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

Listen, Attend, Spell and Adapt: Speaker Adapted Sequence-to-Sequence ASR

Felix Weninger[1], *Jesús Andrés-Ferrer*[2], *Xinwei Li*[1], *Puming Zhan*[1]

[1]Nuance Communications, Inc., Burlington, MA, USA
[2]Nuance Communications, Inc., Valencia, Spain

{felix.weninger,jesusandres.ferrer,xinwei.li,puming.zhan}@nuance.com

Abstract

Sequence-to-sequence (seq2seq) based ASR systems have shown state-of-the-art performances while having clear advantages in terms of simplicity. However, comparisons are mostly done on speaker independent (SI) ASR systems, though speaker adapted conventional systems are commonly used in practice for improving robustness to speaker and environment variations. In this paper, we apply speaker adaptation to seq2seq models with the goal of matching the performance of conventional ASR adaptation. Specifically, we investigate Kullback-Leibler divergence (KLD) as well as Linear Hidden Network (LHN) based adaptation for seq2seq ASR, using different amounts (up to 20 hours) of adaptation data per speaker. Our SI models are trained on large amounts of dictation data and achieve state-of-the-art results. We obtained 25% relative word error rate (WER) improvement with KLD adaptation of the seq2seq model vs. 18.7% gain from acoustic model adaptation in the conventional system. We also show that the WER of the seq2seq model decreases log-linearly with the amount of adaptation data. Finally, we analyze adaptation based on the minimum WER criterion and adapting the language model (LM) for score fusion with the speaker adapted seq2seq model, which result in further improvements of the seq2seq system performance.

Index Terms: automatic speech recognition, speaker adaptation, sequence-to-sequence, end-to-end

1. Introduction

End-to-end ASR systems have recently received increasing attention, due to their ability of integrating all components of an ASR system in a single deep neural network (DNN), which greatly simplifies and unifies the training and decoding process [1, 2, 3]. Sequence-to-sequence (seq2seq) modeling [4, 5] is a state-of-the-art method for end-to-end ASR, which has shown competitive results compared to traditional ASR systems [6].

However, this kind of modeling technique does not have inherent advantages in terms of robustness to speaker and/or environment changes. Thus, in this paper, we propose to apply speaker adaptation for seq2seq end-to-end ASR. Adaptation is commonly applied in practice for conventional ASR, yet it can be challenging for end-to-end systems mainly due to two reasons: First, because end-to-end ASR subsumes all components in a single DNN, this network usually contains a very large number of free parameters (100 m and more), which makes it difficult to improve with a small amount of data (say, a few minutes of speech). Second, since there is no explicit distinction between acoustic model (AM) and language model (LM) in most end-to-end systems[1], adaptation can only be done jointly on AM and LM. This compounds the first problem, as it is generally accepted that effective LM training needs large amounts of text data – more than might be available in the transcripts of acoustic data used for AM adaptation.

In this regard, our paper makes the following contributions: To address the first issue, we employ Kullback-Leibler-Divergence (KLD) regularized and Linear Hidden Network (LHN) based adaptation to cope with scarce data. Moreover, regarding the second issue, we analyze the effects of LM speaker adaptation in combination with shallow fusion [8, 9]. We show that this adapted LM fusion can effectively exploit additional text data from a speaker in addition to the acoustic adaptation data. Finally, we demonstrate competitive accuracy compared to a conventional ASR system that exploits both AM and LM adaptation.

2. Relation to prior work

Many approaches for adapting 'conventional' DNN acoustic models have been developed over the years, such as linear transformation based approaches in [10, 11, 12], training with regularization in [13, 14], and various forms of speaker identity vectors in [15, 16, 17]. However, relatively few studies so far deal with adaptation of the end-to-end ASR systems. [18] proposed to use speaker identity vectors in a hybrid model combining Connectionist Temporal Classification (CTC) with seq2seq modeling. Furthermore, [19] investigated the fine-tuning of this model architecture per speaker. In [20], LHN and KLD were investigated for a CTC model.

Regarding LM adaptation, recently there have been several proposals for adapting recurrent neural network (RNN) LMs to domain specific data, including fast marginal adaptation (FMA) [21], as well as noise contrastive estimation (NCE) [22] combined with KLD regularization [23]. However, these works did not address the interaction with end-to-end ASR, which has an inherent LM adaptation capability. Note that in [24], a technique similar to LM biasing was proposed for seq2seq model. However, in contrast to our techniques, this does not allow for exploiting large amounts of adaptation data per user.

The novelty of our paper lies in the following aspects: First, we are not aware of any study applying KLD and LHN adaptation to the seq2seq models with attention. Furthermore, to our knowledge, there is no previous work comparing the adaptation of end-to-end systems to the offline AM and LM adaptation often used in traditional ASR and demonstrating competitive performance in the speaker adapted scenario. Finally, our paper also sheds light on various kinds of LHN adaptation, and introduces the minimum word error rate (mWER) criterion in the context of adaptation as well as adapted LM fusion using KLD + NCE.

3. Methodology

In this work, we use an encoder-decoder architecture with attention similar to Listen-Attend-Spell (LAS) [4], treating end-to-end ASR as a seq2seq learning task: The goal is to predict a sequence y_i of symbols (here, we use sub-word units) from

[1] There are models such as RNNT [7] that explicitly divide the model into two components which can be understood as equivalent to the LM and AM.

http://dx.doi.org/10.21437/Interspeech.2019-2719

a sequence of acoustic features x_t, $t = 1, \dots, T$, where T is the number of frames in the utterance. The encoder e creates a hidden representation h_j, $j = 1, \dots, T'$ of the acoustic features:

$$h_1, \dots, h_{T'} = e(x_1, \dots, x_T). \tag{1}$$

In our work, e is implemented as a stack of convolutional (CNN) layers followed by bidirectional Long Short-Term Memory (bLSTM) layers.

The decoder of the seq2seq model is similar to an RNN LM that takes into account a context vector c_i built from the encoder representation for predicting the i-th output symbol. An attention mechanism is used to focus c_i on various parts of the encoder output sequence. Here, we use the mechanism proposed by Bahdanau *et al.* [25]. The output distribution $p_i = p(y_i|y_1, \dots, y_{i-1}, x)$ for the i-th symbol is computed by

$$\begin{aligned} s_i &= f(s_{i-1}, \text{Embedding}(y_{i-1}), c_{i-1}), && (2)\\ c_i &= \text{Attention}(s_i, h_1, \dots, h_{T'}), && (3)\\ s'_i &= \text{Dense}(s_i, c_i), && (4)\\ p_i &= g(s'_i), && (5) \end{aligned}$$

where g is a softmax layer and f is a RNN. In our work, f is a stack of LSTM layers. To perform ASR using the seq2seq model, the posterior $p(y|x)$ is factorized as

$$p(y|x) = \prod_i p(y_i|y_1, \dots, y_{i-1}, x). \tag{6}$$

In practice, beam search is used to produce the sequence $y_1, \dots, y_i, \dots$ of output symbols one at a time.

3.1. KLD adaptation

We train speaker adapted (SA) seq2seq models on an adaptation set with transcript y^*, initializing with the parameters of the SI model (cf. Figure 1). The idea of KLD-regularized adaptation [14] is to encourage the output distributions of the SA and SI models to be similar, by minimizing the loss:

$$\mathcal{L}^{\text{KLD}} = \sum_i (1 - \beta)\mathcal{L}^{\text{CE}}(y^*_i, p_i) + \beta\mathcal{L}^{\text{CE}}(p^{\text{SI}}_i, p_i). \tag{7}$$

Here, β is the regularization strength (*KLD relevance*), $\mathcal{L}^{\text{CE}}$ is the cross-entropy (CE) loss, and p^{SI}_i is the output distribution of the SI model. The target y^*_i is represented as a one-hot vector.

3.2. LHN adaptation

The idea of LHN adaptation [11, 12, 26] is to insert a speaker-specific linear feed-forward layer, which has a square weight matrix U. By initializing U with identity, the starting point for adaptation is the SI model. During adaptation, only U and the bias b are updated, which reduces the number of speaker-specific parameters compared to KLD adaptation. In this work, we investigate the application of LHN to various layer outputs in the seq2seq model: the input features x (cf. [11]), the encoder output h_j, and the decoder output s'_i. We found it crucial to perform LHN with the KLD regularized loss (7), as shown in Figure 1, instead of the simple CE loss.

3.3. mWER training and adaptation

mWER training [27] was introduced as a discriminative training method for seq2seq systems and is similar in spirit to traditional sequence training [28]. In our experiments, we always build SA models from mWER trained SI models. Furthermore, in order to exploit mWER training in adaptation, we propose to minimize the KLD loss (7) augmented by the mWER loss:

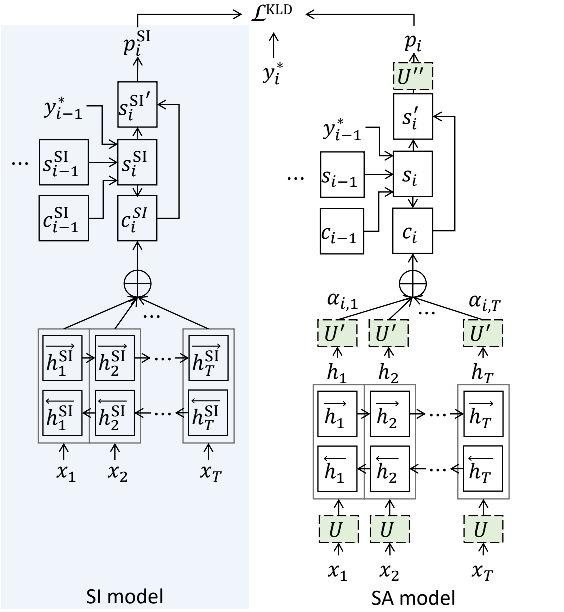

Figure 1: *Block diagram of the proposed LAS-like model adaptation via LHN and KLD. For simplicity, a single layer bLSTM encoder is shown instead of the CNN + pyramid bLSTM used in the experiments. Dashed boxes indicate possible LHN transformations. $\alpha_{i,\cdot}$: attention weights. SI/SA: speaker independent / speaker adapted.*

$$\mathcal{L}^{\text{mWER,KLD}} = \gamma_1\mathcal{L}^{\text{KLD}} + \gamma_2\mathcal{L}^{\text{mWER}}. \tag{8}$$

An alternative way of viewing this is that the mWER loss is regularized with both a CE term (cf. [27]; in our case, with weight $\gamma_1(1 - \beta)$) and a KLD term (with weight $\gamma_1\beta$).

3.4. LM fusion and decoding

In order to assess the effect of an external LM in the adaptation of the seq2seq models, we use a variation of shallow fusion [9], which has reported better results than other methods [29]. To avoid the bias of the seq2seq model to both deletions and insertions, we add a coverage score [8] per each decoded token for which the seq2seq models attended more than 0.5 to different encoder state. Specifically, we used beam search for:

$$\arg\max_y\big\{\log p(y\,|\,x) + \lambda_{\text{LM}}\log p(y) + \lambda_{\text{cov}}\,\text{cov}(x, y)\big\} \tag{9}$$

where $p(y|x)$ is the seq2seq probability, $p(y)$ is the LM probability, $\text{cov}(x, y)$ is the coverage score in Eq. (11) of [8], and λ_{LM} and λ_{cov} are the weights for the LM and the coverage score.

4. Experiments

4.1. Data set

We perform our experiments on a dictation data set. All utterances are anonymized field data. The audio is sampled at 8 kHz. A training set of 7.6 k hours from 58 k speakers is used to train SI models. The performance is measured on an evaluation set with 35 speakers (392 k words). The speakers cover various dictation domains with different vocabularies.

For each speaker in the evaluation set, we have up to 20 hours of acoustic adaptation data available. We also perform experiments where the adaptation data amount is restricted to 5 minutes, 30 minutes, 2 hours, or 6 hours. Besides, for LM adaptation, additional text documents (0.37 m on average per speaker) related to the dictations are available.

The transcripts of the training set as well as the adaptation sets are pseudo truth, i.e., the output of an ASR system along with some user corrections. Thus, the adaptation schema is semi-supervised. In contrast, the evaluation set is manually transcribed.

4.2. End-to-end models

Our ASR systems use 40-dimensional log Mel filterbank outputs as input features. In the seq2seq model, the inputs are first passed through a stack of 3 CNN layers, which is parameterized so as to yield a 768-dimensional embedding for each input frame. Subsequently, there is a pyramid bLSTM encoder with 6 layers (768 LSTM units per layer and direction), which performs frame decimation (by a factor of 2) after every other layer, thus reducing the frame rate by a factor of 8. The decoder uses 2 (unidirectional) LSTM layers (1 536 units per layer). The softmax output layer predicts the posterior probabilities for 20 k word piece targets determined on the training set. In total, the model has 181 m parameters.

The SI end-to-end model is trained using the CE criterion, followed by mWER training [27]. We employ dropout [30] (with probability 0.1), label smoothing [31] (with probability 0.9 for the target class), and early stopping (using a validation set held out from the training data) in order to improve generalization of the model.

For the adaptation experiments, the KLD relevance is set to $\beta = 0.6$ and a small fixed number of epochs is run. In adaptation, we decrease the learning rate to perform fine-tuning, increase the dropout probability to 0.25 to cope with the limited amount of data, and we decrease the batch size, so that the model is updated more frequently. The adaptation hyperparameters were tuned in preliminary experiments separately for the end-to-end and the baseline models.

For LM fusion, we trained a LSTM LM with embeddings of 512 and 2 LSTM layers of 1 k units each, followed by a linear projection to 512. The outputs of the LM correspond to the word pieces of the seq2seq model. The LM is self-normalized and trained with the NCE [22] loss. The LSTM LM is trained by sampling a large training set comprising 33.4 billion running words. We regularized the model applying dropout to the feed-forward connections [32] and some small (10^{-6}) weight decay. For adapting the LM per speaker, we use the transcripts of the 20 h acoustic data and optionally the speakers' text documents to fine-tune the LM with a small learning rate, and optionally also regularize the NCE loss by adding a KLD term [23].

4.3. Baseline models

We compare the performance of the end-to-end system against a 'conventional' hybrid ASR system, which uses a linearly augmented (LA-) DNN AM [33] with 22 layers and 25 m parameters. The acoustic features are the same as in the end-to-end system. The SI LA-DNN AM is initially trained using the CE loss, followed by Hessian-free sequence discriminative training [28]. Then, we train SA LA-DNN AMs using the KLD-regularized CE loss in analogy to (7), with $\beta = 0.2$.

We use a specific LM for each of the dictation domains in the evaluation set. Each LM comprises n-gram as well as RNN components and is trained on several hundred millions of words, collectively summing up to billions of words. The domain specific LMs are then interpolated with speaker specific LMs, which are trained on the same data that is used to fine-tune the LSTM LM used in seq2seq LM fusion.

Table 1: *WER [%] and WERR [%] compared to the SI model with the KLD method, adapting all parameters of the seq2seq model or subsets thereof with 2 h of adaptation data.*

Parameters	# Parameters	WER	WERR
Encoder (e)	83.0 m	9.60	7.7
Decoder (a, f, g)	98.0 m	9.12	12.3
All	181 m	**8.91**	**14.3**

Table 2: *WER and WERR compared to the SI model with the LHN method for various positions of the linear layer in the seq2seq model, adapting with 2 h of data.*

LHN applied to	LHN dim.	WER	WERR
Features (x)	40×40	10.31	0.9
Encoder (h)	1536×1536	9.56	8.1
Decoder (s')	1536×1536	**9.23**	**11.3**

5. Results

5.1. Adaptation of various parameter subsets

We start our evaluation by adapting different parameter subsets of the seq2seq model with the KLD method, using 2 h of adaptation data. We conjecture that due to the importance of language modeling for the dictation task, adapting only the *decoder* should yield a significant gain as well. As can be seen from Table 1, adapting only the decoder increases the WER by only about 2 % relative compared to adapting the entire model. At the same time, it is much less expensive, since the error signal does not have to be backpropagated through the deep bLSTM encoder stack. Conversely, adapting only the encoder as in [19] is not competitive on our data set.

5.2. Position of the LHN layer

Next, Table 2 shows the performance of the LHN method, varying the position of the inserted linear layer. We found that inserting the LHN transformation after the acoustic features has limited potential. Using LHN as encoder output transformation performs slightly better than KLD adaptation of the encoder parameters. Moreover, using LHN as decoder output transformation delivers the most promising results, yielding 11.3 % relative WER reduction (WERR). It is similar in performance to decoder KLD adaptation (see Table 1), although only a small fraction of the parameters are adapted (2.4 m vs. 98 m parameters).

5.3. Influence of the amount of adaptation data

We now compare the results with KLD and LHN for various amounts of adaptation data. As can be seen from Figure 2, the WER decreases log-linearly in the amount of adaptation data with KLD adaptation, yielding 7.77 % WER (25.3 % WERR) with 20 h of data. Regarding LHN, one could conjecture that it could be more effective than KLD for little adaptation data, as it adapts less parameters. For 5 minutes data, we found encoder LHN to be more stable than decoder LHN. Contrary to our expectations, LHN was outperformed by KLD on all the data

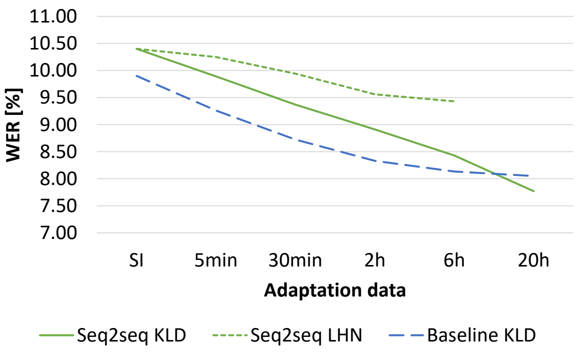

Figure 2: *WER of the speaker adapted seq2seq system (KLD or encoder LHN adaptation) and the speaker adapted conventional ASR system (KLD) with various amounts of adaptation data.*

Table 3: *WER [%] and WERR [%] compared to the SI model using mWER + KLD* (8) *by amount of data and regularization strength (β).*

# Data	β	WER	WERR
2 h	0.6	8.10	22.1
2 h	0.8	8.23	20.9
20 h	0.8	**7.33**	**29.5**

points shown in Figure 2, showing the effectiveness of the KLD regularization.

We also compare the performance to KLD regularized AM adaptation of the conventional system with various amounts of data. We can observe that the performance starts saturating when adapting with more than 2 h of data; as a result, the SA seq2seq system outperforms the SA conventional ASR system with 20 h of adaptation data. We attribute this to implicit LM adaptation in the seq2seq system. Thus, we will also compare the seq2seq and conventional systems under LM adaptation in Section 5.5.

5.4. Minimum WER adaptation

Further, Table 3 shows the results obtained by using the KLD + mWER criterion (8) in adaptation (using $\gamma_1 = \gamma_2 = 1$), starting from KLD SA models. We found some inconsistencies in the formatting of the adaptation vs. the evaluation data, to which the mWER adaptation was particularly sensitive. Using the cleaned data, the standard KLD criterion achieves 8.70 % WER with 2 h data, while we obtain 8.10 % WER from the mWER + KLD criterion. With stronger regularization ($\beta = 0.8$), the WER is slightly higher, but the improvement is more uniform (no degradations for any speaker). Increasing the adaptation data to 20 h, we obtain 7.33 % WER, which is a 3.2 % relative improvement compared to KLD (7.57 % WER).

5.5. Fusion with adapted LM

To verify the gains from LM fusion and adaptation in combination with seq2seq model adaptation, which already adapts the decoder part, we combined a seq2seq model adapted on 2 h acoustic data (using decoder LHN, cf. Table 2) with the generic LSTM LM and compared to the adapted LSTM LM. The results are shown in Table 4. We were able to obtain 3.9 % WERR from LM fusion and 2.4 % WERR from LM adaptation on top of LM fusion. The latter shows that LM adaptation can help incorporating additional user-specific data into the seq2seq model.

Table 4: *WER [%] and WERR [%] compared to seq2seq model adaptation only (decoder LHN, 2 h data) by LM fusion and LM adaptation (20 h adaptation data transcripts and user text).*

Shallow fusion LM	WER	WERR
None	9.23	–
LSTM 33 b words	8.87	3.9
+ LM adaptation	8.66	6.2
+ KLD	**8.48**	**8.1**

Table 5: *Comparison of conventional and seq2seq ASR with and without speaker adaptation, using 20 h of acoustic data for seq2seq / AM adaptation (via mWER + KLD) and 20 h transcripts + user documents for LM adaptation.*

(a) *Baseline system*

Adaptation		WER
AM	LM	
–	–	9.90
✓	–	8.04
–	✓	7.97
✓	✓	**7.18**

(b) *seq2seq system*

LM	Adaptation		WER
fusion	seq2seq	LM	
–	–	–	10.40
–	✓	–	7.33
✓	–	–	9.75
✓	✓	–	7.03
✓	–	✓	9.15
✓	✓	✓	**6.95**

Moreover, there is an additional 2.1 % WERR from using the NCE + KLD LM adaptation instead of simple fine-tuning.

Table 5 shows the summary of results achieved by conventional and seq2seq ASR with and without speaker adaptation, now using 20 h of data. While the seq2seq model without external LM is behind the conventional system, fusion with the generic LSTM LM makes it competitive with the conventional baseline (9.75 vs. 9.90 % WER). Applying LM fusion to the KLD + mWER adapted seq2seq system, we obtain 7.03 % WER (4.1 % WERR), which is competitive with the conventional ASR using AM and LM adaptation. While adaptation of the LM used for LM fusion improves significantly upon the SI seq2seq system (9.15 vs. 9.75 % WER), and LM adaptation yields a considerable improvement upon the conventional system with adapted AM (7.18 vs. 8.04 % WER), there is only a small gain for the SA seq2seq system (6.95 vs. 7.03 % WER). However, this gain is remarkable because the decoder-internal LM adaptation on 20 h transcripts is already done in the seq2seq adaptation.

6. Conclusions

In this paper, we have presented several effective techniques for adapting seq2seq ASR systems. Significant gains have been achieved even with a few minutes of speech data, and at the same time we have shown that seq2seq ASR systems can exploit larger amounts of adaptation data effectively. Furthermore, we have achieved state-of-the-art performances in the SA scenario, comparing to conventional ASR. Nevertheless, we also found that it is hard to get complementary gains from LM adaptation on top of seq2seq adaptation. To improve on this, we will look into interpolation of generic and and speaker adapted LMs. Moreover, in future work, we plan to explore LHN and i-vector on-line adaptation similar to [26].

7. Acknowledgements

We would like to thank Peter Skala and Ming Yang for their help with the baseline ASR adaptation and many helpful discussions.

8. References

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

[2] 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 *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. Okinawa, Japan: IEEE, 2017, pp. 206–213.

[3] J. Li, G. Ye, A. Das, R. Zhao, and Y. Gong, "Advancing acoustic-to-word CTC model," in *Proc. of ICASSP*. Calgary, Canada: IEEE, 2018, pp. 5794–5798.

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

[5] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. of 34th International Conference on Machine Learning (ICML)*. Sydney, Australia: PMLR, 2017, pp. 1243–1252.

[6] 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 *Proc. of ICASSP*. Calgary, Canada: IEEE, 2018, pp. 4774–4778.

[7] 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*, May 2013, pp. 6645–6649.

[8] J. Chorowski and N. Jaitly, "Towards better decoding and language model integration in sequence to sequence models," *arXiv:1612.02695*, 2017.

[9] A. Kannan, Y. Wu, P. Nguyen, T. N. Sainath, Z. Chen, and R. Prabhavalkar, "An analysis of incorporating an external language model into a sequence-to-sequence model," in *Proc. of ICASSP*. IEEE, 2018, pp. 5824–5828.

[10] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, "Speaker adaptation for hybrid HMM-ANN continuous speech recognition system," in *European Conference on Speech Communication and Technology*, Madrid, Spain, 1995, pp. 2171–2174.

[11] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, "Linear hidden transformations for adaptation of hybrid ANN/HMM models," *Speech Communication*, vol. 49, no. 10-11, pp. 827–835, 2007.

[12] K. Kumar, C. Liu, K. Yao, and Y. Gong, "Intermediate-layer DNN adaptation for offline and session-based iterative speaker adaptation," in *Proc. of INTERSPEECH*. Dresden, Germany: ISCA, 2015.

[13] X. Li and J. Bilmes, "Regularized adaptation of discriminative classifiers," in *Proc. of ICASSP*, vol. 1. Toulouse, France: IEEE, 2006, pp. 1–237.

[14] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. of ICASSP*. Vancouver, Canada: IEEE, 2013, pp. 7893–7897.

[15] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. Olomouc, Czech Republic: IEEE, 2013, pp. 55–59.

[16] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *Proc. of ICASSP*. Calgary, Canada: IEEE, 2018, pp. 5329–5333.

[17] K. Veselý, S. Watanabe, K. Žmolíková, M. Karafiát, L. Burget, and J. H. Černocký, "Sequence summarizing neural network for speaker adaptation," in *Proc. of ICASSP*. Shanghai, China: IEEE, 2016, pp. 5315–5319.

[18] M. Delcroix, S. Watanabe, A. Ogawa, S. Karita, and T. Nakatani, "Auxiliary feature based adaptation of end-to-end ASR systems," in *Proc. of INTERSPEECH*. Hyderabad, India: ISCA, 2018, pp. 2444–2448.

[19] T. Ochiai, S. Watanabe, S. Katagiri, T. Hori, and J. Hershey, "Speaker adaptation for multichannel end-to-end speech recognition," in *Proc. of ICASSP*. Calgary, Canada: IEEE, 2018, pp. 6707–6711.

[20] K. Li, J. Li, Y. Zhao, K. Kumar, and Y. Gong, "Speaker adaptation for end-to-end CTC models," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*. Athens, Greece: IEEE, 2018, pp. 542–549.

[21] K. Li, H. Xu, Y. Wang, D. Povey, and S. Khudanpur, "Recurrent neural network language model adaptation for conversational speech recognition," in *Proc. of INTERSPEECH*, Hyderabad, India, 2018, pp. 3373–3377.

[22] A. Mnih and Y. W. Teh, "A fast and simple algorithm for training neural probabilistic language models," in *In Proceedings of the International Conference on Machine Learning*, 2012.

[23] J. Andrés-Ferrer, N. Bodenstab, and P. Vozila, "Efficient language model adaptation with noise contrastive estimation and kullback-leibler regularization," in *Proc. of INTERSPEECH*. Hyderabad, India: ISCA, 2018, pp. 3368–3372.

[24] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *SLT*. IEEE, 2018, pp. 418–425.

[25] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. of International Conference on Learning Representations (ICLR)*. San Diego, CA: open publishing, 2015.

[26] X. Li, Y. Pan, M. Gibson, and P. Zhan, "DNN online adaptation for automatic speech recognition," in *Proc. of 29th Conference on Electronic Speech Signal Processing (ESSV)*. Ulm, Germany: TUDpress, 2018.

[27] R. Prabhavalkar, T. N. Sainath, Y. Wu, P. Nguyen, Z. Chen, C.-C. Chiu, and A. Kannan, "Minimum word error rate training for attention-based sequence-to-sequence models," in *Proc. of ICASSP*. Calgary, Canada: IEEE, 2018, pp. 4839–4843.

[28] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable Minimum Bayes Risk training of Deep Neural Network acoustic models using distributed Hessian-free optimization," in *Proc. of INTERSPEECH*. Portland, OR: ISCA, 2012.

[29] S. Toshniwal, A. Kannan, C.-C. Chiu, Y. Wu, T. Sainath, and K. Livescu, "A comparison of techniques for language model integration in encoder-decoder speech recognition," in *Proc. of IEEE Spoken Language Technology Workshop (SLT)*, Athens, Greece, 2018, pp. 369–375.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Las Vegas, NV: IEEE, 2016, pp. 2818–2826.

[32] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[33] P. Ghahremani, J. Droppo, and M. L. Seltzer, "Linearly augmented deep neural network," in *Proc. of ICASSP*. Shanghai, China: IEEE, 2016, pp. 5085–5089.