COMBINING END-TO-END AND ADVERSARIAL TRAINING FOR LOW-RESOURCE SPEECH RECOGNITION

Jennifer Drexler, James Glass

Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
32 Vassar Street, Cambridge, MA 02139, USA
{jdrexler,glass}@mit.edu

ABSTRACT

In this paper, we develop an end-to-end automatic speech recognition (ASR) model designed for a common low-resource scenario: no pronunciation dictionary or phonemic transcripts, very limited transcribed speech, and much larger non-parallel text and speech corpora. Our semi-supervised model is built on top of an encoder-decoder model with attention and takes advantage of non-parallel speech and text corpora in several ways: a denoising text autoencoder that shares parameters with the ASR decoder, a speech autoencoder that shares parameters with the ASR encoder, and adversarial training that encourages the speech and text encoders to use the same embedding space. We show that a model with this architecture significantly outperforms the baseline in this low-resource condition. We additionally perform an ablation evaluation, demonstrating that all of our added components contribute substantially to the overall performance of our model. We propose several avenues for further work, noting in particular that a model with this architecture could potentially enable fully unsupervised speech recognition.

Index Terms: speech recognition, low-resource, end-to-end, semi-supervised learning

1. INTRODUCTION

While speech recognition technology has improved rapidly in recent years, it remains out of reach for the majority of the world's languages. In many of these so-called "low-resource" languages, the data necessary to train a traditional speech recognition system is simply not available. In particular, training an acoustic model - a key part of a traditional speech recognition system - requires resources like pronunciation dictionaries that are often non-existent or prohibitively expensive to acquire. The need to produce these resources for every language of interest does not scale to the goal of making speech recognition universally available.

Recent end-to-end neural network models have, in part, solved this problem: they have eliminated the need for pronunciation dictionaries. Instead of expertly crafted linguistic knowledge, they rely on word-level speech transcripts that can be produced with reasonable accuracy by a native speaker with minimal training. In return for this reduced need for linguistic expertise, however, these systems typically require hundreds up to many thousands of hours of transcribed speech for training. While these data are easier to collect than those needed to train an acoustic model, this process still does not scale to new languages with minimal existing resources.

In this paper, we extend end-to-end models to smaller ASR corpora by focusing on a low-resource paradigm that is common to many languages but has not received much research attention: a small amount of transcribed speech along with much larger corpora of non-parallel speech and text. Both text and speech data are widely available on the Internet; while there is much prior work using large text corpora to improve ASR performance [1], the potential uses of untranscribed speech, especially for end-to-end models, have gone largely unexplored. We believe that this paradigm of semi-supervised ASR presents a clear path towards making ASR technology available in many currently underserved languages.

Here, we introduce a novel neural network architecture for this task, supplementing an end-to-end speech recognition model with additional components that effectively take advantage of non-parallel speech and text. In addition to an encoder-decoder ASR model, we train a text autoencoder that shares parameters with the ASR decoder and a speech autoencoder that shares parameters with the ASR encoder. We explicitly tie these together with adversarial training to encourage the speech and text encoders to share the same hidden embedding space. We demonstrate significant performance improvements over a baseline end-to-end model trained on the same data. We also analyze the factors contributing to those improvements.

This work borrows many of its ideas from recent work in machine translation which demonstrated that sequence-to-sequence neural network models can be trained without parallel data [2, 3]. Our model does not have some of the features that allowed those models to be trained in a fully unsuper-

vised fashion, but has the potential to enable ASR without transcribed speech with some modifications. While we do not explore this possibility here, we intend to pursue it in future work.

2. PRIOR WORK

On very large speech recognition tasks, end-to-end neural network ASR models have recently overtaken traditional models, which we define here as separately trained acoustic and language models which are combined in a hidden Markov model (HMM)[4]. Despite these recent advances, end-to-end models lag far behind traditional models in the low-resource space. Rosenberg et al. [5] compared a traditional HMM-DNN system against both CTC-based[6] and attention-based[7, 8] end-to-end models on eight low-resource languages, each with 40 hours of training data, and found that the traditional model outperformed both end-to-end architectures across all languages. The uses of additional text and speech data for low-resource ASR have also been explored in the context of traditional ASR models (see [1] and [9] respectively) but have seen limited research for end-to-end models.

The low-resource scenario explored in this paper closely resembles the one used in two recent papers from Tjandra et al.[10, 11]. In those papers, the authors train both speech recognition and text-to-speech (TTS) systems with their small transcribed speech corpus. They then use the ASR system to turn untranscribed speech into a synthetic training set for their TTS model. Similarly, they create a training set for the ASR model from stand-alone text using their TTS system. These synthetic datasets are used to further train their models; the new models are then iteratively used to generate better synthetic training data. As in our paper, Tjandra et al. [11] create a semi-supervised corpus from the Wall Street Journal speech recognition corpus[12] - treating a portion of the original dataset as 'parallel' and the remainder as 'non-parallel'. Using this method, Tjandra et al. achieve remarkable results: their semi-supervised model closes 73% of the gap between the character error rate of their low-resource baseline and high-resource topline results.

While these results are encouraging, the fact that the non-parallel speech and text come from the same dataset means that they are not truly independent from each other. In our experiments, we enforce independence between the non-parallel speech and text by ensuring that there is no overlap between the underlying utterances used for each modality. This creates a more difficult task, but one that must be addressed for real-world low-resource scenarios.

Our work also shares many features with recent work in unsupervised machine translation (MT). Both [2] and [3] start with a common architecture for end-to-end supervised MT and add components to allow for unsupervised training. Specifically, non-parallel texts in the source and target languages are used to train separate encoder-decoder sequence to sequence autoencoder models. Adversarial training (see [13] for an overview) is used to push the hidden representations in the two different encoders to use the same embedding space. Thus, a decoder trained only as part of an autoencoder in the target language can also decode the outputs of the source language encoder.

The success of these unsupervised MT models relies on a key observation: word embedding spaces tend to have similar structure across languages [14]. Conneau et al. [15] use this fact to learn an unsupervised mapping from the embedding space of one language to the embedding space of the other, then use that mapping to learn a dictionary. This dictionary can then be used to seed the training of a fully unsupervised MT system. We construct a similar model but use a small corpus of transcribed speech to seed the training of a semi-supervised ASR system.

While we do not experiment with fully unsupervised ASR here, this model architecture should support such experimentation in the future. In that sense, our work shares some goals with recent efforts in zero resource speech processing, specifically the Zero Resource Challenge [16]. That work is more theoretical than practical: its goal is to explore what can be learned about speech from speech only. Some of that work suggests an eventual path towards unsupervised speech recognition [17, 18], but so far no paradigm exists for connecting the speech domain to the text domain in an unsupervised way, given that the Challenge focuses on systems trained from speech only. Through our use of both speech and text corpora, we see a clear path from our work to a future fully unsupervised speech recognition system.

3. MODEL ARCHITECTURE

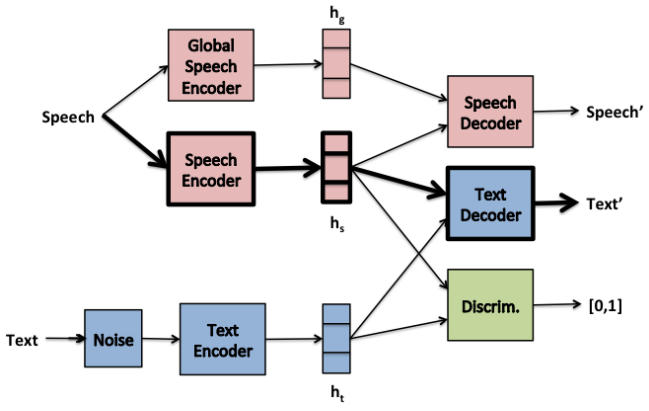


Fig. 1: Semi-supervised ASR model architecture. Speech-to-text model is outlined in bold; text autoencoder is shaded blue; speech autoencoder is shaded red; discriminator for adversarial training is shaded green.

The architecture of our semi-supervised speech recognition model is shown in Figure 1. The components of the core speech recognition system are outlined in bold. These components are trained together, end-to-end, using transcribed speech. Our model also contains a de-noising text autoencoder, shown in blue, which shares the baseline decoder and is trained in an unsupervised fashion using only text. The speech autoencoder is shown in red - it shares the baseline encoder and is trained from untranscribed speech. Finally, our model has a classifier, colored green, used as a discriminator for adversarial training of the outputs of the speech and text encoders. Each component of this system - the speech-to-text model, the text autoencoder, the speech autoencoder, and the adversarial training - is described in detail below.

3.1. Baseline ASR Model

Our baseline speech recognition model is a sequence-to-sequence neural network model composed of a speech encoder and a text decoder with attention. The model is almost identical to the Listen, Attend, and Spell (LAS) model described in [7] and is also very similar to the model from [8].

As in [7], our speech encoder is a recurrent neural network with four layers. The first is a bi-directional long short-term memory (bLSTM) layer [19]. The three subsequent layers are pyramidal bLSTM layers [7], each of which downsamples its input sequence by a factor of two. The encoder takes as input a sequence of simple acoustic features and outputs a sequence of hidden representations, one for every eight input frames.

The architecture of our text decoder also follows the model described in [7]. At its core, it is a recurrent neural network with two LSTM layers. At each time-step, the decoder takes as input the previous character in the sequence (or a special 'start' token at the first time-step) and outputs a probability distribution over the next character. It also takes as input a context vector, created by a trainable attention mechanism from the previous hidden state of the decoder, the previous context vector, and the output sequence from the encoder. This type of attention mechanism was first introduced in the machine translation context in [20] and its use in ASR is explained more fully in [7] and [8].

The speech-to-text model is trained end-to-end to maximize the log likelihood of the correct character sequence during training. During inference, we use beam search decoding [21] to find a high-likelihood transcription of each text utterance. There is a well-known discrepancy between maximum likelihood training and decoding: during decoding, the ground truth sequence of characters is not available and characters sampled from the output distribution are instead fed to the model. To mitigate this effect, we sometimes input a sampled character rather than the ground truth character during training. As in [7], we use a fixed 10% sampling rate, meaning that 10% of the input characters to the decoder during training are sampled from the output distribution of the decoder at the previous time-step.

3.2. Text Autoencoder

The text autoencoder has three components, shown in blue in Figure 1: a noise model, an encoder and a decoder. The decoder is shared with the speech-to-text model described in the previous section - training this autoencoder also trains the parameters of our ASR decoder.

The text encoder takes one-hot character encodings as input. It is composed of a single embedding layer [22] - to convert these inputs to character vectors - followed by two bLSTM layers. As a character autoencoder is a relatively trivial learning task, we add noise to the input before feeding it the encoder, following [2]. While Lample et al. [2] both delete words and shuffle their order when adding noise for unsupervised MT, we empirically found that deleting characters but not shuffling them was most effective here. We drop characters with probability $p = 0.2$. The text autoencoder is trained with the same end-to-end maximum likelihood objective as the speech recognition model, with the same sampling procedure.

3.3. Speech Autoencoder

The speech autoencoder is shown in red in Figure 1. Speech requires a different autoencoder architecture than text, because the speech signal contains both linguistic and non-linguistic information. The outputs of the ASR encoder should contain only linguistic information, but we also need to capture the non-linguistic information in order to train an autoencoder. As in [23], we build a hierarchical autoencoder: one encoder to capture the aspects of the signal that change over time (namely, linguistic content), and one encoder to capture the utterance-level properties of the signal (non-linguistic characteristics). Here, our original ASR encoder serves that first purpose, and a global speech encoder serves the second purpose. The output of the global encoder is appended to the output of the original speech encoder at every time step.

Our global encoder is a convolutional neural network (CNN) [24] with three layers. Each layer has a convolution, batch normalization, rectified linear unit (ReLU) non-linearities, and max-pooling. The first layer performs convolution in frequency, the next two perform convolution in time. The layers look at increasingly larger segments of speech - from a single frame at the lowest layer to 45 frames at the highest. The last layer pools over the entire utterance to generate a single utterance-level representation vector.

The speech decoder is a simple feed-forward neural network with two leaky ReLU layers and a linear layer on top. It takes as input a single vector - a concatenation of the output from the global encoder and a single output from the ASR encoder - and generates eight frames of output speech features.

These are scored against the eight frames of input speech features that produced the given ASR encoder output.

The speech autoencoder is trained end-to-end using smoothed L1 loss, an element-wise loss which equals the L1 loss when the difference between the output and the label is greater than one and equals the L2 loss when that difference is less than one.

3.4. Adversarial Training

Adversarial training [13] is a technique originally developed for training generative models to produce examples whose distribution matches a ground-truth data distribution. In the standard case, an adversarial model has two components - a generator and a discriminator - which are trained alternately. The discriminator is trained to differentiate between examples from the data distribution and outputs from the generator. The generator is trained to 'trick' the discriminator and generate examples that the discriminator will score as likely to have come from the data distribution.

Here, we instead use adversarial training to encourage the outputs of the speech and text encoders to share an embedding space: we have two generators (the encoders) and no data distribution. While there are a number of ways to potentially modify the original adversarial 'game' for our scenario, we chose to treat the output of the text encoder as the data distribution. We theorize that this will further encourage the outputs of the speech encoder to contain only linguistic information. This adversarial training can be performed with all available text and speech data, as it does not assume any parallelism.

In our model, the discriminator is a simple feed-forward network with two fully-connected layers. It takes as input a single vector, and outputs a single real-valued score in the interval $[0, 1]$. The discriminator is trained using binary cross-entropy loss to assign high scores to output vectors generated by the text encoder and low scores to those generated by the speech encoder. We use label smoothing, as recommended in [25].

4. DATA AND TOOLS

For all experiments in this paper, we use the Wall Street Journal (WSJ) corpus, a standard speech recognition benchmark with many comparable results available in the literature [12]. For training, we use the SI284 set, which consists of 81 hours of read speech; each utterance is a single spoken sentence from the Wall Street Journal newspaper. This training corpus includes 37.4K utterances, spread across 284 speakers. We use the standard dev93 and eval92 sets for validation and test, respectively - all reported results are on the eval92 set. For results that incorporate a language model, we use the text corpus that accompanies the WSJ speech recognition corpus.

For our semi-supervised experiments, we divide the training data by powers of two, selecting the utterances so that all 284 speakers are represented in all training corpora. In particular, we present results from three semi-supervised conditions - 2.5, 5, and 10 hours of transcribed speech, representing approximately $1/8^{\text{th}}$, $1/16^{\text{th}}$, and $1/32^{\text{nd}}$ of the original corpus, respectively. In all semi-supervised experiments, we use all speech from the original training corpus and a 37.4K sentence subset of the language model training text as our non-parallel speech and text datasets. We ensure no overlap between the text of the ASR training corpus and our non-parallel text corpus.

The speech input to our model is log-Mel filterbank features, with 40 filters per 25ms frame, calculated at a 10ms frame rate. We normalize all text to contain only alphanumeric characters, along with a 'SPACE' symbol and comma, period, and apostrophe symbols. While it is non-standard, we remove the '<NOISE>' tags from all speech transcripts for compatibility with the language model text, which does not include such tags.

All recurrent layers in all components have 256 units. Character embeddings have 128 units. The discriminator also has 256 units per layer. The convolutional layers in the global speech autoencoder have 32, 64, and 256 filters and kernels of size $(36, 1)$, $(1, 5)$, and $(1, 3)$. All have a stride of one. The first convolutional layer pools over three frames, the second over five inputs (15 frames), and the third over the entire utterance. We use batches of size 32 for discriminator training and 16 for all other training. For optimization, we use stochastic gradient descent (SGD) with momentum [26] and a learning rate of 0.2. We stop training when performance on the validation set stops improving. For all results, we use a beam of size 20 during decoding.

For language model experiments, we used a 3-gram word-level language model. As in [8], we compose the language model finite state transducer (FST) with a lexicon that spells out each vocabulary word, to produce a character-level language model that can be incorporated into the beam search decoding. Again following [8], we use a language model weight of 0.5 and word bonus of 1 when decoding with a language model.

Our model was implemented in PyTorch [27], based in part on the OpenNMT machine translation toolkit [28]. Speech features were computed using Kaldi [29]. The language model was produced using the Kaldi WSJ recipe.

5. RESULTS AND DISCUSSION

Our main results are in Figure 2. We compare our a baseline attention-based ASR model (blue) with our proposed architecture (green and red). All three model architectures were trained separately on 2.5, 5, and 10 hours of transcribed speech. The semi-supervised model (red bar) also our larger non-parallel corpus for the autoencoders and adversarial

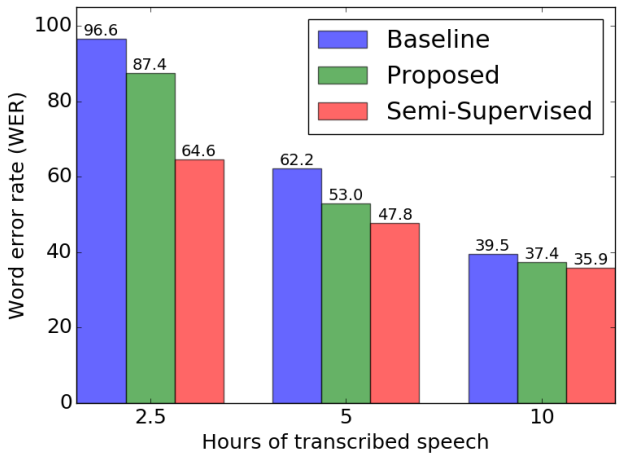


Fig. 2: Word error rate (WER) by model architecture and hours of transcribed training data. The red and green bars both represent the results of our proposed architecture. The semi-supervised model (red) uses our larger non-parallel dataset in addition to the transcribed speech.

training.

The baseline results denoted by the blue bars show clearly the impact of limited training data size. Trained on the full WSJ set, this baseline model achieves a word error rate (WER) of 16.6 and character error rate (CER) of 5.8, in line with previously reported similar models (for example, [8] reports a WER of 18.0). Using only 10 hours of transcribed speech degrades that performance to a WER of 39.5 and CER of 16.1. With only 2.5 hours of training data, the baseline model barely learns anything, resulting in a WER of 96.4 and CER of 83.0.

When trained on the same data as the baseline model, our architecture (green bars) produces improved results in all training conditions. We see significant additional improvements through the use of the extra non-parallel data, as shown by the red bars. The impact of this semi-supervised training is highest when the least parallel training data is available; with 10 hours of transcribed speech, the improvements due to semi-supervised training are modest.

We perform an ablation study to understand which components of our model have the most impact on performance. These results are in Table 1. All models used for this table were trained with 2.5 hours of transcribed speech. In addition to the three models from 2, we experiment with two semi-supervised models that each have a single feature of the complete model removed. We did not experiment with removing the text autoencoder because it is so integral to the model: without the text autoencoder, we cannot perform adversarial training or make any use of the additional text data.

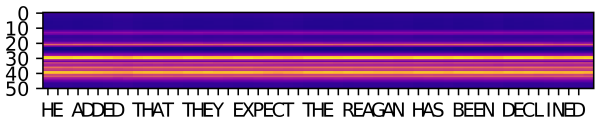
The first three rows of Table 1 mirror the left-hand side of Figure 2. The fourth row shows that removing the speech

Table 1: Ablation results. All models were trained on 2.5 hours transcribed speech.

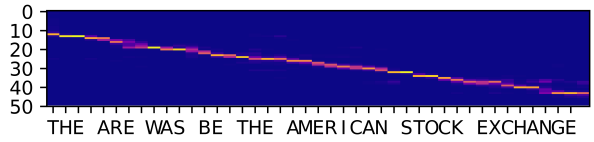
	WER	CER
Baseline model	96.4	83.0
Proposed model:		
with transcribed speech only	87.4	62.0
with parallel and non-parallel data	64.6	34.6
without speech autoencoder	69.0	42.0
without adversarial training	93.3	67.0

autoencoder degrades the performance of our model somewhat, but still allows for a significant improvement over the baseline. We use a relatively simple speech decoder here, and plan to experiment with more complex models in future work, which we hope will elicit further gains.

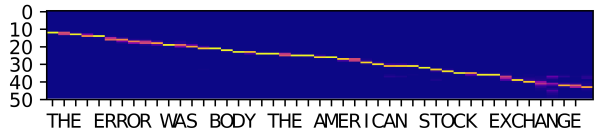
The final row of Table 1 shows that removing the adversarial training sharply reduces the performance of our model, demonstrating that having a shared embedding space for the outputs of the speech and text encoders is critical. Without adversarial training, the text corpus is still used to expose the decoder to a wider range of possible sentences, which likely accounts for the small improvement over the baseline using this model.



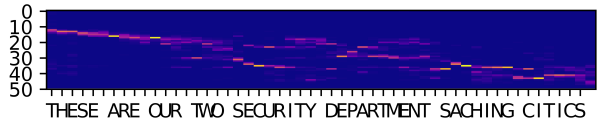
(a) Baseline model



(b) Proposed model with no non-parallel data



(c) Proposed semi-supervised model



(d) Proposed semi-supervised model without adversarial training

Fig. 3: Attention weights during decoding for WSJ utterance 443c040c. The ground truth transcript is: THE ERROR WAS BY THE AMERICAN STOCK EXCHANGE.

In order to further understand the improvements generated by the use of our model architecture, we inspect the attention mechanism during decoding. Figure 3 illustrates the activity of the attention mechanism during decoding of the same test sentence by four different models. In each subfigure, the y-axis represents the outputs from the speech encoder - one for every 8 frames, or 80ms, of speech - while the x-axis represents outputs from the decoder. The lowest weights are shown in blue, while the highest are shown in yellow.

When the baseline model is trained on only 2.5 hours of speech, the attention mechanism has the same weights at all time-steps (Figure 3a) - it has not learned anything about the correspondence between speech and text. When our proposed model architecture is trained with the same data, however, the attention mechanism has clearly learned quite a bit, as shown in Figure 3b.

Incorporating additional non-parallel data - as in Figure 3c - allows our model to learn more confident alignments between speech and text. However, when we remove the adversarial training (Figure 3d), the model struggles to find the correspondence between speech and text. The adversarial training is essential to this method of incorporating additional text data into ASR training.

Our final set of experiments compares our model with a more traditional method of incorporating additional text data: the inclusion of an external language model during decoding. These results are in Figure 4 - the original results from Figure 2 are shown in lighter colors with the corresponding language model results superimposed on top. For reference, our baseline model trained on the full WSJ corpus achieves a WER of 10.5 when combined with a language model, in comparison to the WER of 10.8 reported in [8].

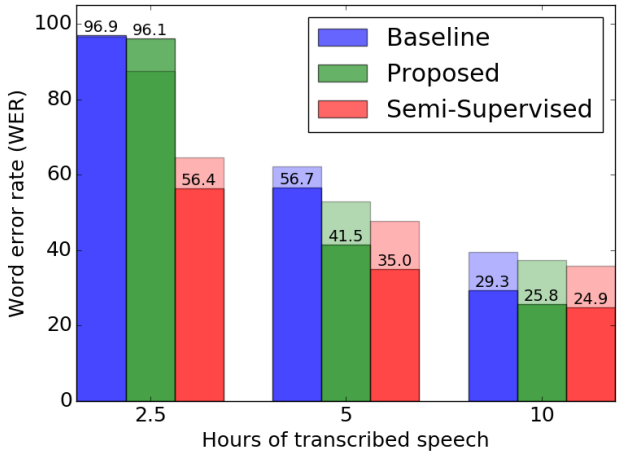


Fig. 4: Word error rate (WER) by model architecture and hours of transcribed training data. The red and green bars both represent the results of our proposed architecture. The semi-supervised model (red) uses the full WSJ dataset as non-parallel text and speech.

As would be expected based on the behavior of the attention mechanism, the baseline model trained on 2.5 hours of transcribed speech does not improve when combined with a language model, while the semi-supervised model does. We get significant improvement with a language model on all model trained on either five or ten hours of transcribed speech; the improvement is greater - both relative and absolute - for all semi-supervised models compared to the baseline. Importantly, the gains achieved through our method of incorporating text data are complimentary with the gains due to the language model.

Somewhat surprisingly, adding a language model significantly hinders the performance of our model architecture trained on 2.5 hours of transcribed speech with no non-parallel data. On further inspection, we find that the language model overwhelms the speech recognition model in this case. Finding the optimal parameters to balance these models is beyond the scope of this paper, but the issue is an important one for our suggested low-resource scenario and warrants further research.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed an effective model for semi-supervised ASR with limited transcribed speech and larger, separate speech and text corpora. We have also shown that our model architecture can be beneficial for fully supervised ASR in very low-resource scenarios.

This paper represents initial experimentation with the use of this class of model architecture for speech recognition. The ASR literature suggests a range of possible adjustments to the model that could yield improvements in future work: attention windowing[8], joint CTC and attention-based decoding[30], decoder pre-training[31], and many more.

We are also eager to further analyze the performance of this model, especially the nature of the shared text and speech embedding space at its center. Additionally, we hope to explore how the properties of this embedding space (as well as our overall performance) change with the use of subword units[32] rather than characters for the input text.

This model architecture also has the potential for fully unsupervised training, which we are eager to explore in future work. In particular, we are hopeful that this model in combination with the speech chain model from Tjandra et al.[11] could prove effective for fully unsupervised speech recognition.

7. REFERENCES

[1] G. Mendels, E. Cooper, V. Soto, J. Hirschberg, M. J. Gales, K. M. Knill, A. Ragni, and H. Wang, "Improving speech recognition and keyword search for low resource languages using web data," in *Sixteenth Annual*

Conference of the International Speech Communication Association, 2015.

[2] G. Lample, L. Denoyer, and M. Ranzato, "Unsupervised machine translation using monolingual corpora only," *arXiv preprint arXiv:1711.00043*, 2017.

[3] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised neural machine translation," *arXiv preprint arXiv:1710.11041*, 2017.

[4] C.-C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, K. Gonina *et al.*, "State-of-the-art speech recognition with sequence-to-sequence models," *arXiv preprint arXiv:1712.01769*, 2017.

[5] A. Rosenberg, K. Audhkhasi, A. Sethy, B. Ramabhadran, and M. Picheny, "End-to-end speech recognition and keyword search on low-resource languages," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 5280–5284.

[6] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*. ACM, 2006, pp. 369–376.

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

[8] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, and Y. Bengio, "End-to-end attention-based large vocabulary speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4945–4949.

[9] T. Fraga-Silva, J.-L. Gauvain, and L. Lamel, "Lattice-based unsupervised acoustic model training," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4656–4659.

[10] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Automatic Speech Recognition and Understanding Workshop (ASRU), 2017 IEEE*. IEEE, 2017, pp. 301–308.

[11] ——, "Machine speech chain with one-shot speaker adaptation," *arXiv preprint arXiv:1803.10525*, 2018.

[12] LDC. (1994) Wall street journal corpus. [Online]. Available: https://catalog.ldc.upenn.edu/ldc94s13a

[13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[14] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[15] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, "Word translation without parallel data," *arXiv preprint arXiv:1710.04087*, 2017.

[16] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results," *Procedia Computer Science*, vol. 81, pp. 67–72, 2016.

[17] H. Kamper, A. Jansen, and S. Goldwater, "Fully unsupervised small-vocabulary speech recognition using a segmental bayesian model," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[18] ——, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

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

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

[21] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[22] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[23] W.-N. Hsu, Y. Zhang, and J. Glass, "Unsupervised learning of disentangled and interpretable representations from sequential data," in *Advances in neural information processing systems*, 2017, pp. 1876–1887.

[24] Y. LeCun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The handbook of brain theory and neural networks*. MIT Press, 1998, pp. 255–258.

[25] I. Goodfellow, "Nips 2016 tutorial: Generative adversarial networks," *arXiv preprint arXiv:1701.00160*, 2016.

[26] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *International conference on machine learning*, 2013, pp. 1139–1147.

[27] A. Paszke, S. Chintala, R. Collobert, K. Kavukcuoglu, C. Farabet, S. Bengio, I. Melvin, J. Weston, and J. Mariethoz, "Pytorch," 2017.

[28] G. Klein, Y. Kim, Y. Deng, J. Senellart, and A. M. Rush, "Opennmt: Open-source toolkit for neural machine translation," *arXiv preprint arXiv:1701.02810*, 2017.

[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 *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[30] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4835–4839.

[31] C. Gulcehre, O. Firat, K. Xu, K. Cho, L. Barrault, H.-C. Lin, F. Bougares, H. Schwenk, and Y. Bengio, "On using monolingual corpora in neural machine translation," *arXiv preprint arXiv:1503.03535*, 2015.

[32] R. Sennrich, B. Haddow, and A. Birch, "Neural machine translation of rare words with subword units," *arXiv preprint arXiv:1508.07909*, 2015.