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

Completely Unsupervised Phoneme Recognition By A Generative Adversarial Network Harmonized With Iteratively Refined Hidden Markov Models

Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-Yi Lee, Lin-shan Lee

Graduate Institute of Communication Engineering, National Taiwan University
{r06942070, r06922039, r07942148}@ntu.edu.tw, tlkagkb93901106@gmail.com,
lslee@gate.sinica.edu.tw

Abstract

Producing a large annotated speech corpus for training ASR systems remains difficult for more than 95% of languages all over the world which are low-resourced, but collecting a relatively big unlabeled data set for such languages is more achievable. This is why some initial effort have been reported on completely unsupervised speech recognition learned from unlabeled data only, although with relatively high error rates. In this paper, we develop a Generative Adversarial Network (GAN) to achieve this purpose, in which a Generator and a Discriminator learn from each other iteratively to improve the performance. We further use a set of Hidden Markov Models (HMMs) iteratively refined from the machine generated labels to work in harmony with the GAN. The initial experiments on TIMIT data set achieve an phone error rate of 33.1%, which is 8.5% lower than the previous state-of-the-art.

Index Terms: Phoneme Recognition, Unsupervised Learning, Generative Adversarial Network, Hidden Markov Models.

1. Introduction

Automatic speech recognition (ASR) has achieved remarkable performance and been widely used. However, the state-of-the-art ASR systems [1, 2] have to learn from massive annotated data which is difficult to obtain for at least 95% of the languages over the world which are low-resourced. Conversely, collecting a relatively big unlabeled corpora for such languages is more achievable in the big data era. This is why unsupervised ASR is attractive.

Substantial effort was made to learn signal representations directly from speech signals in an unsupervised way, which could be a good step towards unsupervised ASR, and has been shown useful in various tasks, such as speaker identification [3], spoken term detection [4, 5, 6], and spoken document retrieval [7, 8, 9]. In particular, a sequence-to-sequence auto-encoder [10, 11] was used to embed audio segments into vectors of fixed-dimensionality. Inspired by Word2Vec [12], it was shown possible to encode some semantics into such audio embeddings [13]. But there still exists a wide gap between all these works and unsupervised ASR.

On the other hand, unsupervised neural machine translation was very successful recently [14, 15, 16], in which a mapping relationship between source and target language word embedding spaces could be learned with adversarial training [17] in an unsupervised manner. This led to several attempts on unsupervised ASR, since ASR is also a kind of translation. This included aligning audio and text embedding spaces for the purpose of unsupervised ASR [18, 19, 20], and our prior work [21] of unsupervised phoneme recognition, with a Generative Adversarial Network (GAN) [17], by clustering audio embeddings into a set of tokens, and learning the mapping relationship between tokens and phonemes.

In the above efforts, it was realized that the primary difficulties for applying unsupervised neural machine translation model for ASR purposes were the segmental structure of the audio signals, i.e. each word or phoneme consists of a segment of consecutive frames of variable length with unknown boundaries in the signal, and ASR is supposed to map such a segmental structure to a sequence of discrete words or phonemes. This is why oracle or forced alignment audio segmentation boundaries was usually helpful to achieve satisfactory performance in these works.

This above problem was properly handled previously by a specially designed cost function called Segmental Empirical Output Distribution Matching, which considered both the n-gram probabilities across the output units and the intra-segment frame-wise smoothness [22]. However, this approach required a very large batch size during training to avoid being biased [23], and became difficult when the training data size grew. Furthermore, the n-gram probabilities considered here only included local statistics in the output sequences, while other information such as long-distance dependency were inevitably ignored.

In this paper, we propose to handle the above problem by a framework using Generative Adversarial Network (GAN) harmonized with a set of iteratively refined hidden Markov models (HMMs). Only unlabeled utterances and unrelated text sentences are needed, but not segment boundaries at all. The overall framework is shown in Fig. 1. The GAN includes a generator and a discriminator iteratively learning from each other. The generator consists of two parts: (a) a frame-wise phoneme classifier and (b) a sampling process. (a) transforms a sequence of acoustic features into a sequence of frame-wise phoneme prediction. Then based on the present segmentation, (b) samples a phoneme prediction from each segment to generate the predicted phoneme sequence. The discriminator is trained to distinguish the predicted phoneme sequences from the real phoneme sequences obtained from text sentences. On the other hand, we use the generator output to train a set of HMM models (not shown in Fig. 1), and use these HMMs to refine the segmentation on the training set, which are used to learn better generator and discriminator. The harmonized process of training GAN and HMM models improves the performance iteratively.

This framework works easily with a very large data set, and the discriminator considers all possible information from the text data set, not limited to the n-gram local statistics. This framework achieved 33.1% phone error rate (PER) on TIMIT in the preliminary experiments, which is 8.5% lower than the previous state-of-the-art [22].

2. Proposed framework

Below we describe the GAN architecture and training loss function in section 2.1 and section 2.2 and the harmonized HMMs in section 2.3.

http://dx.doi.org/10.21437/Interspeech.2019-2068

2.1. GAN model architecture

The Generative Adversarial Networks (GAN) [17, 24, 25] consists of a generator $\mathcal{G}$ and a discriminator $\mathcal{D}$. The discriminator learns to distinguish the generator output from real phoneme sequences, while the generator learns to produce phoneme sequences which can ”fool” the discriminator. So the generator and the discriminator learn from each other iteratively. As shown at the middle of Figure 1, the generator has two parts: a frame-wise phoneme classifier and a sampling process.

An input feature sequence $\mathbf{x} = \{x_1, ..., x_T\}$ is fed to the phoneme classifier, producing predicted phoneme distribution sequence $\mathbf{y} = \{y_1, ..., y_T\}$, where x_t is a d-dimensional acoustic feature at time t, y_t is a probability distribution over all possible M phonemes at time t, and T is the input sequence length. This classifier is a context-dependent DNN network [26], or an Recurrent neural network (RNN).

Sampling process is then applied to address the segmental structure issue. Any unsupervised audio segmentation approach can be used to produce an initial segmentation of the input sequence $\mathbf{x}$ mentioned above, $\mathbf{S} = \{S_1, ..., S_L\}$, where S_l is the lth segment and L is the total number of segments. We randomly sample a phoneme distribution y_t from each segment to generate a phoneme distribution sequence, which is referred to as the generated phoneme sequence and denoted as $P^{gen} = \{y_{t_1}, ..., y_{t_L}\}$, where y_{t_l} is the phoneme distribution sample from the segment S_l.

The discriminator is to discriminate between the real phoneme sequence P^{real} (one-hot) from a text data set and the generated phoneme sequence P^{gen} from the generator giving an output scalar. The higher the scalar, the more probable it is a real phoneme sequence. This discriminator is a two-layer CNN.

2.2. Training loss

There are two objectives for the training loss. First, the generated phoneme distribution sequences should be very close to those of the real phoneme sequences obtained with text data, which is the target of the GAN. Second, the phoneme distributions for frames in the same segment should be very close to one another, which leads to a loss term referred to as intra-segment loss here.

2.2.1. Discriminator loss

The loss for training the discriminator $\mathcal{D}$ follows the concept of Wasserstein GAN [24] with gradient penalty [27]:

$$\mathcal{L}_D = \frac{1}{K}\sum_{k=1}^{K} D(P^{gen(k)}) - \frac{1}{K}\sum_{k=1}^{K} D(P^{real(k)}) + \alpha\mathcal{L}_{gp}, \quad (1)$$

where $D(P)$ is the scalar output of the discriminator for an input sequence P, larger for a real phoneme sequence, K is the number of training examples in a batch, and k is the example index. α is a weight for the gradient penalty $\mathcal{L}_{gp}$.

$$\mathcal{L}_{gp} = \frac{1}{K}\sum_{k=1}^{K}(||\nabla D(P^{inter(k)})|| - 1)^2, \quad (2)$$

where P^{inter} is from real phoneme sequence P^{real} and a generated phoneme sequence P^{gen} with random weights between 0 and 1, and this term is useful in stabilizing the training.

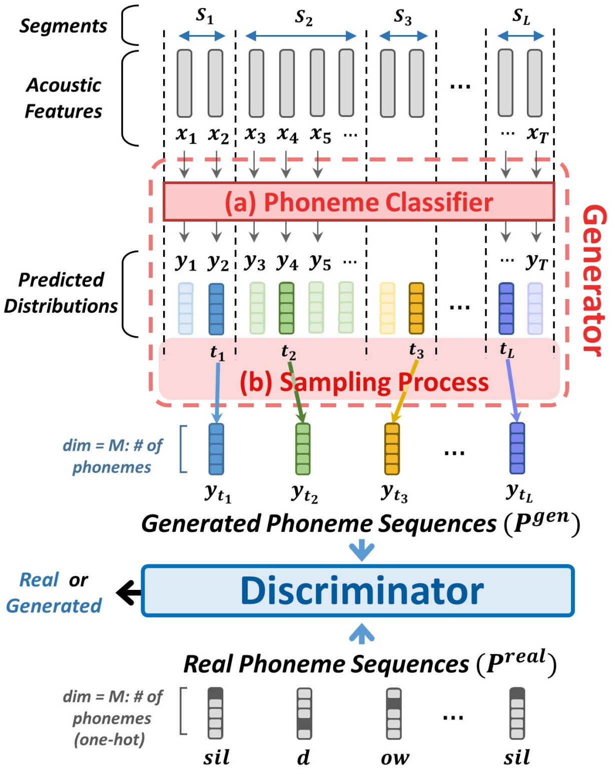
Segments
S_1
S_2
S_3
S_L
Acoustic Features
x_1 x_2 x_3 x_4 x_5 ··· ··· x_T
(a) Phoneme Classifier
Generator
Predicted Distributions
y_1 y_2 y_3 y_4 y_5 ··· ··· y_T
t_1 t_2 t_3 t_L
(b) Sampling Process
dim = M: # of phonemes
y_{t_1} y_{t_2} y_{t_3} y_{t_L}
Generated Phoneme Sequences (P^{gen})
Real or Generated
Discriminator
Real Phoneme Sequences (P^{real})
dim = M: # of phonemes (one-hot)
sil d ow sil

Figure 1: *Overview of the proposed approach. The generator includes (a) phoneme classifier transforming the acoustic features into predicted phoneme distributions, and (b) a phoneme distribution sampled from each segment. The discriminator is trained to distinguish between the generated and real phoneme sequences. The HMMs are not shown.*

2.2.2. Generator loss

Different from the original Wasserstein GAN, here we introduced an intra-segment loss:

$$\mathcal{L}_{intra} = \frac{1}{K}\sum_{k=1}^{K}\sum_{i,j\in\mathbf{S}_k}(y_i - y_j)^2, \quad (3)$$

so the phoneme distributions for frames within the same segment can be more homogeneous. The combined generator loss is:

$$\mathcal{L}_G = -\frac{1}{K}\sum_{k=1}^{K} D(P^{gen(k)}) + \lambda\mathcal{L}_{intra}, \quad (4)$$

where λ is a weight. The generator and the discriminator are iteratively trained to learn from each other, so the phoneme classifier in the generator is eventually able to map acoustic feature sequences to phoneme sequences ”looking real”.

The inference is then performed when the GAN is well trained. We simply map the acoustic feature sequence $\mathbf{x}$ of the training set to the corresponding phoneme distribution sequences $\mathbf{y}$, pick up the phoneme with the highest probability for each frame, and select the phoneme picked from frames within a segment with the highest probability as the phoneme recognition result. The phoneme recognition result can also be obtained using available decoders such as WFST, which includes lexicon and language model information. For example, the segmentation boundaries are even not needed in WFST decoders.

2.3. Harmonization with iteratively refined HMMs

When the training set is decoded into phoneme sequences by a set of well trained GAN as above, these GAN-generated phoneme sequences are taken as labels for the training set to train a set of phoneme HMMs. This set of phoneme HMMs are then used to re-transcribe the training set by force alignment into new phoneme sequences with new segmentation boundaries, which are then used to start a new iteration of GAN training as described in sections 2.1 and 2.2, then train a refined set of HMMs as mentioned above. This GAN/HMM harmonization procedure can be performed iteratively, as depicted in Algorithm 1 until converged.

```
Algorithm 1: GAN/HMM Harmonization
Input: Real phoneme sequences $P^{real}$, Speech utterances,
       initial phoneme segmentation boundaries $b$
Output: Unsupervised ASR system
while not converged do
    Given $b$, in an unsupervised way train the GAN;
    Obtain transcriptions $T$ of speech utterances using the
      generator within the GAN;
    Given $T$, train the HMMs;
    Obtain a new $b$ by forced alignment with the HMMs.
```

3. Experimental Setup

3.1. Dataset

The TIMIT corpus [28] was used in the preliminary experiments, which included recordings of phonetically-balanced read speech, with 6300 utterances from 630 speakers, with 4620 utterances for training and 1680 for testing. Each utterance included manually aligned phonetic/word transcriptions, as well as a 16-bit, 16kHz waveform file. 39-dim MFCCs were extracted with utterance-wise cepstral mean and variance normalization (CMVN) applied. We selected 4000 utterances in original training set for training and others for validation. We further randomly removed 4% of the phonemes and duplicated 11% of the phonemes in the real phoneme sequences for the training set to generate augmented phoneme sequences to be used with real phoneme sequences in training the GAN. This is referred to as data augmentation below.

3.2. Experimental setting

All models were trained with stochastic gradient descent using a mini-batch size of 150 and Adam optimization. For the intra-segment loss in equation (3), we sampled 6 sets of (y_i, y_j) in each segment from each training example. The phoneme classifier in Fig. 1 was an one-layer DNN with 512 ReLU units with 48 output classes. The input feature was a concatenation of 11 windowed frames of MFCCs . The discriminator was a two-layer 1D CNN, first layer with 4 different kernel sizes: 3, 5, 7 and 9, each with 256 channels, while second layer with kernel size 3 and 1024 channels. The weights (λ, α) were $(0.5, 10)$. The learning rate for $\mathcal{G}$ and $\mathcal{D}$ were set to 0.001 and 0.002 respectively. Every GAN training iteration consisted of 3 $\mathcal{D}$ updates and a single $\mathcal{G}$ update.

WFST including an 9-gram phoneme language model was used as mentioned in Section 2.2, where a state represents a phoneme. For modeling state transition probability in the unsupervised setting, we set the self-loop probability to 0.95 and 0.05 to other phonemes for all the phoneme states. HMM (monophone and triphone) training followed the standard recipes of Kaldi [29]. Linear Discriminant Analysis(LDA) and Maximum Likelihood Linear Transform(MLLT) were applied to MFCCs for model training. The ratio of the acoustic to the language models were set to 1:20 and 1:1 for the phoneme classifier and HMM models, respectively. The evaluation metrics were phone error rate (PER) and frame error rate (FER) for 39 phoneme classes mapped from the 48 output classes of the classifier.

Table 1: *Comparison of different methods*

Approaches			Matched (all 4000)		Nonmatched (3000/1000)	
			FER	PER	FER	PER
(I) Supervised (labeled)						
(a) RNN Transducer [22]			-	17.7	-	-
(b) standard HMMs			-	21.5	-	-
(c) Phoneme classifier			27.0	28.9	-	-
(II) Unsupervised (with oracle boundaries)						
(d) Mapping relationship GAN [21]			40.5	40.2	43.6	43.4
(e) Segmental empirical-ODM [22]			33.3	32.5	40.0	40.1
(f) Proposed: GAN			27.6	28.5	32.7	34.3
(III) Completely unsupervised (no label at all)						
(g) Segmental empirical-ODM [22]			-	36.5	-	41.6
Proposed	iteration 1	(h) GAN	48.3	48.6	50.3	50.0
		(i) GAN/HMM	-	30.7	-	39.5
	iteration 2	(j) GAN	41.0	41.0	44.3	44.3
		(k) GAN/HMM	-	27.0	-	35.5
	iteration 3	(l) GAN	39.7	38.4	45.0	44.2
		(m) GAN/HMM	-	26.1	-	33.1

4. Experimental Results

The first set of results are listed in Table 1. The upper section (I) of the table is for the case that all the 4000 training utterances are well labeled. The middle section (II) is the case that the oracle boundaries provided by TIMIT were used but nothing else. The lower section (III) is the case that the initial boundaries were obtained automatically with GAS [30]. The phoneme sequences of the training set were used in two different ways: In the left column of the table labeled ”Matched”, the phoneme transcriptions in all the 4000 training utterances were used as the real phoneme sequences in GAN training, which means the utterances and the real phoneme sequences are matched but not aligned during training. In the right column labeled ”Nonmatched”, 3000 utterances were taken as acoustic features while the phoneme transcriptions of the other 1000 utterances taken as the real phoneme sequences, no overlap between the two. All HMMs in this table had the same setting, triphone models with LDA+MLLT features.

4.1. Supervised baselines

In section (I) for supervised approaches with completely unlabeled data, in row (a) the RNN Transducer [31] was very powerful, while the standard triphone HMMs in row (b) were very strong too. The phoneme classifier in row (c) was exactly the phoneme classifier in the generator of Fig.1, except trained with annotated transcriptions.

4.2. Unsupervised but with oracle boundaries

With the oracle phoneme boundaries provided by TIMIT, rows (d) (e) in the middle section (II) are for two previously reported baselines, while row (f) for exactly the proposed GAN in Fig.1 except without the harmonized HMMs and data augmentation process mentioned in section 3.1 achieved significantly lower PER and FER. Interestingly, the PER in matched case achieved (28.5%)

Table 2: *Ablation studies for completely unsupervised model after 1 iteration of GAN/HMM harmonization (row (i) of Table 1) in Nonmatched case.*

Completely unsupervised (iteration 1, Nonmatched)		FER	PER
(1)	GAN/HMM (tri+LDA+MLLT) (row (i) of Table 1)	-	39.5
(2)	GAN/HMM (tri)	-	41.9
(3)	GAN/HMM (mono)	-	43.8
(4)	GAN (row (h) of Table 1)	50.3	50.0
(5)	GAN - Augm	53.6	51.9
(6)	GAN - Augm - $\mathcal{L}_{intra}$	63.0	62.6
(7)	GAN in (4) with RNN	75.5	71.6

in row (f) was even better than the supervised phoneme classifier (28.9%) in row (c) trained with labeled data, indicating the power of GAN. The discriminator of GAN in row (f) considered the whole generated phoneme sequences, while the DNN in row (c) considered only the input acoustic features. We also see the PER gap between matched and nonmatched cases for the proposed GAN in row (f) (5.8%) is smaller than the prior work of Segmental Empirical-ODM in row (e) (7.6%). The PER (34.3%) in nonmatched case by the proposed GAN was also close to the matched case of the prior work of Segmental Empirical-ODM in row (e) (32.5%).

4.3. Completely unsupervised

In the lowest section (III) of Table 1, only the first row (g) is for the prior work of [22], while all other rows (h)-(m) are for the approaches proposed here, respectively with 1, 2, and 3 iterations of harmonized GAN/HMM, rows (h) (j) (l) for GAN alone not further harmonized with HMMs, while rows (i) (k) (m) further harmonized with HMMs. All these approaches were based on the initial segmentation boundaries automatically generated by GAS [30].

We see the performance was consistently improved significantly after each iteration for either GAN alone (rows (h) (j) (l)) or GAN/HMM (rows (i) (k) (m)), or after harmonization with HMMs at each iteration (rows (i) v.s. (h), (k) v.s. (j), (m) v.s. (l)), for both matched and nonmatched cases, for both PER and FER. Not to mention the prior work of Segmental Empirical-ODM in row (g) needed a large batch-size (up to 20000 training examples in a batch) to achieve a satisfactory performance, while the training process here was done with a batch size as small as 150.

Note that the PER for GAN/HMM after iteration 2 (row (k)) in the matched case (27.0%) was even lower than all results with oracle boundaries (rows (d) (e) (f)) and supervised phoneme classifier with labeled data (row (c)). The GAN/HMM harmonization algorithm converged after three iterations in the preliminary experiments, ended up with PER of 26.1% and 33.1% in matched and nonmatched cases respectively, which were in fact 10.4% and 8.5% lower than the prior work in row (g), although still far behind the strong supervised baselines in rows (a) (b). All these verified the power of the proposed harmonized GAN/HMM approach.

4.4. Ablation studies

Ablation studies for the completely unsupervised harmonized GAN/HMM model obtained after iteration 1 in row (i) of Table 1 in the nonmatched case, initiated with the segmentation boundaries obtained by GAS, are reported in Table 2.

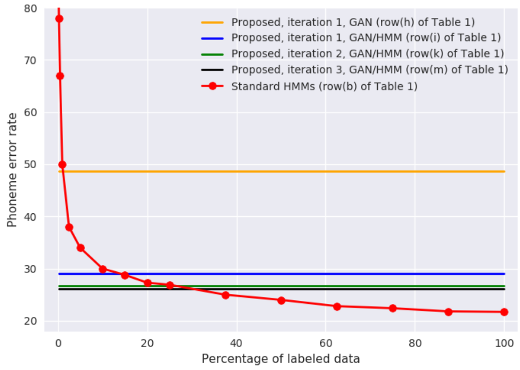

Figure 2: *Comparison of the proposed approaches to standard supervised HMMs with varying quantity of labeled data.*

Row (1) in Table 2 corresponds to row (i) in Table 1, GAN/HMM at iteration 1, in which the HMMs were triphones with LDA and MLLT. Exactly the same except without LDA and MLLT is in row (2), and triphones further replaced by monophones in row (3). GAN alone without HMM harmonization (exactly row (h) of Table 1) is in row (4), from which the data augmentation process mentioned in section 3.1 was removed is in row (5), while the loss $\mathcal{L}_{intra}$ in section 2.2.2 is further removed in row (6). Here we see the performance degraded step by step. For the HMMs used here, triphones are obviously better than mono-phones, and LDA and MLLT helped. For the GAN both data augmentation and the intra-segment loss made contributions. When we replaced the DNN used in the phoneme classifier in the GAN in row (4) here or row (h) in Table 1 by a RNN, the results is in row (7), which showed RNN did not work here, probably because the long term dependency captured by RNN was able to ”fool” the discriminator while generating output unrelated to the input.

4.5. Comparison with supervised approach

Here we wish to find out how the performance of the proposed approach compare to the standard supervised method when less labeled data are available. This is shown in Fig.2, in which the red curve is for the standard HMMs in row (b) of Table 1, whose right lower end is 21.5% as in Table 1. Here we show how PER goes up when only a percentage of the labeled training data (4000 utterances) are available. The horizontal lines then correspond to the proposed approach GAN/HMM at iteration 3, 2, 1 and GAN alone at iteration 1, or rows (m) (k) (i) (h) of Table 1, all in the matched case. We see the proposed approaches achieved the PER of standard HMMs when roughly 30%, 25%, 15% and 1% of the labeled data are available. This also demonstrates how the harmonized GAN/HMM proposed here offered improved performance step by step.

5. Conclusions

In this work we proposed a framework to achieve unsupervised phoneme recognition without any labeled data. A GAN is used in which a generator and a discriminator learn form each other iteratively, and a set of HMMs is further harmonized iteratively with the GAN. Dramatically improved performance was obtained compared to the previously reported results.

6. References

[1] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, “Attention-based models for speech recognition,” in *Advances in neural information processing systems*, 2015, pp. 577–585.

[2] 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 *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[3] N. Dehak, R. Dehak, P. Kenny, N. Brümmer, P. Ouellet, and P. Dumouchel, “Support vector machines versus fast scoring in the low-dimensional total variability space for speaker verification,” in *Tenth Annual conference of the international speech communication association*, 2009.

[4] H.-y. Lee and L.-s. Lee, “Enhanced spoken term detection using support vector machines and weighted pseudo examples,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 6, pp. 1272–1284, 2013.

[5] I.-F. Chen and C.-H. Lee, “A hybrid hmm/dnn approach to keyword spotting of short words.” in *INTERSPEECH*, 2013, pp. 1574–1578.

[6] A. Norouzian, A. Jansen, R. C. Rose, and S. Thomas, “Exploiting discriminative point process models for spoken term detection,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[7] K. Levin, K. Henry, A. Jansen, and K. Livescu, “Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings,” in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 410–415.

[8] K. Levin, A. Jansen, and B. Van Durme, “Segmental acoustic indexing for zero resource keyword search,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 5828–5832.

[9] H. Kamper, W. Wang, and K. Livescu, “Deep convolutional acoustic word embeddings using word-pair side information,” in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 4950–4954.

[10] K. Cho, B. van Merriënboer, Ç. Gülçehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “Learning phrase representations using rnn encoder–decoder for statistical machine translation,” in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: http://www.aclweb.org/anthology/D14-1179

[11] 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.

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

[13] Y.-A. Chung and J. Glass, “Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech,” *Proc. Interspeech 2018*, pp. 811–815, 2018.

[14] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, “Unsupervised neural machine translation,” 2018.

[15] A. Conneau, G. Lample, M. Ranzato, L. Denoyer, and H. Jégou, “Word translation without parallel data,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[16] G. Lample, L. Denoyer, and M. Ranzato, “Unsupervised machine translation using monolingual corpora only,” in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2018.

[17] 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.

[18] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, “Unsupervised cross-modal alignment of speech and text embedding spaces,” in *Advances in Neural Information Processing Systems*, 2018, pp. 7365–7375.

[19] Y.-C. Chen, S.-F. Huang, C.-H. Shen, H.-y. Lee, and L.-s. Lee, “Phonetic-and-semantic embedding of spoken words with applications in spoken content retrieval,” 2018.

[20] Y.-C. Chen, C.-H. Shen, S.-F. Huang, H.-y. Lee, and L.-s. Lee, “Almost-unsupervised speech recognition with close-to-zero resource based on phonetic structures learned from very small unpaired speech and text data,” *arXiv preprint arXiv:1810.12566*, 2018.

[21] D.-R. Liu, K.-Y. Chen, H.-y. Lee, and L.-s. Lee, “Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings,” *Proc. Interspeech 2018*, pp. 3748–3752, 2018.

[22] C.-K. Yeh, J. Chen, C. Yu, and D. Yu, “Unsupervised speech recognition via segmental empirical output distribution matching,” in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=Bylmkh05KX

[23] Y. Liu, J. Chen, and L. Deng, “Unsupervised sequence classification using sequential output statistics,” in *Advances in Neural Information Processing Systems*, 2017, pp. 3550–3559.

[24] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. International Convention Centre, Sydney, Australia: PMLR, 06–11 Aug 2017, pp. 214–223. [Online]. Available: http://proceedings.mlr.press/v70/arjovsky17a.html

[25] L. Yu, W. Zhang, J. Wang, and Y. Yu, “Seqgan: Sequence generative adversarial nets with policy gradient.” 2017.

[26] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on audio, speech, and language processing*, vol. 20, no. 1, pp. 30–42, 2012.

[27] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 5767–5777. [Online]. Available: http://papers.nips.cc/paper/7159-improved-training-of-wasserstein-gans.pdf

[28] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, “Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1,” *NASA STI/Recon technical report n*, vol. 93, 1993.

[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,” IEEE Signal Processing Society, Tech. Rep., 2011.

[30] Y.-H. Wang, C.-T. Chung, and H.-Y. Lee, “Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries,” *Proc. Interspeech 2017*, pp. 3822–3826, 2017.

[31] A. Graves, “Sequence transduction with recurrent neural networks,” *arXiv preprint arXiv:1211.3711*, 2012.