INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

Analysis of Multilingual Sequence-to-Sequence Speech Recognition Systems

Martin Karafiát[1], *Murali Karthick Baskar*[1], *Shinji Watanabe*[2], *Takaaki Hori*[3], *Matthew Wiesner*[2], *Jan "Honza" Černocký*[1]

[1]Brno University of Technology
[2]John Hopkins University
[3]Mitsubishi Electric Research Laboratories (MERL)

karafiat@fit.vutbr.cz,baskar@fit.vutbr.cz,shinjiw@ieee.org
thori@merl.com,wiesner@jhu.edu,cernocky@fit.vutbr.cz

Abstract

This paper investigates the applications of various multilingual approaches developed in conventional deep neural network - hidden Markov model (DNN-HMM) systems to sequence-to-sequence (seq2seq) automatic speech recognition (ASR). We employ a joint connectionist temporal classification-attention network as our base model. Our main contribution is separated into two parts. First, we investigate the effectiveness of the seq2seq model with stacked multilingual bottle-neck features obtained from a conventional DNN-HMM system on the Babel multilingual speech corpus. Second, we investigate the effectiveness of transfer learning from a pre-trained multilingual seq2seq model with and without the target language included in the original multilingual training data. In this experiment, we also explore various architectures and training strategies of the multilingual seq2seq model by making use of knowledge obtained in the DNN-HMM based transfer-learning. Although both approaches significantly improved the performance from a monolingual seq2seq baseline, interestingly, we found the multilingual bottle-neck features to be superior to multilingual models with transfer learning. This finding suggests that we can efficiently combine the benefits of the DNN-HMM system with the seq2seq system through multilingual bottle-neck feature techniques.

Index Terms: multilingual ASR, sequence-to-sequence, language-transfer, multilingual bottle-neck feature

1. Introduction

The sequence-to-sequence (seq2seq) model proposed in [1–3] is based on deep neural network (DNN) and recurrent neural network (RNN) architectures for performing sequential prediction. Later, it was also adopted to perform automatic speech recognition (ASR) [4–6], as an alternative to traditional hidden Markov model (HMM)-based ASR [7, 8]. The model allows to integrate the main ASR blocks (acoustic, alignment, and language models) into a single neural network architecture. The recent success of connectionist temporal classification (CTC) [5,6] and attention based encoder-decoder [4,9] architectures generated significant interest in the speech community to study seq2seq models. However, outperforming conventional hybrid RNN/DNN-HMM models with seq2seq requires a huge amount of data [10–12]. Intuitively, this is due to the range of roles this model needs to perform: alignment and language modeling along with acoustics to character label mapping.

Another interesting research direction is multilingual ASR. Multilingual approaches have been mainly developed in hybrid RNN/DNN-HMM systems for tackling the problem of low-resource data. These include language adaptive training and shared layer retraining [13]. Parameter sharing investigated in our previous work [14] is one of the most beneficial techniques. This paper investigates the applications of such multilingual techniques developed in conventional RNN/DNN-HMMs to seq2seq ASR.

Existing multilingual approaches for seq2seq modeling mainly focus on CTC. A multilingual CTC proposed in [15] follows a similar protocol to DNN-HMM based on a universal phone set, finite state transducer (FST) decoder and language model. The authors also use the linear hidden unit contribution (LHUC) [16] technique to rescale the hidden unit outputs for each language as a way to adapt to a particular language. Another work [17] on multilingual CTC shows the importance of language adaptive vectors as an auxiliary input to the encoder in multilingual CTC model. An extensive analysis of multilingual CTC performance with limited data is done in [18] with a word level FST decoder integrated with CTC during decoding.

On a similar front, *attention models* were explored within a multilingual setup in [19, 20], where an attempt was made to build a single attention-based seq2seq model from multiple languages. Here, the multilingual training data is just pulled together assuming that the target languages are seen during training. Our prior study [21] extends this multilingual training and performs a preliminary investigation of transfer learning techniques to address the unseen languages during training. However, it is far from being exhaustive and does not cover the above mentioned various multilingual techniques.

In this paper, we fully make use of an experience in multilingual training [22, 23] developed for hybrid RNN/DNN-HMMs, and incorporate it into the seq2seq models. Especially, in our recent work of the RNN/DNN-HMM [23], we showed the multilingual acoustic models with transfer learning to be superior to multilingual bottle-neck features in RNN/DNN-HMM systems. We aim to design an experimental setup similar to [23] to investigate the effectiveness of the multilingual training on a seq2seq scheme. The main motivation and contribution of this work are:

- Incorporating the existing multilingual approaches in a seq2seq model.
- Comparing various multilingual approaches: especially multilingual bottle-neck features vs. transfer learning.

2. Sequence-to-Sequence Model

In this work, we use the attention based approach [2] as it provides an effective methodology to perform sequence-to-

Copyright © 2019 ISCA

http://dx.doi.org/10.21437/Interspeech.2019-2355

sequence training. Considering the limitations of attention in performing monotonic alignment [24, 25], we choose to use CTC loss function to aid the attention mechanism in both training and decoding.

Let $X = (\mathbf{x}_1, \mathbf{x}_2, \cdots)$ be a speech feature sequence and $C = (c_l | l = 1, \ldots, L)$ be an L-length grapheme sequence. A multitask learning framework $\mathcal{L}_{\text{mtl}}$ [26, 27] is used in this work to unify attention loss $p_{\text{att}}(C|X)$ and CTC loss $p_{\text{ctc}}(C|X)$ with a linear interpolation weight λ, as follows:

$$\mathcal{L}_{\text{mtl}} = \lambda \log p_{\text{ctc}}(C|X) + (1-\lambda) \log p^*_{\text{att}}(C|X). \quad (1)$$

The unified model benefits from both effective sequence level training and the monotonic alignment property enforced by the CTC loss.

$p_{\text{att}}(C|X)$ in Eq. (1) represents the posterior probability of character label sequence C w.r.t input sequence X based on the attention approach, which is decomposed with the probabilistic chain rule, as:

$$p^*_{\text{att}}(C|X) \approx \prod_{l=1}^{L} p\left(c_l | c^*_1, ..., c^*_{l-1},\ X\right), \quad (2)$$

where c^*_l denotes the ground truth history. Detailed explanation of the attention mechanism is given later.

Similarly, $p_{\text{ctc}}(C|X)$ in Eq. (1) represents the CTC posterior probability, which is factorized based on the conditional independence assumption as follows:

$$p_{\text{ctc}}(C|X) = \sum_{Z \in \mathcal{Z}(C)} p(Z|X) \approx \sum_{Z \in \mathcal{Z}(C)} \prod_{t=1}^{T} p(z_t|X), \quad (3)$$

where $Z = (z_t | t = 1, \ldots, T)$ is a CTC state sequence composed of the original grapheme set and the additional blank symbol. t and T denote frame index and input length after subsampling in the encoder, respectively. $\mathcal{Z}(C)$ is a set of all possible sequences given the character sequence C.

The following sections explain the encoder, attention decoder, and CTC used in our approach.

2.1. Encoder

In our approach, both CTC and attention use the same encoder function:

$$\mathbf{h}_t = \text{Encoder}(X), \quad (4)$$

where $\mathbf{h}_t$ is an encoder output state at subsampled frame t. As Encoder$(\cdot)$, we use a bidirectional long short-term memory (BLSTM) RNN .

2.2. Attention Decoder

Location-aware attention mechanism [28] is used in this work. The output of location-aware attention is:

$$a_{lt} = \text{LocAttention}\left(\{a_{l-1,t'}\}_{t'=1}^{T}, \mathbf{q}_{l-1}, \mathbf{h}_t; \Theta^{\text{att}}\right). \quad (5)$$

Here, a_{lt} acts as attention weight, $\mathbf{q}_{l-1}$ denotes the decoder hidden state introduced later, and $\mathbf{h}_t$ is the encoder hidden state obtained in Eq. (4). The location-attention function has trainable parameters Θ^{att} composed of 1) the convolution filters for the attention weight $a_{l-1,t}$ in the previous time step $l-1$, and 2) the linear transformation matrices to project the hidden states and convoluted attention weights to compute the new attention weight a_{lt}.

Finally, the context vector $\mathbf{r}_l$ is obtained as a weighted sum of the encoder output states $\mathbf{h}_t$ over all frames, with the attention weight a_{lt} obtained from Eq. (5):

$$\mathbf{r}_l = \sum_{t=1} a_{lt} \mathbf{h}_t. \quad (6)$$

The decoder function is based on unidirectional LSTM and character embedding layers. This function outputs the next decoder hidden state $\mathbf{q}_l$ from its previous label c_{l-1}, decoder hidden state $\mathbf{q}_{l-1}$ and attention output $\mathbf{r}_l$ obained from Eq. (6):

$$\mathbf{q}_l = \text{Decoder}(\mathbf{r}_l, \mathbf{q}_{l-1}, c_{l-1}). \quad (7)$$

Finally, the decoder hidden state $\mathbf{q}_l$ is used to compute the posterior distribution of a label c_l by using the linear transformation layer Linear$(\cdot)$ with trainable parameters Θ^{out} and standard softmax$(\cdot)$ operation as follows:

$$p\left(c_l | c_1, ..., c_{l-1},\ X\right) = \text{softmax}(\text{Linear}(\mathbf{q}_l; \Theta^{\text{out}})) \quad (8)$$

This equation is incrementally applied to form p^*_{att} in Eq. (2).

2.3. Connectionist temporal classification (CTC)

Unlike the attention approach, CTC does not use any specific decoder network, but directly computes the posterior distribution $p(z_t|X)$ in Eq. (3) from the encoder output state $\mathbf{h}_t$ in Eq. (4) as follows:

$$p(z_t|X) = \text{softmax}(\text{Linear}(\mathbf{h}_t; \Theta^{\text{ctc}})), \quad (9)$$

where the linear transformation layer Linear$(\cdot)$ has additional trainable parameters Θ^{ctc}.

Our prior study in [23] revealed the importance of the linear transformation parameters in RNN-HMMs before the softmax operation compared with the LSTM parameters in the multilingual transfer learning scenario. In seq2seq, these correspond to Θ^{out} and Θ^{ctc} introduced in Eqs. (8) and (9), and we focus on the effectiveness of transfer learning with Θ^{out} and Θ^{ctc}. Additionally, we also investigate the effectiveness of the attention parameters Θ^{att} introduced in Eq. (5).

3. Experiments

3.1. Data

The experiments are conducted using the BABEL speech corpus collected during the IARPA Babel program. Table 1 presents the details of the languages used for training and evaluation in this work. We split the language into 10 train ("seen") and 2 target ("unseen") languages to deal with challenging unseen language ASR. However, we also decided to evaluate on training languages to see the effect of multilingual training on both train and target languages. Therefore, **TokPisin** and **Georgian** from the train language set are treated as "seen" languages and **Assamese** and **Swahili** from the target language set as "unseen" languages and used for evaluation.

3.2. Sequence to sequence model setup

The baseline systems are built on 80-dimensional Mel-filter bank (fbank) features extracted using a sliding window of size 25 ms with 10ms stride. KALDI toolkit [29] is used to perform the feature processing. The "fbank" features are then fed to a seq2seq model with the following configuration:

The Bi-RNN [30] models mentioned above uses an bidirectional LSTM [31] cell followed by a projection layer (BLSTMP). The encoder consists of 6 BLSTMP layers with

Table 1: *Details of the BABEL data used for experiments.*

Usage	Language	Train		Eval		# of characters
		# spkrs.	# hours	# spkrs.	# hours	
Train	Cantonese	952	126.73	120	17.71	3302
	Bengali	720	55.18	121	9.79	66
	Pashto	959	70.26	121	9.95	49
	Turkish	963	68.98	121	9.76	66
	Vietnamese	954	78.62	120	10.9	131
	Haitian	724	60.11	120	10.63	60
	Tamil	724	62.11	121	11.61	49
	Kurdish	502	37.69	120	10.21	64
	Tokpisin	482	35.32	120	9.88	55
	Georgian	490	45.35	120	12.30	35
Target	Assamese	720	54.35	120	9.58	66
	Swahili	491	40.0	120	10.58	56

320 memory cells. The output frame rate is reduced by stacking and sub-sampling by factor two in second and third BLSTMP layer (4x in total). The location attention is used and the decoder has a single LSTM layer with 300 cells. In our experiments below, we only use a character-level seq2seq model based on CTC and attention. Thus, in the following experiments, we will use character error rate (%CER) as a suitable measure to analyze the model performance. All models are trained in ESPnet, end-to-end speech processing toolkit [32].

3.3. Multilingual features

Multilingual features are trained separately from seq2seq model according to a setup from our previous RNN/DNN-HMM work [23]. It allows us to easily combine the traditional RNN/DNN-HMM techniques such as GMM based alignments for DNN target estimation, phoneme units and frame-level randomization with the seq2seq model. Such multilingual features incorporate additional knowledge from non-target languages into features which should better guide the model.

3.3.1. Stacked Bottle-Neck feature extraction

The original idea of stacked bottle-neck feature extraction is described in [33]. The scheme consists of two DNN stages. The input features in the first stage DNN are 24 log Mel filterbank coefficients concatenated with fundamental frequency features. Conversation-side based mean subtraction is applied and 11 consecutive frames are stacked. Hamming window followed by discrete cosine transform (DCT) retaining 0^{th} to 5^{th} coefficients are applied on the time trajectory of each parameter resulting in 37×6=222 coefficients at the first-stage DNN input. In this work, the first-stage DNN has 4 hidden layers with 1500 units in each except the bottle-neck (BN) one. The BN layer has 80 neurons. The neurons in the BN layer have linear activations as found optimal in [34]. 21 consecutive frames from the first-stage DNN are stacked, down-sampled (each 5th frame is taken) and fed into the second-stage DNN with an architecture similar to the first-stage DNN, except for BN layer with only 30 neurons. Both neural networks were trained jointly as suggested in [34] in CNTK toolkit [35] with a block-softmax final layer [36]. Context-independent phoneme states are used as the training targets for the feature-extraction DNN, otherwise the size of the final layer would be prohibitive.

Finally, the BN outputs from the second stage DNN are used as our features for further experiments and will be called as "Mult11-SBN".

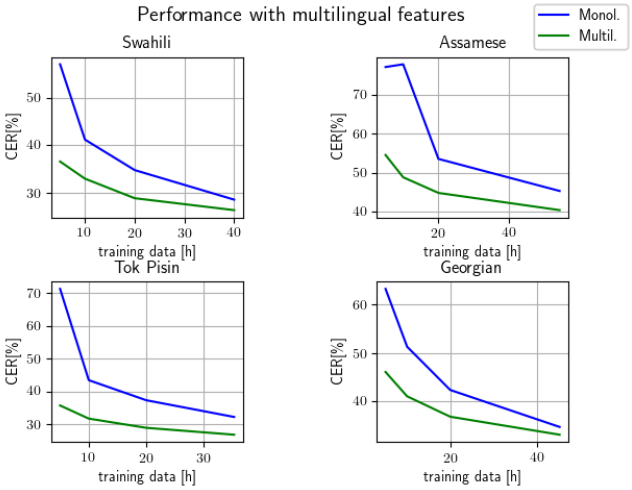

Figure 1: *Performance of Monolingual models trained with multilingual features on different amount of data.*

Table 2: *Comparison of Monolingual models trained on top of multilingual features and fbank features.*

Features	Swahili %CER	Assamese %CER	Tok Pisin %CER	Georgian %CER
FBANK	28.6	45.3	32.2	34.8
Mult11-SBN	**26.4**	**40.4**	**26.8**	**33.2**

3.3.2. Results

Figure 1 presents the performance of the seq2seq model depending on the amount of training data. Four (two "seen" and two "unseen", as introduced in Section 3.1) languages are analyzed. When the amount of training data is lowered, the performance with "fbank" features is very poor. On the other hand, the multilingual features provide:

- Huge improvement (at most 30% absolute) on small amounts of training data from the "fbank" baseline.
- Consistent improvement on both train (seen) and target (unseen) languages even when we only use train (seen) languages for feature extractor training.
- Significant improvement, 1.6%-5.0% absolute (see Table 2), even on the full training set.

3.4. Multilingual models

Next, we focus on the training of multilingual seq2seq models. For this experiment, we prepared two character dictionaries, one is created by augmenting all characters in both "seen" and "unseen" languages (**AllDic**), while the other is created by only using the "seen' languages (**TrainDic**) in Table 1. The model is trained in the same way as monolingual one but with all training data in the "seen" languages.

3.4.1. Multilingual seq2seq fine-tuing

Our multilingual seq2seq with **TrainDic** can potentially output the text of any language from "seen" by automatically handling language identification [19]. However, this language identification is not always perfect, and the model could also output a sequence of a different language character set. Adding language identification information as an additional feature, similarly to [37], may avoid the issue, but it also complicates the system. Instead, we performed experiments with fine-tuning of the

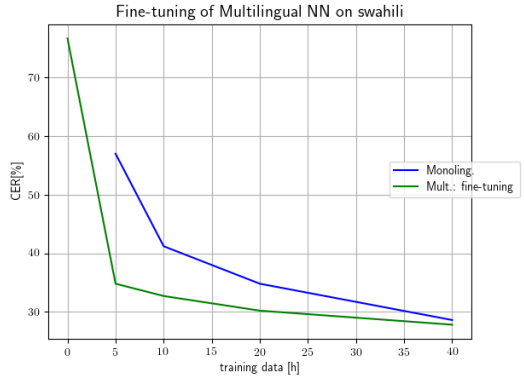

Figure 2: *Fine-tuning of multilingual seq2seq on Swahili.*

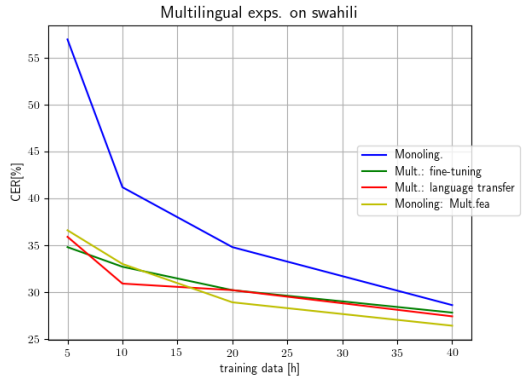

Figure 3: *Comparison of various multilingual approaches on Swahili.*

system into the TokPisin and Georgian languages in the "seen" language set by running a few epochs only (early stopping) on the desired language data. This strengthens the network to output their language characters, therefore it makes the system less prone to the above language- and character-set-mismatch errors.

The first two rows of table 3 present significant performance degradation from the monolingual to multilingual seq2seq models caused by wrong decision of output character set in about 20% of test utterances. However, no out-of-language characters are observed after "fine-tuning" and 1.5% and 4.7% improvement over monolingual baseline is reached.

A multilingual seq2seq can be fine-tuned to an "unseen" language with **AllDic**. Figure 2 shows the results on Swahili, which is not part of the multilingual training. Similarly to experiments with multilingual features in Figure 1, the fine-tuned multilingual seq2seq systems are effective especially on small amounts of data, but also defeat baseline models on the full ∼40h language set.

Table 3: *Multilingual fine tuning of seq2seq model.*

Model	TokPisin %CER	Georgian %CER
Monolingual	32.2	34.8
Multilingual	37.2	51.1
Multilingual fine-tuned	**27.5**	**33.3**

Table 4: *Multilingual Language Transfer*

Language Transfer	Swahili %CER	Assamese %CER	Tok Pisin %CER	Georgian %CER
Monoling.	28.6	45.3	32.2	34.8
Out	**27.4**	**41.2**	**27.7**	**33.6**
+Att	27.5	41.2	28.3	34.2
CTC+Out	27.6	41.2	27.9	33.7
+Att	28.0	42.1	27.6	34.1

3.4.2. Language-Transfer learning

Language-Transfer learning is necessary if the "unseen" language character set is different from the "seen" language set ones. The whole process can be described in three steps: 1) randomly initialize the pre-softmax layer parameters for the "unseen" language, 2) only train these new parameters and freeze the remaining ones 3) fine-tune the whole seq2seq. Various experiments are conducted by focusing on pre-softmax layer parameters in CTC (Θ^{ctc} in Eq. (9)) and decoder output (Θ^{out} in Eq. (8)). Additionally, we also investigate the effectiveness of the attention parameters Θ^{att} introduced in Eq. (5). Table 4 compares all combinations and clearly shows that retraining of the decoder pre-softmax only (Θ^{out}) gives the best results.

Finally, we compare the result with the multilingual features for the seq2seq model, as discussed in Section 3.3, and the previous result with the language transfer learning of multilingual seq2seq model in Figure 3. Interestingly, on contrary to our previous observations on DNN-HMM systems [23], we found multilingual features superior to language transfer learning in seq2seq model case.

With this result, we could conclude that the use of the multilingual feature is a quite effective way when we make use of the benefit of DNN-HMMs for the seq2seq model. This is the most important finding among our extensive analysis on the multilingual experiments.

4. Conclusions

We have presented various multilingual approaches in seq2seq systems including multilingual features and multilingual models by leveraging our multilingual DNN-HMM expertise. Unlike DNN-HMM systems [23], we obtain the opposite conclusion: multilingual features are more effective in seq2seq systems. It is probably due to efficient fusion of two complementary approaches: explicit GMM-HMM alignment incorporated in BN features and seq2seq models in the final system. With this finding, we will further explore efficient combinations of the DNN-HMM and seq2seq systems as our future work.

5. Acknowledgements

The work reported here was carried out during the 2018 Jelinek Memorial Summer Workshop on Speech and Language Technologies, supported by Johns Hopkins University via gifts from Microsoft, Amazon, Google, Facebook, and MERL/Mitsubishi Electric. It was also supported by Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4Innovations excellence in science - LQ1602" and by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA) MATERIAL program, via Air Force Research Laboratory (AFRL) contract # FA8650-17-C-9118. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ODNI, IARPA, AFRL or the U.S. Government

6. References

[1] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *NIPS*, 2014, pp. 3104–3112.

[2] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[3] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *NIPS*, 2015, pp. 577–585.

[5] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks." in *ICML*, vol. 14, 2014, pp. 1764–1772.

[6] A. Graves, "Supervised sequence labelling with recurrent neural networks," *URL http://books. google. com/books*, 2012.

[7] F. Jelinek, *Statistical methods for speech recognition*. MIT press, 1997.

[8] G. Hinton, L. Deng, D. Yu, G. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, B. Kingsbury *et al.*, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal processing magazine*, vol. 29, 2012.

[9] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *ICASSP*, 2016, pp. 4960–4964.

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

[11] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *ICASSP*, 2017, pp. 5280–5284.

[12] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," in *ICASSP*, 2018, pp. 4774–4778.

[13] S. Tong, P. N. Garner, and H. Bourlard, "An investigation of deep neural networks for multilingual speech recognition training and adaptation," Tech. Rep., 2017.

[14] M. Karafiát, M. K. Baskar, P. Matějka, K. Veselỳ, F. Grézl, and J. Černocky, "Multilingual blstm and speaker-specific vector adaptation in 2016 BUT Babel system," in *SLT*, 2016, pp. 637–643.

[15] S. Tong, P. N. Garner, and H. Bourlard, "Multilingual training and cross-lingual adaptation on CTC-based acoustic model," *arXiv preprint arXiv:1711.10025*, 2017.

[16] P. Swietojanski and S. Renals, "Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models," in *SLT*, 2014, pp. 171–176.

[17] M. Müller, S. Stüker, and A. Waibel, "Language adaptive multilingual CTC speech recognition," in *International Conference on Speech and Computer*. Springer, 2017, pp. 473–482.

[18] S. Dalmia, R. Sanabria, F. Metze, and A. W. Black, "Sequence-based multi-lingual low resource speech recognition," in *ICASSP*, 2018, pp. 4909–4913.

[19] S. Watanabe, T. Hori, and J. R. Hershey, "Language independent end-to-end architecture for joint language identification and speech recognition," in *ASRU*, 2017, pp. 265–271.

[20] S. Toshniwal, T. N. Sainath, R. J. Weiss, B. Li, P. Moreno, E. Weinstein, and K. Rao, "Towards language-universal end-to-end speech recognition," in *ICASSP*, 2018.

[21] J. Cho, M. K. Baskar, R. Li, M. Wiesner, S. H. Mallidi, N. Yalta, M. Karafiat, S. Watanabe, and T. Hori, "Multilingual sequence-to-sequence speech recognition: architecture, transfer learning, and language modeling," in *SLT*, 2018.

[22] Z. Tuske, D. Nolden, R. Schluter, and H. Ney, "Multilingual mrasta features for low-resource keyword search and speech recognition systems," in *ICASSP*, 2014, pp. 7854–7858.

[23] M. Karafiát, K. M. Baskar, P. Matějka, K. Veselý, F. Grézl, and J. Černocký, "Multilingual blstm and speaker-specific vector adaptation in 2016 but babel system," in *SLT*, 2016, pp. 637–643.

[24] M. Sperber, J. Niehues, G. Neubig, S. Stker, and A. Waibel, "Self-attentional acoustic models," in *InterSpeech*, 2018.

[25] C. Chiu and C. Raffel, "Monotonic chunkwise attention," *CoRR*, vol. abs/1712.05382, 2017. [Online]. Available: http://arxiv.org/abs/1712.05382

[26] 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.

[27] A. Zeyer, K. Irie, R. Schlter, and H. Ney, "Improved training of end-to-end attention models for speech recognition," in *Proc. Interspeech*, 2018, pp. 7–11.

[28] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in Neural Information Processing Systems*, vol. 2015-January. Neural information processing systems foundation, 2015, pp. 577–585.

[29] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The Kaldi speech recognition toolkit," in *ASRU*, 2011, pp. 1–4.

[30] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

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

[32] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique Yalta Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," in *Interspeech*, 2018, pp. 2207–2211.

[33] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, I. Szoke, and J. H. Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Interspeech*, 2014.

[34] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *ASRU*, 2011, pp. 42–47.

[35] A. Agarwal *et al.*, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR-TR-2014-112, August 2014. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=226641

[36] K. Veselý, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *SLT*, 2012, pp. 336–341.

[37] S. Kim and M. L. Seltzer, "Towards language-universal end-to-end speech recognition," in *ICASSP*, 2018.