INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

Transfer Learning Approaches for Streaming End-to-End Speech Recognition System

Vikas Joshi, Rui Zhao, Rupesh R. Mehta, Kshitiz Kumar, Jinyu Li

Microsoft Corporation

vikas.joshi, ruzhao, rupesh.mehta, kshitiz.kumar, jinyli@microsoft.com

Abstract

Transfer learning (TL) is widely used in conventional hybrid automatic speech recognition (ASR) system, to transfer the knowledge from source to target language. TL can be applied to end-to-end (E2E) ASR system such as recurrent neural network transducer (RNN-T) models, by initializing the encoder and/or prediction network of the target language with the pre-trained models from source language. In the hybrid ASR system, transfer learning is typically done by initializing the target language acoustic model (AM) with source language AM. Several transfer learning strategies exist in the case of the RNN-T framework, depending upon the choice of the initialization model for encoder and prediction networks. This paper presents a comparative study of four different TL methods for RNN-T framework. We show $10\% - 17\%$ relative word error rate reduction with different TL methods over randomly initialized RNN-T model. We also study the impact of TL with varying amount of training data ranging from 50 hours to 1000 hours and show the efficacy of TL for languages with a very small amount of training data.

Index Terms: speech recognition, transfer learning, end-to-end systems, low resource learning, adaptation

1. Introduction

Speech enabled applications are increasingly gaining popularity across the world. This has initiated a need to build accurate automatic speech recognition (ASR) system across different languages. Also, End-to-End (E2E) ASR systems are emerging as a popular alternative to conventional hybrid ASR systems. They replace the acoustic model (AM), language model (LM) and pronunciation model with a single neural network [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11]. Recurrent neural network transducer (RNN-T) [1] is one such E2E system that allow streaming input and is suitable for real-time ASR applications. Therefore there is a lot of interest in building accurate RNN-T models for different languages spoken across the world.

There is often disparity in the availability of transcribed data for different languages. In most cases, a lot more data is available for American English than other languages. The quality of ASR model depends on a number of factors including, the training data quantity and diversity, acoustic model structure, and optimization algorithm. Furthermore, training data diversity spans a number of factors in adults, kids, speaking rate, accents, near-field, and far-field acoustic conditions. A low-resource locale has limited ASR training data, and may not meet the acoustic diversity needed to train a robust model that can generalize to above acoustic factors. To overcome the low-resource constraint, transfer learning has been widely used in the hybrid ASR system to transfer the knowledge from a well trained source locale to a low-resource target locale that bring significant acoustic robustness for the target locale. In our recent work, we applied TL from a large scale en-US conventional hybrid model to the corresponding models in en-IN and it-IT locales, and achieved over 8% word error rate relative reduction (WERR). Motivated by the success of the TL methods in the hybrid ASR system, we explore TL methods to improve low-resource RNN-T models.

Besides improving the target model acoustic robustness, TL is also crucial for training large and complex deep learning architectures. RNN-T models are difficult to train [12] and also require significantly large amount of data to jointly train the acoustic as well as language model attributes. In our study we have noted weaker convergence or significant parameter tuning requirements for desirable E2E training outcome for low-resource locale. Therefore we expect TL techniques to be even more relevant for E2E systems to stabilize training and improve ASR accuracy.

In the hybrid ASR system, transfer learning is typically done by initializing the target AM with the source AM. In the RNN-T framework, several transfer learning strategies exist depending upon the choice of the initialization model for the encoder and prediction networks. In this paper, we compare different transfer learning strategies in the RNN-T framework. We propose two-stage TL, by first training a target initialization model bootstrapped with a pretrained source model. Subsequently, this model is used to initialize the target RNN-T model. The two-stage TL approach shows 17% WERR reduction and faster convergence in the training loss as compared to randomly initialized RNN-T model. We also study the effect of TL with different amount of training data and show the importance of transfer learning in the case of low-resource languages.

2. Relation to prior work

Several methods have been proposed to improve the performance of low-resource ASR models [13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. Successful strategies include transfer learning [17, 18], that leverage a well trained AM from high-resource language to bootstrap the low-resource AM; multi-task training [19, 20] and ensemble learning [21, 22] that aim to utilize multilingual data and share the model parameters. However, most of these methods are studied in the context of hybrid ASR system.

A few multi-lingual approaches are recently proposed in the E2E framework [23, 24, 25]. Authors in [23] propose the multilingual RNN-T model with language specific adapters and data-sampling to handle data imbalance. Audio-to-byte E2E system is proposed in [24] where bytes are used as target units instead of grapheme or word piece units, as bytes are suitable to scale to multiple languages. A transformer based multi-lingual E2E model, along with methods to incorporate language information is proposed in [25]. Although multi-lingual methods are attractive to address the problem of low-resource languages, the

http://dx.doi.org/10.21437/Interspeech.2020-2345

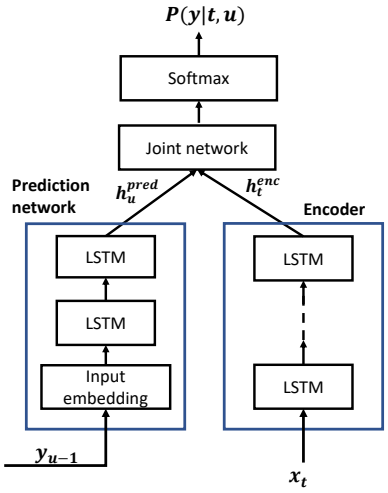

Figure 1: *The RNN-T model.*

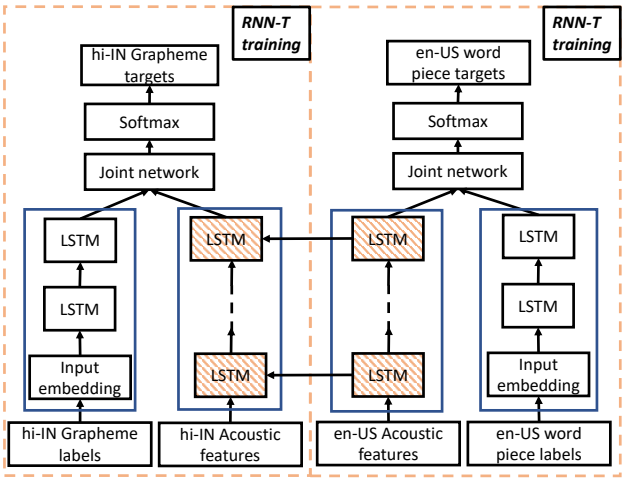

Figure 2: *en-US RNN-T initialization.*

transfer learning methods, besides being simple and effective, have the benefit of not needing the high-resource language data, but only the models trained on them. In many practical scenarios, trained models are available, however the original corpus is not. Given the simplicity and effectiveness of TL, we explore transfer learning approaches to improve the performance of low-resource RNN-T models.

The rest of this paper is organized as follows: In Section 3, we briefly discuss the RNN-T model. The transfer learning methods for RNN-T are described in Section 4 and experimental setup in Section 5. Next, we discuss results in Section 6, followed by conclusions in Section 7.

3. RNN Transducer model

The RNN-T model was proposed by Alex Graves [1]. The RNN-T model architecture has three components; an encoder, prediction network and joint network as shown in Fig. 1. The encoder maps the input acoustic feature, x_t, to a high level representation, ${h_t}^{enc}$, where t represents the time index. The prediction network receives the previously predicted non-blank symbol, y_{u-1} and maps it to a high level representation, h_u^{pred}. The joint network is a feed forward network that combines the encoder and prediction network outputs. The posterior probability over all the targets, $p(y|t,u)$ is obtained after softmax operation on the output of joint network. The whole network is trained jointly to minimize the RNN-T loss [1]. In our implementation, the encoder consists of 6 long short-term memory (LSTM) [26] layers and the prediction network has 2 LSTM layers along with the input embedding matrix. During inference, beam search decoding is used to find the most likely label sequence.

4. Transfer learning methods for RNN-T

The RNN-T models are difficult to train and are often initialized with the pretrained models. Initializing the encoder with connectionist temporal classification (CTC) model [2] or cross entropy (CE) model [12], and the prediction network with LSTM language model (LM) is proven to be beneficial [27]. Transfer learning can also be used to overcome the RNN-T training difficulty by initializing the low-resource (target) RNN-T models with the models trained on high-resource (source) languages.

A number of choices exist in selecting the initialization model for encoder and prediction network in the context of TL the RNN-T model. Authors in [12] have shown that CE initialized RNN-T models perform better than CTC initialized models, and hence, we only explore CE models for initialization. The following choices exist for encoder/prediction network initialization of the target RNN-T model: a) Source RNN-T encoder/prediction networks b) Pretrained networks used to initialize the source RNN-T model c) Pretrained models trained only on the target language. Therefore several combinations are possible depending upon the choice of the initialization model for encoder and prediction network.

In this paper, we explore TL methods in the context of Hindi as the target language and American English as the source language. The goal is to improve Hindi RNN-T model by leveraging models trained on American English, which has approximately ten times more data than Hindi. 'en-US' prefix is used to refer to models trained with American English and 'hi-IN' prefix is used to refer to models trained with Hindi data. We next discuss different transfer learning strategies in detail.

4.1. en-US RNN-T initialization

The hi-IN RNN-T encoder is initialized with en-US RNN-T encoder as shown in Fig.2. The hi-IN RNN-T model is trained with hi-IN acoustic data and grapheme targets. The en-US RNN-T model is trained with en-US acoustic data and the corresponding word piece targets. The encoder of the en-US RNN-T model is in turn initialized with a pretrained en-US CE model. Note that the prediction network of both hi-IN and en-US RNN-T models are randomly initialized. After initialization, all parameters of the RNN-T model are trained to minimize the RNN-T loss. In all the figures, layers initialized with pretrained networks are represented by cross lined blocks and randomly initialized layers are represented with plain blocks. The details of how we develop en-US RNN-T model can be found in [28].

4.2. en-US CE initialization

In en-US CE initialization, the hi-IN encoder is initialized with en-US CE model which was used to initialize the en-US RNN-T encoder, discussed in Section 4.1. The en-US CE model is trained on en-US acoustic data and the corresponding word piece targets. The frame level alignment with word piece tar-

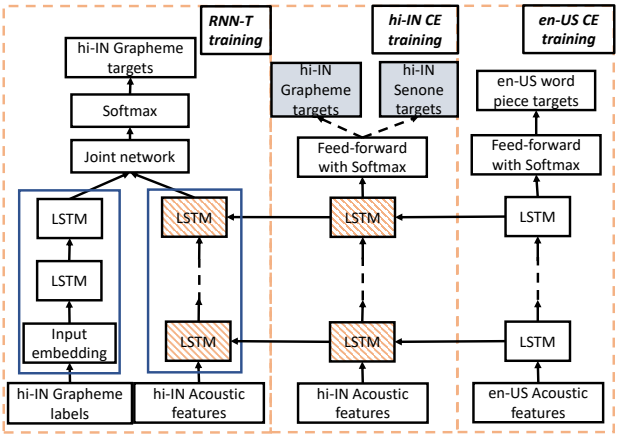

Figure 3: *Two stage transfer learning.*

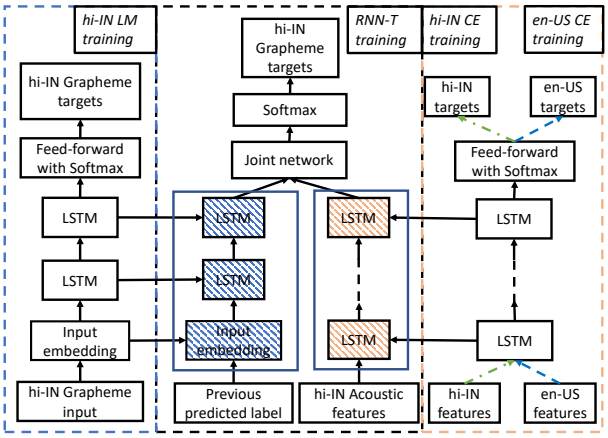

Figure 4: *Encoder and prediction network initialization. Two transfer schemes are shown in this figure: a) hi-IN CE + hi-IN LM initialization where the encoder is initialized with hi-IN CE model b) en-US CE + hi-IN LM initialization where the encoder is initialized with en-US CE model.*

gets (necessary for CE training), is obtained from word level alignments as discussed in [12]. From the word alignments, the start frame, end frame and total number of frames corresponding to each word is known. The words are then divided into corresponding word pieces, and equal number of frames are allocated to each word piece within the boundary of the frames corresponding to the word. In this scheme, the hi-IN RNN-T prediction network is randomly initialized.

4.3. Two-stage transfer learning

Transfer learning can be done in two stages as shown in Fig. 3. In the first stage, hi-IN CE model is trained starting from en-US CE model. Subsequently, the hi-IN RNN-T model is trained by initializing the encoder with hi-IN CE model. The hi-IN CE model can be trained either with senone and grapheme targets as depicted in Fig. 3. The senone based CE model can distinguish more finer acoustic classes as senones represent much finer acoustic information than graphemes. However, the grapheme based CE model is better aligned with the RNN-T model as they both are trained with grapheme targets. The prediction network is again randomly initialized.

4.4. Encoder and prediction network initialization

In the previously described transfer learning methods, only the encoder is initialized with a pretrained model. In this section, we will discuss initializing both encoder and prediction network with pretrained models as shown in Fig. 4.

The prediction network is initialized with a pretrained LSTM LM which is trained on an external text corpus as a language model using grapheme units, referred to as hi-IN LM. The sentence count (number of repetitions of the sentence or the query) differ significantly from one source to the other. In order to avoid biasing the LM towards the source with large sentence counts, we only select unique sentences for LM training. After selecting the unique sentences, the hi-IN LSTM LM is trained on approximately 200 million words.

The encoder is initialized with either hi-IN or en-US CE model as shown in Fig.4. For the purpose of compact representation, hi-IN and en-US CE model training is shown in a single block in Fig. 4. They do not share any parameters and are trained independently. The hi-IN CE model is trained with hi-IN acoustic data and grapheme targets, and en-US CE model is trained with en-US acoustic data and word piece targets. The configuration of initializing the prediction network with hi-IN LM and encoder with hi-IN CE model is referred to as hi-IN CE + hi-IN LM initialization. Similarly, the configuration of initializing the prediction network with hi-IN LM and the encoder with en-US CE model is referred to as en-US CE + hi-IN LM initialization.

We did not explore initializing the prediction network with en-US LSTM LM, as en-US and hi-IN lexical units differ (grapheme vs word piece), resulting in the input embedding matrices being significantly different, and thereby initializing the prediction network with en-US LSTM LM might not be beneficial. We also did not experiment initializing encoder with en-US RNN-T model in the context of encoder and prediction network initialization, as our experiments suggested that en-US CE initialization is better than en-US RNN-T initialization as discussed later in Section 6.

5. Experimental setup

The hi-IN models are trained with approximately 4 million utterances amounting to few thousand hours of speech data. The speech data is distorted by noise to achieve robustness to noisy conditions. The en-US models were trained with 65000 hours of data. The hi-IN test set contains 17619 utterances consisting of five different scenarios including phrasal, conversational and code-mixed utterances. Training and test utterances are anonymized to remove any personally identifiable information.

We use 80-dimensional log Mel filter bank features computed every 10 milliseconds (ms). Eight vectors are stacked together to form 640-dimensional acoustic features fed to the encoder. The frames are shifted by 30ms. The hyper-parameters such as number of layers, layer dimension, frame size was tuned for en-US model [12] and we adopted the same parameters for Hindi. All encoders have six LSTM layers with 1600 hidden dimension and 800 projection dimension. All prediction networks have two LSTM layers with same cell dimension as encoders. Such a model setup follows the en-US work in [28]. All models are evaluated after training for 6 sweeps of the training data.

Experiment	WER
Random initialization	26.53
en-US RNN-T initialization	22.97
en-US CE initialization	22.38
Two-stage initialization with senone targets	22.31
Two-stage initialization with grapheme targets	**21.89**
hi-IN CE + hi-IN LM initialization	24.29
en-US CE + hi-IN LM initialization	22.63
Hybrid model	22.32

Table 1: *WER [%] on hi-IN test sets for random initialization, different transfer learning methods and the hybrid model.*

The hi-IN grapheme targets contain all the unique graphemes in the Hindi native script. We also include grapheme targets with B_ prefix to be able to segment the grapheme sequence into word sequence. Some research works use <space> symbol, however, we observed better accuracy with B_ prefix based grapheme targets. A total of 130 grapheme targets are obtained by combining the the original Hindi graphemes, graphemes with B_ prefix and <blank> symbol. The word piece targets for en-US model is obtained by using byte pair encoding [29] algorithm as described in [12].

We also report the word error rate (WER) on hybrid ASR model trained with same amount of data. The AM consists of 6 layers of latency-controlled bidirectional LSTM [30] with 1024 hidden dimension and 512 projection dimension. AM is CE trained followed by EMBR training. The softmax layer has 9212 senone labels. 80-dimensional log Mel filter bank features are computed every 10ms. Frame skipping [31] is done by a factor of 2. Run-time decoding is performed using a 5-gram language model.

6. Discussion of results

Table. 1 shows WER for different transfer learning methods on hi-IN test sets. en-US CE initialization outperforms random initialization with 15.6% relative WER (WERR) reduction. en-US RNN-T initialization is better than random, while is slightly inferior to en-US CE initialization. This could be because the en-US RNN-T encoder representations are influenced by en-US prediction network representations, as they are trained jointly. However, en-US CE model is trained in isolation and could serve as a better initialization model for the encoder. Two-stage transfer learning with grapheme targets performs better than the rest of the methods with 17.4% WERR reduction over random initialization. The pretraining method, hi-IN CE + hi-IN LM initialization improves over random initialization showing the importance of pretraining. The en-US CE + hi-IN LM initialization is better than pretraining, however, is inferior compared to en-US CE initialization, contrary to our expectation. The reason for such a behaviour is not known and we look to investigate this further in the future. With the improvements obtained from transfer learning, the WER of hi-IN RNN-T model (with transfer learning) is in parity with the hybrid model.

6.1. Efficacy of TL for different amount of training data

To study the efficacy of the TL with different amount of training data, we sample the original hi-IN data into smaller datasets consisting of 50 hours, 500 hours and 1000 hours. The RNN-T models are trained with random initialization and en-US CE initialization. The corresponding WERs on hi-IN testset are shown in Table. 2. en-US CE initialization shows 42.7% WERR reduction for 50 hour training over random initialization. The large WER gains with smaller training sets could be due to better RNN-T training convergence with TL as discussed in the next section. The above results also show a need for larger training data in RNN-T models.

	50 hours	500 hours	1000 hours
Random initialization	83.77	69.32	51.56
en-US CE initialization	47.96	35.07	32.75

Table 2: *WER [%] comparison on hi-IN test set between random and en-US CE initialized RNN-T models trained with 50 hours, 500 hours and 1000 hours.*

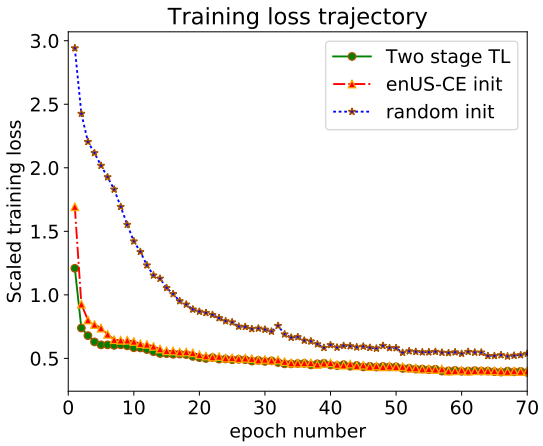

Figure 5: *Training loss with increasing epoch number for random initialization, en-US CE initialization and Two stage TL with grapheme targets.*

6.2. RNN-T training convergence

The training loss with increasing epochs for random initialization, en-US CE initialization and Two-stage TL with grapheme targets is shown in Fig. 5. Each epoch is trained with 60 hours of non-overlapping speech data. The training parameters such as learning rate, mini-batch size are identical for all three methods shown in Fig. 5. The training loss converges much faster with the transfer learning methods than random initialized models. The Two-stage TL converges faster than en-US CE initialization. It is also interesting to note that the first epoch training loss is much lower for Two-stage TL than others, thereby indicating the superiority of initialization models in Two-stage TL.

7. Conclusion

In this paper, we explore transfer learning methods for RNN-T models. Our motivation is to leverage well-trained en-US models to bootstrap hi-IN RNN-T models and also to stabilize the hi-IN RNN-T model training. We evaluated the following transfer learning methods: a) en-US CE initialization b) en-US RNN-T initialization c) Two-stage transfer learning and d) Encoder and prediction network initialization. Based on the WER gains and training convergence, we propose Two-stage learning approach with grapheme targets as the preferred transfer learning strategy. The experiments on smaller data-sets and training loss convergence reveal the importance of transfer learning for low-resource RNN-T models. The methods discussed in this paper can be generalized to other low-resource languages as well. In future, we plan to explore other transfer learning methods and its extension to multi-lingual RNN-T models.

8. References

[1] A. Graves, "Sequence transduction with recurrent neural networks," arXiv:1211.3711, 2012.

[2] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6645–6649.

[3] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 4960–4964.

[4] R. Prabhavalkar, K. Rao, T. Sainath, B. Li, L. Johnson, and N. Jaitly, "A comparison of sequence-to-sequence models for speech recognition," in *Proceedings of Interspeech*, 2017.

[5] C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani, "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4774–4778.

[6] E. Battenberg, J. Chen, R. Child, A. Coates, Y. G. Y. Li, H. Liu, S. Satheesh, A. Sriram, and Z. Zhu, "Exploring neural transducers for end-to-end speech recognition," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 206–213.

[7] T. Sainath, C.-C. Chiu, R. Prabhavalkar, A. Kannan, Y. Wu, P. Nguyen, and Z. Chen, "Improving the performance of online neural transducer models," in *Proc. ICASSP*, 2018. [Online]. Available: https://arxiv.org/pdf/1712.01807.pdf

[8] Y. He, T. N. Sainath, R. Prabhavalkar, I. McGraw, R. Alvarez, D. Zhao, D. Rybach, A. Kannan, Y. Wu, R. Pang, Q. Liang, D. Bhatia, Y. Shangguan, B. Li, G. Pundak, K. C. Sim, T. Bagby, S. Chang, K. Rao, and A. Gruenstein, "Streaming end-to-end speech recognition for mobile devices," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6381–6385.

[9] J. Li, R. Zhao, H. Hu, and Y. Gong, "Improving RNN transducer modeling for end-to-end speech recognition," in *Proc. ASRU*, 2019.

[10] T. N. Sainath, Y. He, B. Li, A. Narayanan, R. Pang, A. Bruguier, S. yiin Chang, W. Li, R. Alvarez, Z. Chen, C.-C. Chiu, D. Garcia, A. Gruenstein, K. Hu, M. Jin, A. Kannan, Q. Liang, I. McGraw, C. Peyser, R. Prabhavalkar, G. Pundak, D. Rybach, Y. Shangguan, Y. Sheth, T. Strohman, M. Visontai, Y. Wu, Y. Zhang, and D. Zhao, "A streaming on-device end-to-end model surpassing server-side conventional model quality and latency," 2020.

[11] J. Li, Y. Wu, Y. Gaur, C. Wang, R. Zhao, and S. Liu, "On the comparison of popular end-to-end models for large scale speech recognition," in *Proc. Interspeech*, 2020.

[12] H. Hu, R. Zhao, J. Li, L. Lu, and Y. Gong, "Exploring pre-training with alignments for rnn transducer based end-to-end speech recognition," in *ICASSP*, April 2020. [Online]. Available: https://www.microsoft.com/en-us/research/publication/exploring-pre-training-with-alignments-for-rnn-transducer-based-end-to-end-speech-recognition/

[13] S. Thomas, S. Ganapathy, and H. Hermansky, "Multilingual mlp features for low-resource LVCSR systems," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 4269–4272.

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

[15] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 8619–8623.

[16] H. Arsikere, A. Sapru, and S. Garimella, "Multi-Dialect Acoustic Modeling Using Phone Mapping and Online i-Vectors," in *Proc. Interspeech 2019*, 2019, pp. 2125–2129. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-2881

[17] J. Kunze, L. Kirsch, I. Kurenkov, A. Krug, J. Johannsmeier, and S. Stober, "Transfer learning for speech recognition on a budget," in *Proceedings of the 2nd Workshop on Representation Learning for NLP*. Vancouver, Canada: Association for Computational Linguistics, Aug. 2017, pp. 168–177. [Online]. Available: https://www.aclweb.org/anthology/W17-2620

[18] Y. Huang, D. Yu, C. Liu, and Y. Gong, "Multi-accent deep neural network acoustic model with accent-specific top layer using the KLD-regularized model adaptation," in *Interspeech 2014*, September 2014.

[19] D. Yu and L. Deng, "Efficient and effective algorithms for training single-hidden-layer neural networks," *Pattern Recognition Letters*, January 2012.

[20] J. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 7304–7308.

[21] L. Deng and J. C. Platt, "Ensemble deep learning for speech recognition," in *INTERSPEECH*, 2014.

[22] A. Waters, M. Bastani, M. G. Elfeky, P. Moreno, and X. Velez, "Towards acoustic model unification across dialects," in *2016 IEEE Workshop on Spoken Language Technology*, 2016.

[23] A. Kannan, A. Datta, T. Sainath, E. Weinstein, B. Ramabhadran, Y. Wu, A. Bapna, and Z. Chen, "Large-scale multilingual speech recognition with a streaming end-to-end model," 2019.

[24] B. Li, Y. Zhang, T. Sainath, Y. Wu, and W. Chan, "Bytes are all you need: End-to-end multilingual speech recognition and synthesis with bytes," in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5621–5625.

[25] V. M. Shetty, M. Sagaya Mary N J, and S. Umesh, "Improving the performance of transformer based low resource speech recognition for Indian languages," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 8279–8283.

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

[27] K. Rao, H. Sak, and R. Prabhavalkar, "Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 193–199.

[28] J. Li, , R. Zhao, Z. Meng *et al.*, "Developing RNN-T models surpassing high-performance hybrid models with customization capability," in *Proc. Interspeech*, 2020.

[29] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1715–1725. [Online]. Available: https://www.aclweb.org/anthology/P16-1162

[30] Y. Zhang, G. Chen, D. Yu, K. Yaco, S. Khudanpur, and J. Glass, "Highway long short-term memory RNNS for distant speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5755–5759.

[31] Y. Miao, J. Li, Y. Wang, S. Zhang, and Y. Gong, "Simplifying long short-term memory acoustic models for fast training and decoding," in *Proc. ICASSP*, 2016.