Text and Synthetic Data for Domain Adaptation in End-to-End Speech Recognition

Juan Hussain[1], Christian Huber[1], Sebastian Stüker[1], and Alexander Waibel[1,2]

[1] Interactive Systems Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany
firstname.lastname@kit.edu
[2] Carnegie Mellon University, Pittsburgh PA, USA
alexander.waibel@cmu.edu

Abstract. Neural sequence-to-sequence systems deliver state-of-the-art performance for automatic speech recognition (ASR). When training such systems, one often faces the situation where sufficient amounts of training data for the language in question are available, however, with only small amounts of data for the domain in question. This problem is even bigger for end-to-end speech recognition systems that only accept transcribed speech as training data, which is harder and more expensive to obtain than text data. To alleviate this problem we supplement an end-to-end ASR system with a Text-Encoder which injects text-only input directly into the decoder. In addition, we compare the performance of using text-only input with synthetic speech. Furthermore, we prove for a specific domain that using a very small amount of transcribed speech and a sufficient amount of text-only data from the target domain outperforms adapting with a large amount of domain transcribed speech. Finally, we improve with the Text-Encoder learning new words, e.g., named entities, with no need for any context.

Keywords: Speech Recognition · Domain Adaptation · Text-Encoder

1 Introduction

Lately, end-to-end approaches to automatic speech recognition (ASR) have started to outperform traditional Bayes Classifier based approaches that used neural networks to estimate the emission probabilities of Hidden Markov Models for acoustic modeling and n-gram models for language modeling. The end-to-end approaches can be roughly divided into CTC [8], RNN-T [7] and Sequence-to-Sequence (S2S) [2] models. In this work we will focus on Sequence-to-Sequence models.

S2S models can be adapted to a new domain, given large amounts of transcribed speech data. If such data is not available in sufficient amounts, the question arises how adaptation can be done anyway. One common case is the availability of a small amount of transcribed speech and sufficient amounts of text-only data for the target domain.

In this paper we present experiments with a chosen domain where small amounts of transcribed audio is insufficient for achieving a well-performing adaptation, as we will see in section 4.1. We use additional textual data from the target domain either in conjunction with a multi speaker text-to-speech (TTS) systems (section 3.3) or with a Text-Encoder (section 3.4). Furthermore, we combine the textual domain data with the transcribed speech data from the target domain (section 4.3). Finally, we experiment with the Text-Encoder learning new-words, e.g., named entities where no context is provided. We introduce the method in section 4.3.

2 Related Work

In our previous work [10] we adapted an S2S ASR system with a small amount of domain transcribed speech using a batch weighting scheme, in order to avoid the the problem of catastrophic forgetting during adaptation. The amount of data was sufficient to achieve satisfying results, however, applying the method for other domains with wider language variability yielded insufficient performance. Our work in this paper focuses on enhancing the adaption with text-only data using a Text-Encoder. Several previous works used text input for speech recognition, however, the scenarios considered where different from ours. [11] used text data for semi-supervised learning. The speech and Text-Encoder is shared and supplied with a sub-sampling layer for speech to achieve a similar dimensionality as text. In [4], a separate encoder for text is employed for low resource speech recognition. Adversarial training is used to increase the similarity between speech and text features . Other work employed synthetic audio, as in [21]. They show that the method improves the recognition for utterances with out-of-vocabulary (OOV) words. Other related works enhances the model by re-scoring the output with a language model via two pass decoding. [16] incorporated a multi corpora language model for second pass re-scoring, while [5] and [18] re-score with a second model by attending to the audio or as in [9], in which the second model attends to both the audio and the output using a deliberation model.

3 Method and Training

3.1 Baseline

As our baseline system we use a long short-term memory (LSTM) based sequence-to-sequence model[14]. The encoder consists of six layers, the decoder of two. Before the encoder, two convolutional neural network (CNN) layers with 32 filters and a stride of two are used to down-sample the audio features. The LSTM-layers of the encoder and the decoder have a model dimension of 1024. As output vocabulary, we use a byte-pair encoding (BPE) [20] with 4000 tokens trained on the training data.

3.2 Training and Domain Data

The baseline model is trained on the HOW2 [19] and TED [17] data sets (see table 1). For the adaptation of the model we use the Wall Street Journal (WSJ) data set [6,13]. As text-only data from the Wall Street Journal domain we use [12] and refer to this data set as TXT-WSJ. It contains two million lines of text data. The validation and test set for the baseline (HOW2+TED) and for the target domain (WSJ) can be seen in table 1

3.3 Synthesized Speech

For the TTS system we use Flite [1]. We synthesized audio from the TXT-WSJ data and refer to this as Synthetic TXT-WSJ. Thereby we select for each sentence one of 16 different speakers to obtain speaker variability.

Table 1. Summary of the English speech data-sets

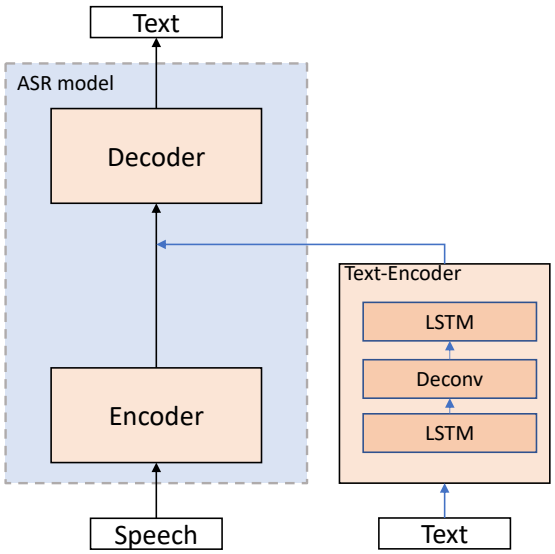
Corpus	Speech data	Utterances
How2+Ted training set	789 h	473K
How2+Ted validation set	18.3 h	11K
WSJ training set	80 h	36k
WSJ validation set	3.2 h	1421
Synthetic TXT-WSJ training set	4500 h	2M
Ted test set	2.6 h	1155
WSJ test set	1.1 h	503

3.4 Text-Encoder

In the following sections we describe the Text-Encoder model as well as the training process for the domain adaptation or learning new words with no context.

Model Architecture While in [11] one encoder is used for both text and speech, we use a separate encoder for each input as figure 1 shows. We take the speech encoder and the decoder from our pre-trained baseline ASR model. The Text-Encoder has a simple architecture consisting of an LSTM-layer followed by a deconvolution layer [15] followed again by an LSTM-layer. Since the length of the text token sequence is much shorter than the speech feature sequence length, we use a deconvolution layer to up-sample the input features. The up-sampling maps the text token sequence to a higher dimensionality similar in length to the speech feature sequence. Thereby, we aim to produce a similar features presentation to the one generated by the speech encoder.

Fig. 1. Model architecture. Left: Baseline ASR model. Right: Text-Encoder architecture in detail.



Training of the Text-Encoder for Domain Adaptation We alternate the training between two cases:

- In the first case, the Text-Encoder is trained with frozen decoder until saturation. We input noised text sequences from a large text corpus to the Text-Encoder and reconstruct the clean sequences by the frozen decoder. The noise we use is applied by masking tokens (substitute them with the masking token) with a probability of 0.2 (similar to [3]).
- In the second case, we train the Text-Encoder and the decoder on both text and speech inputs. In this case we freeze the speech encoder. We apply one pass from the Text-Encoder and one pass from the speech encoder, accumulate the gradient and update the parameters after reaching about 12000 input text tokens.

The reason for freezing the speech encoder in the second case is that we noticed degradation in the performance as the decoder was trying to adapt the speech encoder to generate similar features as the Text-Encoder. Our baseline speech encoder was already trained thoroughly and reached a satisfying level of abstraction for speech features. Therefore, compromises with the Text-Encoder do not benefit the model, instead harm the general performance of it. Besides, using a discriminator for adversarial training as in [4] for motivating the speech encoder to generate a similar output to Text-Encoder was not of advantage in our case. Our suggestion is that we have a different case than in [4]. They train a system for low-resourced-language i.e very few transcribed speech data are available to reach a well trained speech encoder. In our case we have enough

speech data and could reach a very good performing baseline as mentioned above. Our speech encoder is able to reach a sufficient abstraction on general domain.

Training of the Text-Encoder for New-Word Learning We try to use the Text-Encoder to approach the problem of the new words or words not seen during the training (also known as Out-of-Vocabulary Problem OOV). We assume that we neither have context nor audio for those word.

We take the parallel training set which we used for training the baseline. From this set we take only the text. In each text utterance, we insert a word from the new-word list in a random place. We avoid to place the tokens of new words between tokens of a single word in the training utterance. Thereafter, we train the model using this text data with the Text-Encoder. Each time we insert the words in a new random location in order to avoid harming the language model learned by the decoder with fixed not real contexts.
Similar to the training for domain adaption above, we noise the text input of the Text-Encoder. The difference here is that the text is supplied with randomly inserted new words we want to recognize. We noticed here that adding more noise yields better results. Therefore, We mask with the probability 0.3 instead of 0.2 and insert random tokens to the input with the probability of 0.3 within tokens of the training text.
It is worth to mention that we employ here only the transcription of the training set and not big text corpora, to study the effect of our method without adding additional information.

4 Results

4.1 Basic Adaptation Methods

To examine our methods of domain adaptation using text data or synthetic data, we first employ the conventional adapting methods, such as, fine-tuning and batch-weighting [10]. Fine-tuning yields good performance on the new domain but the model suffers from catastrophic forgetting (table 2). For the rest of the experiments we use batch-weighting with ratio 0.9 for the original training data and 0.1 for the new domain data. From table 2 we notice that adapting with 80 hours of data (experiment Batch-Weighting-80) obviously outperforms adapting with only two hours (experiment Batch-Weighting-02).

4.2 Comparison Synthetic Speech and Text-Encoder

Using the Synthetic TXT-WSJ dataset to adapt the model (experiment Batch-Weighting (Synthesis)) improves over the baseline as well as over the approach using only texts from new domain (experiment Batch-Weighting (Text-Encoder)). Another Experiment (Batch-Weighting (Synthesis+Text-Encoder)) shows comparable results. Despite the slight improvement of the above mentioned methods,

we are still far from the results of using additional trancribed audio (section 4.1 and 4.3).

We also tried to use a separate encoder for the synthetic audio initialized with the encoder of the baseline. However, these experiments did not show improvements over using one encoder for both the synthetic and the real audio.

4.3 Results with Text-Encoder

Result for Domain Adaption Remarkably, the results of using two hours (experiment Batch-Weighting-02 (Text-Encoder)) and 80 hours (experiment Batch-Weighting-80 (Text-Encoder)) WSJ are comparable in the case of injecting the decoder with the additional large new domain texts from TXT-WSJ using Text-Encoder (section 3.3). Our interpretation is that only little amount of data is needed for the speech encoder to capture and adapt towards speech features of the new domain. Such features might be the recording channel characteristics or the speaking style. Moreover, the decoder adaptation needs language characteristics of the new domain, which is achievable only with a large text data set, such as, the two million lines TXT-WSJ.

Table 2. Summary of the results.

Method	Additional data	WER Ted	WER WSJ
Baseline	–	7.40	12.60
Fine-tuning	WSJ-80h	10.27	5.55
Batch-Weighting-80	WSJ-80h	7.33	5.51
Batch-Weighting-02	WSJ-2h	7.80	8.54
Batch-Weighting (Synthesis)	Synthetic TXT-WSJ	**7.35**	9.51
Batch-Weighting (Text-Encoder)	TXT-WSJ	7.48	11.34
Batch-Weighting (Synthesis +Text-Encoder)	Synthetic TXT-WSJ + TXT-WSJ	7.54	**9.22**
Batch-Weighting-80 (Synthesis)	WSJ-80h + Synthetic TXT-WSJ	7.21	**4.74**
Batch-Weighting-80 (Text-Encoder)	WSJ-80h + TXT-WSJ	6.98	4.89
Batch-Weighting-02 (Text-Encoder)	WSJ-2h + TXT-WSJ	**6.83**	4.85

Result for new words We experimented the Text-Encoder for learning new words problems as described in section 4.3. A set of 69 words containing mainly named entities. For the test we put the names in a context and recorded them. The baseline model recognizes only 15.9% percent of the new words. After training with the Text-Encoder we obtain 43.5% accuracy of the new word. The WER of the baseline 32.1% is also reduced by the Text-Encoder to 27.1%. Furthermore, the model does not loose the generality as the WER on the TED test-set remains the same.

5 Conclusion

In this work we extend our previous work on domain adaptation with batch-weighting to domains with language variability that need larger amounts of transcribed speech from the target domain. We examined using textual data directly either with a supplemented Text-Encoder or after synthesizing it with a multi-speakers TTS system. We obtain better results with synthesizing text if we do not employ transcribed data from the target domain. However, the results are comparable when using only a small amount of transcribed speech from the target domain. Furthermore, we notice that the performance equalizes when using small or large amount of transcribed speech data as long as we use enough amounts of textual data. The reason might be that the system is able to capture sufficient information from a small amount of transcribed data related to the audio and speaking characteristics. However, the system looks for language modeling information in the large amount textual data. In addition, we were able to improve the recognition of OOV without using additional context. In future works, we will focus on the training mechanism and the model structure of the Text-Encoder to achieve at least the performance of multi-speaker synthesis in scenarios in which no transcribed speech data is available. Furthermore, we will experiment to extract the audio and speaking characteristics with even smaller amounts of transcribed speech from the target domain. For the OOV problem, we will experiment to insert the words in large text corpus or generate contexts for the new-words.

References

1. Black, A.W., Lenzo, K.A.: Flite: a small fast run-time synthesis engine. In: 4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis (2001)
2. Chorowski, J., Bahdanau, D., Serdyuk, D., Cho, K., Bengio, Y.: Attention-based models for speech recognition. arXiv preprint arXiv:1506.07503 (2015)
3. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:1810.04805 (2018)
4. Drexler, J., Glass, J.: Combining end-to-end and adversarial training for low-resource speech recognition. In: 2018 IEEE Spoken Language Technology Workshop (SLT). pp. 361–368. IEEE (2018)
5. Gandhe, A., Rastrow, A.: Audio-attention discriminative language model for asr rescoring. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7944–7948. IEEE (2020)
6. Garofolo, J.S., Graff, D., Paul, D., David, P.: CSR-I (WSJ0) Sennheiser LDC93S6B. `https://doi.org/10.35111/ap42-7n83`. https://doi.org/10.35111/ap42-7n83
7. Graves, A.: Sequence transduction with recurrent neural networks. arXiv preprint arXiv:1211.3711 (2012)
8. Graves, A., Fernández, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In: Proceedings of the 23rd international conference on Machine learning. pp. 369–376 (2006)

9. Hu, K., Sainath, T.N., Pang, R., Prabhavalkar, R.: Deliberation model based two-pass end-to-end speech recognition. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7799–7803. IEEE (2020)
10. Huber, C., Hussain, J., Nguyen, T.N., Song, K., Stüker, S., Waibel, A.: Supervised adaptation of sequence-to-sequence speech recognition systems using batch-weighting. In: Proceedings of the 2nd Workshop on Life-long Learning for Spoken Language Systems. pp. 9–17 (2020)
11. Karita, S., Watanabe, S., Iwata, T., Ogawa, A., Delcroix, M.: Semi-supervised end-to-end speech recognition. In: Interspeech. pp. 2–6 (2018)
12. Linguistic Data Consortium: ACL/DCI LDC93T1. `https://doi.org/10.35111/vdfv-av77`. https://doi.org/10.35111/vdfv-av77
13. Linguistic Data Consortium, NIST Multimodal Information Group: CSR-II (WSJ1) Sennheiser LDC94S13B. `https://doi.org/10.35111/5jkw-xt28`. https://doi.org/10.35111/5jkw-xt28
14. Nguyen, T.S., Stueker, S., Niehues, J., Waibel, A.: Improving sequence-to-sequence speech recognition training with on-the-fly data augmentation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 7689–7693. IEEE (2020)
15. Noh, H., Hong, S., Han, B.: Learning deconvolution network for semantic segmentation. In: Proceedings of the IEEE international conference on computer vision. pp. 1520–1528 (2015)
16. Raju, A., Filimonov, D., Tiwari, G., Lan, G., Rastrow, A.: Scalable multi corpora neural language models for asr. arXiv preprint arXiv:1907.01677 (2019)
17. Rousseau, A., Deléglise, P., Esteve, Y.: Ted-lium: an automatic speech recognition dedicated corpus. In: LREC. pp. 125–129 (2012)
18. Sainath, T.N., Pang, R., Rybach, D., He, Y., Prabhavalkar, R., Li, W., Visontai, M., Liang, Q., Strohman, T., Wu, Y., et al.: Two-pass end-to-end speech recognition. arXiv preprint arXiv:1908.10992 (2019)
19. Sanabria, R., Caglayan, O., Palaskar, S., Elliott, D., Barrault, L., Specia, L., Metze, F.: How2: a large-scale dataset for multimodal language understanding. arXiv preprint arXiv:1811.00347 (2018)
20. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. arXiv preprint arXiv:1508.07909 (2015)
21. Zheng, X., Liu, Y., Gunceler, D., Willett, D.: Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems. In: ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5674–5678. IEEE (2021)