Hopkins University, Baltimore, MD, USA, <sup>3</sup>Brno University of Technology, Czech Republic

## ABSTRACT

This work explores better adaptation methods to low-resource languages using an external language model (LM) under the framework of transfer learning. We first build a language-independent ASR system in a unified sequence-to-sequence (S2S) architecture with a shared vocabulary among all languages. During adaptation, we perform *LM fusion transfer*, where an external LM is integrated into the decoder network of the attention-based S2S model in the whole adaptation stage, to effectively incorporate linguistic context of the target language. We also investigate various seed models for transfer learning. Experimental evaluations using the IARPA BABEL data set show that LM fusion transfer improves performances on all target five languages compared with simple transfer learning when the external text data is available. Our final system drastically reduces the performance gap from the hybrid systems.

**Index Terms**— end-to-end ASR, multilingual speech recognition, low-resource language, transfer learning

## 1. INTRODUCTION

Fast system development for low-resourced new languages is one of the challenges in automatic speech recognition (ASR). Recently, end-to-end ASR systems based on the sequence-to-sequence (S2S) architecture [1, 2] are filling up the gap of performance from the conventional HMM-based hybrid systems and showing promising results in many tasks with its extremely simplified training and decoding schemes [3–5]. This is very attractive when building systems in new languages quickly. However, models tend to suffer from the data sparseness problems in the low-resource scenario, especially in S2S models due to its data-driven optimization.

One possible approach to this problem is to utilize data of other languages. There are various approaches to leverage other languages: (a) to train a model multilingually (multi-task learning with other languages), and then further fine-tune to a particular language [6], and (b) to adapt a multilingual model to a new language using transfer learning [6–9] and additional features obtained from the multilingual model such as multilingual bottleneck features (BNF) [10–13] and language feature vectors (LFV) [14] (cross-lingual adaptation). To obtain a multilingual S2S model, a part of parameters can be shared while preparing the output layers per language [6], and we can further use a unified architecture with a shared vocabulary among multiple languages [15–17]. Since it would take much time to train such systems from scratch for many languages including new languages, we focus on the cross-lingual adaptation approach (b).

\*Part of the work reported here was conducted while the author was visiting Johns Hopkins University.

While a majority of the conventional transfer learning is concerned with acoustic model, using linguistic context during adaptation has not been investigated yet. The research question in this paper is: *Is linguistic context also helpful for adaptation to new languages?* The most common approach to integrate the external language model (LM) is referred to as *shallow fusion*, where LM scores are interpolated with scores from the S2S model [5, 18, 19]. Recently, several methods to leverage an external LM during training of S2S models are proposed: *deep fusion* [20] and *cold fusion* [21]. In deep fusion, the decoder network in the pre-trained S2S model and an external Recurrent neural network language model (RNNLM) are integrated into a single architecture by the gating mechanism and only the gating part is re-trained. In contrast, cold fusion integrates an external LM during the entire training stage.

In this paper, we investigate methods to fully utilize text data for adaptation to unseen low-resource languages. We propose *LM fusion transfer*, where an external LM is integrated into the decoder network of the S2S model only in the adaptation stage<sup>1</sup>, as an extension of cold fusion. Since the decoder network is already well-trained in a language-independent manner, the model can better incorporate linguistic context from the external LM. The extra cost to integrate the external LM during adaptation is trivial in the resource constrained condition. We also investigate various seed multilingual models trained with 600 to 2200-hours speech data and show the effect of the amount and variety of multilingual training data.

Experimental evaluations on the IARPA BABEL corpus show that the LM fusion transfer improves performance compared to simple transfer learning with shallow fusion when the additional text data is available. The performance of the transferred models is drastically improved by increasing the model capacity and incorporating the external LM, and the resulting models perform comparably with the latest BLSTM-HMM hybrid systems [10]. To our best knowledge, this is the first results for the S2S model to show the competitive performance to the conventional hybrid systems in the low-resource scenario (~50 hours).

## 2. RELATED WORK

The traditional usage of unpaired text data in the S2S framework is categorized to four approaches: LM integration, pre-training, multi-task learning (MTL), and data augmentation. In the LM integration approach, there are three methods: *shallow fusion*, *deep fusion*, and *cold fusion* as described in Section 1. Their differences are in the timing to integrate an external LM and the existence of additional parameters of the gating mechanism. We depict these fusion meth-

<sup>1</sup>Although we can perform LM fusion during training of the seed multilingual model, we focus on applying it during adaptation because our goal is to adapt it to a particular language rapidly.

ods in Fig. 1. In, [19], these fusion methods are compared in middle-size English conversational speech ( $\sim 300\text{h}$ ) and large-scale Google voice search data. However, none of previous works investigated the effect of them in other languages especially for low-resource languages, which is the focus of this paper. In [21], the authors show the effectiveness of cold fusion in a cross-domain scenario. Since the external LM is more likely to be changed frequently than the acoustic model, it is time-consuming to train a new S2S model with the LM integration from scratch every time the external LM is updated. In this work, we conduct LM fusion during adaptation to target languages, which is regarded as a more realistic scenario.

Another usage of the external LM is to initialize the lower layer in the decoder network with the pre-trained LM [19, 22]. However, we transfer almost all parameters in a multilingual S2S model (both encoder and decoder networks), and thus we do not explore this direction. Apart from the external LM, the MTL approach with LM objective are investigated in [19, 23]. Although the MTL approach does not require any additional parameters, it gets minor gains compared to LM fusion methods [19].

Recently, data augmentation of speech data based on text-to-speech (TTS) synthesis is investigated in the S2S framework [24, 25]. Since we are interested in the usage of linguistic context during adaptation, we leave this direction to the future work.

### 3. END-TO-END ASR

#### 3.1. Attention-based sequence-to-sequence

We build all models with attention-based sequence-to-sequence (S2S) models, which can learn soft alignments between input and output sequences of variable lengths [1]. They are composed of encoder and decoder networks. The encoder network transforms input features  $\mathbf{x} = (\mathbf{x}_1, \dots, \mathbf{x}_T)$  to a high-level continuous representation  $\mathbf{h} = (\mathbf{h}_1, \dots, \mathbf{h}_{T'})$  ( $T' \leq T$ ), interleaved with subsampling layers to reduce the computational complexity [26]. The decoder network generates a probability distribution  $P_{\text{S2S}}$  of the corresponding  $U$ -length transcription  $\mathbf{y} = (y_1, \dots, y_U)$  conditioned over all previous generated tokens:

$$\begin{aligned} \mathbf{s}_u^{\text{S2S}} &= \text{Decoder}(\mathbf{s}_{u-1}^{\text{S2S}}, y_{u-1}, \mathbf{c}_u) \\ P_{\text{S2S}}(\mathbf{y}|\mathbf{x}) &= \text{softmax}(\mathbf{W}^\circ \mathbf{s}_u^{\text{S2S}} + \mathbf{b}^\circ) \end{aligned}$$

where  $\mathbf{W}^\circ$  and  $\mathbf{b}^\circ$  are trainable parameters,  $\mathbf{s}_u^{\text{S2S}}$  is a decoder state at the  $u$ -th timestep, and  $\mathbf{c}_u$  is a context vector summarizing notable parts from the encoder states  $\mathbf{h}$ . We adopt the location-based scoring function [2]. To encourage monotonic alignments, the auxiliary Connectionist Temporal Classification (CTC) [27] objective is linearly interpolated [28].

During the inference stage, scores from the softmax layer used for the CTC objective are linearly interpolated in log-scale with a tunable parameter  $\lambda$  ( $0 \leq \lambda \leq 1$ ) to avoid generating incomplete and repeated hypotheses as follows [4]:

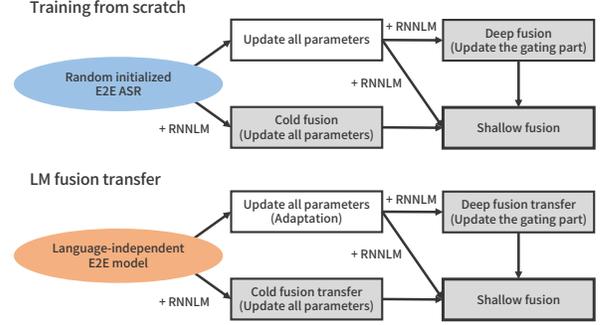
$$\ln P_{\text{ASR}}(\mathbf{y}|\mathbf{x}) = (1 - \lambda) \ln P_{\text{S2S}}(\mathbf{y}|\mathbf{x}) + \lambda \ln P_{\text{CTC}}(\mathbf{y}|\mathbf{x})$$

#### 3.2. LM fusion

##### 3.2.1. Shallow fusion

In the conventional decoding paradigm with an external LM, referred to as *shallow fusion*, scores from both the S2S model and LM are linearly interpolated to maximize the following criterion:

$$\mathbf{y}^* = \arg \max_{\mathbf{y} \in \Omega^*} \{\ln P_{\text{ASR}}(\mathbf{y}|\mathbf{x}) + \beta \ln P_{\text{LM}}(\mathbf{y})\}$$



**Fig. 1:** Overview of language model fusion transfer. LM fusion transfer is conducted with monolingual data only.

where  $\beta$  is a tunable parameter to define the importance of the external LM. The separate LM, especially trained with a larger external text, has complementary effects to the implicit LM modeled in the decoder network. Therefore, shallow fusion shows performance gains in many ASR tasks [5, 18, 19].

##### 3.2.2. Cold fusion (flat-start fusion)

While shallow fusion uses the external LM only in the inference stage, *cold fusion* [21] uses the pre-trained LM during training of the S2S model to provide effective linguistic context. The fine-grained element-wise gating function is equipped to flexibly rely on the LM depending on the uncertainty of prediction:

$$\begin{aligned} \mathbf{s}_u^{\text{LM}} &= \mathbf{W}^{\text{LM}} \mathbf{d}_u^{\text{LM}} + \mathbf{b}^{\text{LM}} \\ \mathbf{g}_u &= \sigma(\mathbf{W}^{\text{g}}[\mathbf{s}_u^{\text{S2S}}; \mathbf{s}_u^{\text{LM}}] + \mathbf{b}^{\text{g}}) \\ \mathbf{s}_u^{\text{CF}} &= \mathbf{W}^{\text{CF}}[\mathbf{s}_u^{\text{S2S}}; \mathbf{g}_u \odot \mathbf{s}_u^{\text{LM}}] + \mathbf{b}^{\text{CF}} \\ P_{\text{S2S}}(\mathbf{y}|\mathbf{x}) &= \text{softmax}(\text{ReLU}(\mathbf{W}^{\text{out}} \mathbf{s}_u^{\text{CF}} + \mathbf{b}^\circ)) \end{aligned}$$

where  $\mathbf{W}^*$  and  $\mathbf{b}^*$  are trainable parameters,  $\mathbf{d}_u^{\text{LM}}$  is a hidden state of RNNLM,  $\mathbf{s}_u^{\text{LM}}$  is a feature from the external LM,  $\mathbf{s}_u^{\text{CF}}$  is a bottleneck feature before the final softmax layer,  $\mathbf{g}_u$  is a gating function, and  $\odot$  represents element-wise multiplication. ReLU non-linear function is inserted before the softmax layer as suggested in [21]. We use the hidden state as a feature from RNNLM instead of logits because we use the universal character vocabulary for multilingual experiments, which results in the large softmax layer and increases the computational time [19].

In the original formulation in [19, 21], scores from the external LM are not used. We found that linear interpolation of log probabilities from the LM with those from the S2S model during the inference as in shallow fusion still has complementary effects to improve performance. Therefore, we adopt it in all experiments.

##### 3.2.3. Deep fusion (fine-tuning fusion)

*Deep fusion* [20] is another method to integrate an external LM during training. Unlike cold fusion, deep fusion is applied only for fine-tuning the gating part after parameters of both the pre-trained S2S model and RNNLM are frozen. Although deep fusion is formulated with a scalar gating function in [20], we use the same architecture as cold fusion in Section 3.2.2 to make a strict comparison. Then, the difference from the cold fusion is in the timing to integrate the external LM (from scratch or in the middle stage) and which parameters to update after integration (see Figure 1).

## 4. TRANSFER LEARNING OF MULTILINGUAL ASR

### 4.1. Adaptation to a target language

We adapt a seed language-independent end-to-end ASR model to an (unseen) target language. We investigate the following four scenarios:

**multi10:** From non-target 10 languages to an unseen target language

**high2:** From 2 high resource languages (English and Japanese) to an unseen target language

**multi10+high2:** From the mix of non-target 10 languages and 2 high resource languages to an unseen target language

**multi15:** From the mix of non-target 10 languages and target 5 languages to a particular target language

The top three conditions are regarded as cross-lingual adaptation.

### 4.2. LM fusion transfer

During adaptation, all parameters are copied from the seed language-independent S2S model, then training is continued toward a target language. We investigate improved adaptation methods by integrating the external LM during and/or after transfer learning from the seed model. Three methods are considered as follows:

**Transfer + SF:** Shallow fusion in Section 3.2.1 is conducted in the inference stage after adaptation.

**Cold fusion transfer (CF-transfer):** Cold fusion in Section 3.2.2 is conducted during adaptation. We integrate the external RNNLM from the start point of adaptation to a target language. The softmax layer is randomly initialized before adaptation due to the additional gating part.

**Deep fusion transfer (DF-transfer):** Deep fusion in Section 3.2.3 is conducted after adaptation. DF-transfer is composed of two stages: (1) adaptation by updating the whole parameters until convergence, and (2) fine-tuning only the gating part after integrating the external RNNLM. The softmax layer is randomly initialized before stage (2).

## 5. EXPERIMENTAL EVALUATION

### 5.1. Experimental setting

We used data from the IARPA BABEL project [29] and selected 10 languages as non-target languages for training the seed language-independent model: Cantonese, Bengali, Pashto, Turkish, Vietnamese, Haitian, Tamil, Kurmanji, Tokpisin and Georgian, and 5 languages for adaptation: Assamese (AS), Swahili (SW), Lao (LA), Tagalog (TA) and Zulu (ZU). Full language pack (FLP) is used for all experiments except for Section 5.2.3, where limited language pack (LLP) which consists of about 10% of FLP is used for adaptation. We sampled 10% of data from the training data for each language as the validation set. In addition, we used Librispeech corpus [30] and the Corpus of Spontaneous Japanese (CSJ) [31] as additional high resources.

We used Kaldi toolkit [32] for feature extraction. The input features were static 80-channel log-mel filterbank outputs appended with 3-dimensional pitch features computed with a 25ms window and shifted every 10ms. The features were normalized by the mean and the standard deviation on the whole training set. For the vocabulary, we used the universal character set including all characters from

**Table 1:** Results of baseline monolingual systems. None of adaptation methods is conducted.

Model	WER (%)				
	AS (54h)	SW (39h)	LA (58h)	TA (75h)	ZU (54h)
Old baseline [7]	73.9	66.5	64.5	73.6	76.4
New baseline	64.5	56.6	56.2	56.4	69.5
+ large units	59.9	50.9	51.7	52.7	65.5
+ shallow fusion	<b>57.4</b>	<b>46.5</b>	<b>49.8</b>	<b>49.9</b>	<b>62.9</b>
BLSTM-HMM	49.1	38.3	45.7	46.3	61.1

all languages [15], resulting in the vocabulary size of 5,353 classes including 17 language IDs, *sos*, *eos*, *unk*, and blank labels. For multilingual experiments, we prepended the corresponding language ID so that the decoder network can jointly identify the correct target language while recognizing speech [15].

Our encoder network is composed of two VGG-like CNN blocks [33] followed by a max-pooling layer with a stride of  $2 \times 2$ , and 5 layers of bidirectional long short-term memory (BLSTM) [34] with 1024 memory cells, which results in time reduction by a factor of 4. The decoder network consists of two layers of LSTM with 1024 memory cells. For both monolingual and multilingual experiments, we used the same architecture. Training was performed on the mini-batch size of 15 utterances using Adadelta [35] algorithm with an initial epsilon  $1e - 8$ . Epsilon was divided by a factor of 0.01 when the teacher-forcing accuracy does not improve for the validation set at each epoch. Scheduled sampling [36] with probability 0.4 and dropout for the encoder network with probability 0.2 were performed in all experiments during adaptation. We set the CTC weight during training and decoding to 0.5 and 0.3, respectively. We also set the beam width to 20 and the LM weight to 0.3.

For RNNLM, we used two layers of LSTM with 650 memory cells. All RNNLMs were trained with transcriptions in the parallel data except for experiments in Table 4. We used stochastic gradient descent (SGD) for RNNLM optimization. All networks are implemented by ESPnet toolkit [37] with pytorch backend [38].

### 5.2. Results

#### 5.2.1. Baseline monolingual systems for target 5 languages

First, we present the results of the baseline monolingual end-to-end systems in Table 1. Our new systems (line 2) significantly outperformed the old baseline reported on [7]. The gain mostly came from adding VGG blocks before BLSTM encoder and one more decoder LSTM layer though we also tuned other hyper-parameters. Next, changing the unit sizes of each LSTM layer from 320 to 1024 drastically improved the performance. This is surprising because increasing the number of parameters often makes the model overfit to the small amount of training data. Finally, shallow fusion with the monolingual RNNLM further boosted the performance although the RNNLM was trained with the small amount of transcriptions only. We use this setting as default in the rest of experiments.

We also built BLSTM-HMM hybrid systems for comparison. The BLSTM-HMM architecture includes 3 BLSTM layers each with 512 memory cells and 300 projection units<sup>2</sup>. The BLSTM acoustic model was trained using the latency control technique with 22 past frames and 21 future frames. The acoustic model receives 40-dimensional filterbank features as input. N-gram language model is built with the training transcriptions. WERs by our end-to-end sys-

<sup>2</sup>Increasing the unit size did not lead to any improvement.

**Table 2:** Results of adaptation from the different seed language-independent models. Shallow fusion with the corresponding monolingual RNNLM was conducted.

Seed	hours	WER (%)				
		AS	SW	LA	TA	ZU
multi10	643	53.4	41.3	46.1	46.4	60.2
high2	1,472	57.8	45.0	48.6	49.4	61.9
multi10+high2	2,115	<b>53.2</b>	40.7	45.1	<b>45.3</b>	<b>58.5</b>
multi15	929	53.4	<b>40.6</b>	<b>45.0</b>	46.1	58.8
multi15 w/o FT	929	56.2	44.2	47.1	47.8	60.6

(FT: fine-tuning to a target language)

tems with shallow fusion are close to those of the hybrid system, just 3.6 and 1.8 % absolute difference for Tagalog and Zulu, respectively.

### 5.2.2. Comparison of seed language-independent models

We compared the seed language-independent models for adaptation to target languages. All models were transferred, and shallow fusion with the corresponding monolingual RNNLM trained with the parallel data was performed. The results are shown in Table 2. The overall performance was significantly improved by transfer learning. The transferred S2S models achieved comparable WER to BLSTM-HMM for Tagalog and outperformed for Zulu in Table 1. We can see that *multi10* model is generally better than *high2* model despite the smaller data size, and combination of them (*multi10+high2*) gives slight improvement. On the other hand, *multi15* model that includes the target language does not lead to further improvement even after fine-tuning. We can conclude that the diversity of languages is more important than the total amount of training data, and 10 languages are almost sufficient for learning language-independent feature representation and generalized to other languages well [6]. Since *multi10* shows the competitive results to *multi10+high2* only with one third training data, we use *multi10* as the seed model and investigate cross-lingual adaptation in the following experiments.

### 5.2.3. Effect of LM fusion transfer

The results of our proposed LM fusion transfer are given in Table 3. When training S2S models from scratch, there is no difference among all fusion methods. When transferred from the language-independent S2S model, significant improvement is observed by integrating the external RNNLM. Shallow fusion was more effective than when training the S2S models from scratch in Table 1 because the multilingual training led to generalization and the affinity for the external LM was enhanced. CF-transfer got some improvements compared to transfer learning with shallow fusion for 3 target languages, but the effects of DF-transfer and CF-transfer are not significant. This is because RNNLMs were trained with text in the small parallel data only, therefore linguistic context during adaptation was not so effective. However, CF-transfer in Tagalog outperformed the monolingual hybrid system in Table 1. When compared to the previous work using the same data [7], CF-transfer yielded 21.6% gains relatively on average. Furthermore, 6.8% gains were achieved from transfer learning without the external LM.

To investigate the effect of additional text data, we evaluate the LM fusion transfer with LLP on each target language (~10 hours). The results are shown in Table 4. We used monolingual RNNLM trained with LLP (parallel data) and FLP (~50 hours), respectively. The latter setting of a small speech data set (~10 hours) and a larger text data set (~50 hours) is regarded as a more realistic scenario in low-resource languages. When training S2S models from scratch,

**Table 3:** Results of LM fusion transfer on FLP (~50h)

Model		WER (%)				
		AS	SW	LA	TA	ZU
Transfer [7]	SF	65.3	56.2	57.9	64.3	71.1
Scratch	—	59.9	50.9	51.7	52.7	65.5
	SF	57.4	46.5	49.8	49.9	62.9
	DF+SF	57.5	46.4	49.9	49.9	62.6
	CF+SF	57.5	47.3	50.0	50.2	62.9
Transfer (multi10)	—	56.4	46.4	48.6	50.1	63.5
	SF	<b>53.4</b>	41.3	46.1	46.4	60.2
	DF+SF	53.5	<b>41.2</b>	46.2	<b>46.2</b>	59.9
	CF+SF	53.6	41.6	<b>45.9</b>	<b>46.2</b>	<b>59.5</b>

(SF: shallow fusion, DF: deep fusion, CF: cold fusion)

**Table 4:** Results of LM fusion transfer on LLP (~10h)

Model		LM data	WER (%)				
			AS (8h)	SW (9h)	LA (9h)	TA (9h)	ZU (9h)
Scratch	—	—	not converge				
Transfer (multi10)	—	—	67.5	59.7	60.3	66.2	75.4
	SF	—	63.3	52.8	57.2	60.8	71.2
	DF+SF	LLP	68.0	52.4	57.3	60.7	70.9
	CF+SF	LLP	63.2	52.8	58.4	60.6	71.0
	SF	FLP	62.7	51.7	56.4	60.0	71.0
	DF+SF	FLP	66.8	50.7	56.1	60.0	69.9
	CF+SF		<b>61.7</b>	<b>50.3</b>	<b>56.0</b>	<b>57.9</b>	<b>69.8</b>

all models could not converge in our implementation even when reducing the unit sizes. The Babel corpus is mostly composed of conversational telephone speech (CTS), so it is difficult to optimize the S2S model from scratch with just around 10-hour training data. In the transfer learning approach, all three fusion methods got significant gains by using the external LM except for deep fusion in Assamese. For RNNLM trained with LLP, all fusion methods achieved a larger improvement than in Table 3. Interestingly, WER significantly dropped even when each RNNLM was trained with 10-hour data only. But all fusion methods show similar performance. In contrast, CF-transfer significantly outperformed simple transfer learning with shallow fusion on all 5 target languages when the RNNLM was trained with FLP, which is five-times larger than LLP. Therefore, we can conclude that linguistic context is helpful for adaptation when additional text data is available. This shows CF-transfer in Table 3 has the potential to surpass transfer learning with shallow fusion if we can access to additional text data<sup>3</sup>. In summary, CF-transfer yielded relative 10.4% and 2.3% gains on average compared to transfer learning without and with shallow fusion, respectively.

## 6. CONCLUSION

We explored the usage of linguistic context from the external LM during adaptation of the language-independent S2S model to target low-resource languages. We empirically compared various LM fusion methods and confirmed their effectiveness in resource limited situations. We showed that cold fusion transfer is more effective than simply applying shallow fusion after adaptation when additional text is available, which means linguistic context is also helpful in addition to acoustic adaptation. Our S2S model drastically closed the gap from the BLSTM-HMM hybrid system.

<sup>3</sup>Since *the provided data only* can be used for system training in BABEL rules, we do not explore to crawl text data from the WEB.

## 7. REFERENCES

- [1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio, “Neural machine translation by jointly learning to align and translate,” in *Proc. of ICLR*, 2015.
- [2] Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio, “Attention-based models for speech recognition,” in *Proc. of NIPS*, 2015, pp. 577–585.
- [3] Chung-Cheng Chiu, Tara N Sainath, Yonghui Wu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjali Kannan, Ron J Weiss, Kanishka Rao, Katya Gonina, et al., “State-of-the-art speech recognition with sequence-to-sequence models,” in *Proc. of ICASSP*, 2018, pp. 4774–4778.
- [4] Shinji Watanabe, Takaaki Hori, Suyoun Kim, John R Hershey, and Tomoki Hayashi, “Hybrid CTC/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [5] Albert Zeyer, Kazuki Irie, Ralf Schlüter, and Hermann Ney, “Improved training of end-to-end attention models for speech recognition,” in *Proc. of Interspeech*, 2018, pp. 7–11.
- [6] Siddharth Dalmia, Ramon Sanabria, Florian Metze, and Alan W Black, “Sequence-based multi-lingual low resource speech recognition,” in *Proc. of ICASSP*, 2018, pp. 4909–4913.
- [7] Jaemin 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 *Proc. of SLT*, 2018.
- [8] Sibong Tong, Philip N Garner, and Hervé Bourlard, “Multilingual training and cross-lingual adaptation on CTC-based acoustic model,” *arXiv preprint arXiv:1711.10025*, 2017.
- [9] Sibong Tong, Philip N Garner, and Hervé Bourlard, “Cross-lingual adaptation of a CTC-based multilingual acoustic model,” *Speech Communication*, vol. 104, pp. 39–46, 2018.
- [10] Martin Karafiat, Murali Karthick Baskar, Karel Vesely, František Grézl, Lukáš Burget, et al., “Analysis of multilingual lstm acoustic model on low and high resource languages,” in *Proc. of ICASSP*, 2018, pp. 5789–5793.
- [11] Karel Vesely, Martin Karafiat, František Grézl, Miloš Janda, and Ekaterina Egorova, “The language-independent bottleneck features,” in *Proc. of SLT*, 2012, pp. 336–341.
- [12] Frantisek Grézl and Martin Karafiat, “Adapting multilingual neural network hierarchy to a new language,” in *Spoken Language Technologies for Under-Resourced Languages*, 2014.
- [13] Ngoc Thang Vu, David Imseng, Daniel Povey, Petr Motlicek, Tanja Schultz, and Hervé Bourlard, “Multilingual deep neural network based acoustic modeling for rapid language adaptation,” in *Proc. of ICASSP*, 2014, pp. 7639–7643.
- [14] Markus Müller, Sebastian Stüker, and Alex Waibel, “Multilingual adaptation of RNN based ASR systems,” in *Proc. of ICASSP*, 2018, pp. 5219–5223.
- [15] Shinji Watanabe, Takaaki Hori, and John R Hershey, “Language independent end-to-end architecture for joint language identification and speech recognition,” in *Proc. of ASRU*, 2017, pp. 265–271.
- [16] Shubham Toshniwal, Tara N Sainath, Ron J Weiss, Bo Li, Pedro Moreno, Eugene Weinstein, and Kanishka Rao, “Multilingual speech recognition with a single end-to-end model,” in *Proc. of ICASSP*, 2018, pp. 4904–4908.
- [17] Suyoun Kim and Michael L Seltzer, “Towards language-universal end-to-end speech recognition,” in *Proc. of ICASSP*, 2018, pp. 4914–4918.
- [18] Anjali Kannan, Yonghui Wu, Patrick Nguyen, Tara N Sainath, Zhifeng Chen, and Rohit Prabhavalkar, “An analysis of incorporating an external language model into a sequence-to-sequence model,” in *Proc. of ICASSP*, 2017, pp. 5824–5828.
- [19] Shubham Toshniwal, Anjali Kannan, Chung-Cheng Chiu, Yonghui Wu, Tara N Sainath, and Karen Livescu, “A comparison of techniques for language model integration in encoder-decoder speech recognition,” *arXiv preprint arXiv:1807.10857*, 2018.
- [20] Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Hui-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio, “On using monolingual corpora in neural machine translation,” *arXiv preprint arXiv:1503.03535*, 2015.
- [21] Anuroop Sriram, Heewoo Jun, Sanjeev Satheesh, and Adam Coates, “Cold fusion: Training seq2seq models together with language models,” in *Proc. of Interspeech*, 2018, pp. 387–391.
- [22] Prajit Ramachandran, Peter J Liu, and Quoc V Le, “Unsupervised pre-training for sequence to sequence learning,” in *Proc. of EMNLP*, 2017.
- [23] Tobias Domhan and Felix Hieber, “Using target-side monolingual data for neural machine translation through multi-task learning,” in *Proc. of EMNLP*, 2017, pp. 1500–1505.
- [24] Tomoki Hayashi, Shinji Watanabe, Yu Zhang, Tomoki Toda, Takaaki Hori, Ramon Astudillo, and Kazuya Takeda, “Back-translation-style data augmentation for end-to-end ASR,” in *Proc. of SLT*, 2018, to appear.
- [25] Masato Mimura, Sei Ueno, Hirofumi Inaguma, Shinsuke Sakai, and Tatsuya Kawahara, “Leveraging sequence-to-sequence speech synthesis for enhancing acoustic-to-word speech recognition,” in *Proc. of SLT*, 2018, to appear.
- [26] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. of ICASSP*, 2016, pp. 4960–4964.
- [27] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, “Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks,” in *Proc. of ICML*, 2006, pp. 369–376.
- [28] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, “Joint CTC-attention based end-to-end speech recognition using multi-task learning,” in *Proc. of ICASSP*, 2017, pp. 4835–4839.
- [29] Mark JF Gales, Kate M Knill, Anton Ragni, and Shakti P Rath, “Speech recognition and keyword spotting for low-resource languages: BABEL project research at CUED,” in *Spoken Language Technologies for Under-Resourced Languages*, 2014.
- [30] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an ASR corpus based on public domain audio books,” in *Proc. of ICASSP*, 2015, pp. 5206–5210.
- [31] Kikuo Maekawa, “Corpus of Spontaneous Japanese: Its design and evaluation,” in *ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition*, 2003.
- [32] 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 *Proc. of ASRU*, 2011.
- [33] Karen Simonyan and Andrew Zisserman, “Very deep convolutional networks for large-scale image recognition,” *arXiv preprint arXiv:1409.1556*, 2014.
- [34] Sepp Hochreiter and Jürgen Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [35] Matthew D Zeiler, “Adadelta: an adaptive learning rate method,” *arXiv preprint arXiv:1212.5701*, 2012.
- [36] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer, “Scheduled sampling for sequence prediction with recurrent neural networks,” in *Proc. of NIPS*, 2015, pp. 1171–1179.
- [37] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplín, Jahn Heymann, Matthew Wiesner, Nanxin Chen, et al., “ESPnet: End-to-End Speech Processing Toolkit,” in *Proc. of Interspeech*, 2018, pp. 2207–2211.
- [38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer, “Automatic differentiation in PyTorch,” 2017.