Interspeech 2018
2-6 September 2018, Hyderabad

Semi-Supervised End-to-End Speech Recognition

Shigeki Karita[1], *Shinji Watanabe*[2], *Tomoharu Iwata*[1], *Atsunori Ogawa*[1], *Marc Delcroix*[1]

[1]NTT Communication Science Laboratories
[2]Center for Language and Speech Processing, Johns Hopkins University

{karita.shigeki, iwata.tomoharu, ogawa.atsunori, marc.delcroix}@lab.ntt.co.jp,
shinjiw@jhu.edu

Abstract

We propose a novel semi-supervised method for end-to-end automatic speech recognition (ASR). It can exploit large unpaired speech and text datasets, which require much less human effort to create paired speech-to-text datasets. Our semi-supervised method targets the extraction of an intermediate representation between speech and text data using a shared encoder network. Autoencoding of text data with this shared encoder improves the feature extraction of text data as well as that of speech data when the intermediate representations of speech and text are similar to each other as an inter-domain feature. In other words, by combining speech-to-text and text-to-text mappings through the shared network, we can improve speech-to-text mapping by learning to reconstruct the unpaired text data in a semi-supervised end-to-end manner. We investigate how to design suitable inter-domain loss, which minimizes the dissimilarity between the encoded speech and text sequences, which originally belong to quite different domains. The experimental results we obtained with our proposed semi-supervised training shows a larger character error rate reduction from 15.8% to 14.4% than a conventional language model integration on the Wall Street Journal dataset.

Index Terms: speech recognition, semi-supervised learning, adversarial training, encoder-decoder

1. Introduction

End-to-end automatic speech recognition (ASR) systems learn speech-to-text mapping directly, where the speech feature is often a sequence of log Mel filterbank and the text is a sequence of character ids [1]. Those systems have advanced many aspects of ASR. For example, they do not need hand-crafted lexicons or complex weighted finite state transducer (WFST)-based decoders [2], which are used in the conventional ASR systems [3]. Actually [4] reports that their end-to-end ASR systems can outperform a single human transcriber when the training dataset is sufficiently large.

However, there still remains a major issue, which is the preparation of a supervised dataset, namely a speech-to-text corpora, because this requires a huge amount of human effort. We call such a dataset as a "paired" dataset because it contains pairs of speech and the corresponding text that is transcribed from the speech by human. According to [5], careful transcription costs 20 hours of human effort to create paired text for each hour of speech. To reduce the need for such hard effort, many researchers have developed semi-supervised training methods for ASR systems [6]–[10] because this way we can easily obtain a lot of unpaired data without such effort.

In this paper, we work on a new semi-supervised training method for end-to-end ASR systems that can improve performance by learning from unpaired data. The limitation of previous semi-supervised methods (e.g., restricted Boltzmann machine [6], [11] and language model integration [7], [12]) is their inability to learn unpaired speech and text simultaneously. Moreover, the previous methods require different aspects of human effort to build systems since they are not for end-to-end systems but employ a large ensemble of multiple acoustic and language models, and require careful tuning of their models. On the other hand, our method simply exploit both unpaired speech and text data in a fully data-driven and end-to-end manner.

Our basic framework is inspired by the recent image-to-image or text-to-text translation methods [13]–[15] that use a shared encoder network or a multi-task loss for unsupervised training. We propose a new shared encoder architecture for speech and text inputs that can encode speech and text from their different domains or modalities into a common intermediate representation. The encoded features are then proceeded by a decoder network with an attention mechanism [16] to predict text in a similar way as encoder-decoder based ASR systems. We refer to the common intermediate representation as an "inter-domain feature". To obtain and exploit such a feature, in the training of the encoder-decoder network, we combine three loss functions: 1. speech-to-text supervised loss on a small paired dataset that measures the negative log likelihood of ground-truth transcription text among the predictions from the paired speech; 2. text-to-text autoencoder [17] loss on a large unpaired dataset that measures the negative log likelihood of the input text; 3. inter-domain loss, which is the dissimilarity between the encoded features of speech and text on a large unpaired dataset. Here, we expect that the joint minimization of these losses will improve the speech-to-text transformation by considering text-to-text transformation via the inter-domain feature. In other words, the autoencoding of text may not only improve the feature extraction of text data but also that of speech data when we ensure the inter-domain feature to be similar to each other by training the shared encoder with inter-domain loss. Because of these unsupervised and supervised components, this framework performs semi-supervised learning.

The most important parts in this framework are the shared encoder and inter-domain loss. As the speech and text have quite different lengths and values, our shared encoder has subsampling layers to shorten the length of speech input to a length similar to that of text, and an embedding layer to convert character ids of text input to continuous real vectors like speech. In addition, we explore two different inter-domain losses that can train the inter-domain feature extractor by learning from large unpaired speech and text simultaneously. We compare the conventional inter-domain loss using generative adversarial network (GAN) that has been employed for text-to-text translation [15] and a newly proposed loss using Gaussian Kullback-Leibler (KL) divergence that measures dissimilarity between two distributions of real vectors (i.e., encoded speech and text).

10.21437/Interspeech.2018-1746

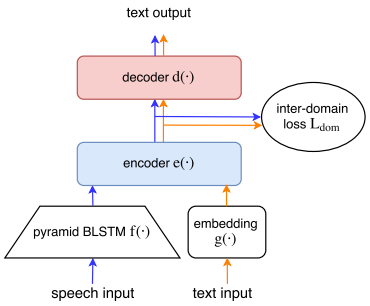

Figure 1: *Network architecture used in this research.*

We summarize our contributions below:

- We propose a novel end-to-end ASR method that efficiently exploits both speech and text features in a semi-supervised manner.
- Our proposed Gaussian KL divergence based inter-domain loss is simpler and shows better performance than the adversarial loss proposed in [13], [15].
- Our system reduced a character error rate (CER) from 15.8% to 14.4% with the small paired and the large unpaired data in the Wall Street Journal (WSJ) dataset.

2. Semi-supervised framework

In this section, we introduce our definitions of semi-supervised end-to-end speech recognition using an encoder-decoder.

2.1. Baseline encoder-decoder model

An encoder-decoder model consists of two parts called "encoder" and "decoder" networks [18]. The encoder consists of a pyramid bidirectional long short-term memory (BLSTM) $\mathrm{f}(\cdot)$ [19] and conventional BLSTM $\mathrm{e}(\cdot)$. The encoder receives an utterance consisting of a sequence of log Mel filterbank features $\mathbf{x} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_u]$, where u is the number of frames, and transforms it to an intermediate representation $\mathbf{e} = [\boldsymbol{e}_1, \boldsymbol{e}_2, \ldots, \boldsymbol{e}_{u'}]$, where u' is the number of sub-sampled frames. Then, the decoder network $\mathrm{d}(\cdot)$ predicts a current token y_t in a character vocabulary set $Y = \{\text{'a'}, \text{'b'}, \ldots, \text{<EOS>}\}$ using the encoder's output $\mathbf{e}$, the decoder's state vector $\boldsymbol{s}_t$ and the embedded vector of the previous token y_{t-1}. We describe this processing pipeline as follows:

$$\mathbf{e} = \mathrm{e}(\mathrm{f}(\mathbf{x})), \tag{1}$$

$$y_0 = \text{<SOS>}, \tag{2}$$

$$[\Pr(y_t|y_{t-1}, \mathbf{e}), \boldsymbol{h}_t] = \mathrm{d}(y_{t-1}, \boldsymbol{h}_{t-1}, \mathbf{e}), \tag{3}$$

where <SOS> is the start of a sequence token, and the initial state $\boldsymbol{h}_0$ is zero. For simplicity, in the remainder of this paper, we omit the states $\boldsymbol{h}_t$ and write Eqs. (1)–(3) as a sequence form

$$\Pr(\mathbf{y}|\mathbf{e}) = \prod_{t=1}^{|\mathbf{y}|} \Pr(y_t|y_{t-1}, \mathbf{e}) = d(\mathbf{e}), \tag{4}$$

where $\mathbf{y} = [y_1, y_2, \ldots, y_{|\mathbf{y}|}]$ is a predicted text, and $|\mathbf{y}|$ denotes the length of $\mathbf{y}$. The detailed definitions of the encoder and decoder are similar to the model proposed in [20] (i.e., the encoder consists of six pyramid and conventional BLSTMs, and the decoder consists of one LSTM and a location based attention mechanism). Note that, the decoder emits an <EOS> token when it predicts the end of a sequence.

Algorithm 1 Semi-supervised training algorithm.

unpaired speech and text datasets: $\mathcal{S}, \mathcal{T}$,
paired speech-text dataset: $\mathcal{Z}$.
hyper-parameters: $\alpha \in [0.5, 0.9]$, $\beta \in [0.5, 0.9]$, $\epsilon = 10^{-8}$.
initial parameters of encoder e, decoder d, and discriminator h: Φ, Θ, Ξ.
procedure SEMISUPERVISEDTRAIN(Φ, Θ, Ξ)
  for $i = 1, 2, \ldots, \max(\#\mathcal{S}, \#\mathcal{T}, \#\mathcal{Z})$ **do**
    $S_i \sim \mathcal{S}, T_i \sim \mathcal{T}$, ▷ sample unpaired minibatches
    $Z_i \sim \mathcal{Z}$ ▷ sample one paired minibatch
    if use KL **then**
      $\mathrm{L}_{\mathrm{dom}}(S_i, T_i) = \mathrm{KL}_{\mathbf{x}\sim S_i, \mathbf{y}\sim T_i}(\mathrm{e}(\mathrm{f}(\mathbf{x}))||\mathrm{e}(\mathrm{g}(\mathbf{y})))$
    else if use GAN **then**
      $\mathrm{L}_{\mathrm{dom}}(S_i, T_i, \Xi) = \sum_{\mathbf{x}\in S_i} \log \mathrm{h}(\mathrm{e}(\mathrm{f}(\mathbf{x})))$
        $+ \sum_{\mathbf{y}\in T_i} \log(1 - \mathrm{h}(\mathrm{e}(\mathrm{g}(\mathbf{y}))))$
      $\xi \leftarrow \mathrm{Adadelta}_\epsilon(\xi, \frac{\partial \mathrm{L}_{\mathrm{dom}}}{\partial \xi}), \xi \in \Xi$ ▷ update discriminator
    else
      $\mathrm{L}_{\mathrm{dom}} = 0$
    end if
    $\mathrm{L}_{\mathrm{text}}(S_i, T_i, \Theta, \Phi) = -\sum_{\mathbf{y}\in T_i} \log \Pr(\mathbf{y}|\mathrm{e}(\mathrm{g}(\mathbf{y})))$
    $\mathrm{L}_{\mathrm{pair}}(Z_i, \Theta, \Phi) = -\sum_{(\mathbf{x}', \mathbf{y}')\in Z_i} \log \Pr(\mathbf{y}'|\mathrm{e}(\mathrm{f}(\mathbf{x}')))$
    $\mathrm{L} = \alpha \mathrm{L}_{\mathrm{pair}} + (1-\alpha)(\beta \mathrm{L}_{\mathrm{dom}} + (1-\beta)\mathrm{L}_{\mathrm{text}})$
    $\theta \leftarrow \mathrm{Adadelta}_\epsilon(\theta, \frac{\partial \mathrm{L}}{\partial \theta}), \theta \in \Theta$ ▷ update encoder
    $\phi \leftarrow \mathrm{Adadelta}_\epsilon(\phi, \frac{\partial \mathrm{L}}{\partial \phi}), \phi \in \Phi$ ▷ update decoder
  end for
end procedure

2.2. Shared encoder-decoder model for speech and text

We extend the encoder decoder network to allow not only speech but also text input as illustrated in Figure 1. This network is a variant of the network proposed in the unsupervised text translation approach described in [14], which consists of one shared encoder and two decoders. The shared encoder $\mathrm{e}(\cdot)$ aims to extract the common inter-domain feature between speech and text for learning unpaired speech and text simultaneously.

Compared to [14], we use only a target-side (i.e. text-side) decoder and drop the source-side (i.e., speech-side) decider because our focus is not text-to-speech in this paper. In other words, we simply combine speech-to-text encoder-decoder and text-to-text autoencoder by sharing their layers. In addition, the input layers of the encoder are different from [15] as speech and text are different types of data. For example, speech is a sequence of continuous vectors while text is a sequence of discrete symbols. Moreover, the length of the speech is longer than the length of the text in ASR. Hence, we use a embedding layer $\mathrm{g}(\cdot)$ for the text input to convert discrete ids of characters $\mathbf{y}$ into a continuous representation $\mathrm{g}(\mathbf{y})$ while we use the pyramid BLSTM $\mathrm{f}(\cdot)$ [19] for the speech input to shorten the length as described in the previous subsection.

We expect that such a shared model will enable the semi-supervised learning, i.e., autoencoding of text data may not only enhance a intermediate representation of text data but also that of speech data when we regularize those representations to be similar to each other during the semi-supervised training.

2.3. Semi-supervised training

We detail the training pipeline of our semi-supervised framework in Algorithm 1. First, we sample one minibatch each from unpaired speech dataset $\mathcal{S}$, unpaired text dataset $\mathcal{T}$, and paired speech-to-text dataset $\mathcal{Z}$. Then, we compute a training loss for semi-supervised training that consists of the following three terms:

1. For paired speech and text data $(\mathbf{x}', \mathbf{y}') \in \mathcal{Z}$, we use the conventional speech-to-text loss $\mathrm{L}_{\mathrm{pair}}$, which consists of a negative log likelihood of the ground-truth text $\mathbf{y}'$ given by the encoded speech $\mathrm{e}(\mathrm{f}(\mathbf{x}'))$ as follows:

$$\mathrm{L}_{\mathrm{pair}} = -\sum_{(\mathbf{x}', \mathbf{y}') \in \mathcal{Z}} \log \Pr(\mathbf{y}' | \mathrm{e}(\mathrm{f}(\mathbf{x}'))); \quad (5)$$

2. $\mathrm{L}_{\mathrm{text}}$ is a negative log-likelihood that the encoder-decoder network can reconstruct text from unpaired text dataset $\mathcal{T}$ like an autoencoder [17] as follows:

$$\mathrm{L}_{\mathrm{text}} = -\sum_{\mathbf{y} \in \mathcal{T}} \log \Pr(\mathbf{y} | \mathrm{e}(\mathrm{g}(\mathbf{y}))); \quad (6)$$

   Note that, it is the same loss for Eq. (5) except that the input of decoder is the encoded text $\mathrm{e}(\mathrm{g}(\mathbf{y}))$ here.

3. $\mathrm{L}_{\mathrm{dom}}$ is the dissimilarity between distributions of the encoded speech feature $\mathrm{e}(\mathrm{f}(\mathbf{x})), \mathbf{x} \in \mathcal{S}$ and the encoded text feature $\mathrm{e}(\mathrm{g}(\mathbf{y})), \mathbf{y} \in \mathcal{T}$. We refer to this loss as an "inter-domain loss". We discuss its in detail in the next section.

This multi-task loss aims to learn both the speech-to-text mapping and the inter-domain feature extraction between speech and text. According to [13], [15], the minimization of the inter-domain loss and the reconstruction loss can improve source-to-target mapping (here speech-to-text mapping) for unsupervised training, i.e., the minimization of the text reconstruction error $\mathrm{L}_{\mathrm{text}}$ can improve the text decoding from the encoded speech features as well as from the encoded text features when we also minimize the inter-domain loss as a dissimilarity between those intermediate representations during the training. Note that, the inter-domain loss for speech-to-text plays a more difficult role than that in image-to-image or text-to-text translation systems in [13], [15] because the source and target domains are quite different. Therefore, we introduce the third loss $\mathrm{L}_{\mathrm{pair}}$ to ensure speech-to-text mapping within a smaller paired dataset.

3. Inter-domain loss

In this section, we explore suitable definitions of the inter-domain loss for the semi-supervised training algorithm.

3.1. Adversarial Loss

First, we adopt an adversarial loss for the inter-domain loss between speech and text because it has the ability to implicitly model any unknown distributions through the data-driven training. This method is also known as a "generative adversarial network" (GAN) [21]. The loss function requires generative and discriminator networks to learn the dissimilarity. In our case, the generative network is the encoder network $\mathrm{e}(\cdot)$. Additionally, we introduce the discriminator network $\mathrm{h}(\cdot)$ that predicts the speech domain likelihood $\Pr(\mathbf{x} \in \mathcal{S}) = \mathrm{h}(\mathrm{e}(\mathbf{x}))$ or the text domain likelihood $\Pr(\mathbf{x} \in \mathcal{T}) = 1 - \Pr(\mathbf{x} \in \mathcal{S})$ from encoded features for the adversarial training. We define the loss function using those networks as follows:

$$\mathrm{L}_{\mathrm{GAN}} = \sum_{\mathbf{x} \in S_i} \log \mathrm{h}(\mathrm{e}(\mathrm{f}(\mathbf{x}))) + \sum_{\mathbf{y} \in T_i} \log(1 - \mathrm{h}(\mathrm{e}(\mathrm{g}(\mathbf{y})))). \quad (7)$$

Here the discriminator $\mathrm{h}(\cdot)$ is trained to discriminate whether the input is encoded speech $\mathrm{e}(\mathrm{f}(\mathbf{x}))$ or encoded text $\mathrm{e}(\mathrm{g}(\mathbf{y}))$, i.e., to maximize this loss. On the other hand, the generator $\mathrm{e}(\cdot)$ is trained to "fool" the discriminator, i.e., to minimize this loss. This min-max training aims to implicitly model the dissimilarity and the common distribution of encoded speech and text. Although GAN provides the flexibility as regards the dissimilarity modeling, its training is difficult because there is a conflict between the optimization of the generator $\mathrm{e}(\cdot)$ and the discriminator $\mathrm{h}(\cdot)$. Moreover, we need to tune some configurations of the discriminator network $\mathrm{h}(\cdot)$.

3.2. Gaussian KL-divergence

Here we propose an alternative to the adversarial loss that does not suffer from the tuning issue. For simplicity, we assume that inter-domain features of speech and text follow a multivariate Gaussian distribution. To ensure that speech and text are mapped to the common Gaussian distribution, we impose that two Gaussian distributions of encoded speech and text features become close to each other. We achieve this property by minimizing the KL divergence between the Gaussian distributions. Let P and Q be the z-dimensional Gaussian distributions of encoded speech and encoded text, respectively as follows:

$$P = \mathrm{Normal}(\boldsymbol{\mu}_P, \boldsymbol{\Sigma}_P), \quad (8)$$

$$Q = \mathrm{Normal}(\boldsymbol{\mu}_Q, \boldsymbol{\Sigma}_Q), \quad (9)$$

where the statistics of Gaussian distributions are $\boldsymbol{\mu}_P = \mathrm{E}_{\mathbf{x} \sim \mathcal{S}}\left[\mathrm{e}(\mathrm{f}(\mathbf{x}))\right], \boldsymbol{\Sigma}_P = \mathrm{Cov}_{\mathbf{x} \in \mathcal{S}}\left[\mathrm{e}(\mathrm{f}(\mathbf{x}))\right]$, $\boldsymbol{\mu}_Q = \mathrm{E}_{\mathbf{y} \sim \mathcal{T}}\left[\mathrm{e}(\mathrm{g}(\mathbf{y}))\right]$, and $\boldsymbol{\Sigma}_P = \mathrm{Cov}_{\mathbf{y} \sim \mathcal{T}}\left[\mathrm{e}(\mathrm{g}(\mathbf{y}))\right]$. The KL divergence between these distributions is defined as

$$\mathrm{L}_{\mathrm{KL}} = 0.5 \left[\log \frac{\det \boldsymbol{\Sigma}_P}{\det \boldsymbol{\Sigma}_Q} + \mathrm{tr} \boldsymbol{\Sigma}_Q^{-1} \boldsymbol{\Sigma}_P + \boldsymbol{a}^\top \boldsymbol{\Sigma}_Q^{-1} \boldsymbol{a} - z \right], \quad (10)$$

where $\boldsymbol{a} = \boldsymbol{\mu}_P - \boldsymbol{\mu}_Q$ [22]. We compute the gradients of this loss w.r.t. the encoder's parameter with the standard back-propagation. To stabilize the training, we implement the determinant and its gradient in log-domain that results in much better convergence. In experiments, training with this Gaussian KL divergence is faster, more stable, and memory efficient than the adversarial loss $\mathrm{L}_{\mathrm{GAN}}$ because there is no additional NN.

4. Related work

There are two well-known techniques for the semi-supervised learning of ASR. One is a restricted Boltzmann machine (RBM) [23]. The RBM enables pre-training with the speech data alone before training the feed-forward neural networks (NN) based acoustic models with the paired data [3]. The problem with this method is that it is inapplicable to recent NN variants such as the recurrent NN used in end-to-end speech recognition systems [24].

The other approach is language modeling with text data [12], [25]. This technique learns the text likelihood related to the grammar or spelling from text dataset without human efforts for transcribing the speech. In ASR, the "decoding" procedure exploits both the supervised speech-to-text model and unsupervised language model scores to search for the maximum likelihood text. Obviously, the limitation of this language modeling is the lack of an ability to learn the speech features in an unsupervised manner.

To realize joint training on an unpaired dataset of speech and text, we follow the recent advances in image [13] and text translation [14], [15] that propose unsupervised training for autoencoder and encoder-decoder models. However, we could not

obtain reasonable ASR results similar to those for images and texts in our initial experiments. The reason for this difficulty was the clear difference between source and target domains in ASR (speech-to-text) unlike in image-to-image and text-to-text translation. Therefore, we decided to relax the ASR problem from unsupervised learning to semi-supervised learning.

5. Experiments

5.1. Settings

The WSJ dataset is suitable for use in examining the semi-supervised ASR because it has a small 15-hour (7138 utterances) dataset "si84" and a large 81-hour (37416 utterances) dataset "si284" as its official training datasets. We used si84 as a paired dataset and shuffled two independent half sets of si284 as unpaired speech and text dataset, respectively. In addition, we used the official test dataset "dev93" for a hyper-parameter and decoding parameter search and "eval92" for performance evaluation. First, we trained the basic network defined in Section 2.1 with paired si84. Then, we retrained the model with additional embedding layer $\mathrm{g}(\cdot)$ using the semi-supervised loss with paired si84 and unpaired si284 with Algorithm 1. As seen in [8]–[10], we observed that the retraining always results better than training from random weights. We searched the best hyper-parameters $\alpha, \beta \in [0.5, 0.9]$ on the dev93 set. As discussed in Section 4, we also adopted a recurrent NN language model (RNNLM) [25] as a conventional semi-supervised method for comparison. We follow all the settings for the large dataset si284 in the early version of ESPnet [26], which is basically the same as [27] except for the small dataset si84 and the semi-supervised training that we undertook for this experiment. For training with $\mathrm{L}_{\mathrm{GAN}}$, we use a multilayer perceptron with three hidden layers of size 1024 and ReLU nonlinearity as the discriminator. Our entire source code will be available to make it possible to reproduce this experiment [1].

5.2. Character/word error rates

Table 1 summarizes the character/word error rates on the WSJ eval92. The baseline systems trained on the paired si284 reproduced almost the same results as in [20], [27] (CER7.4%) and [1] (CER6.4%, WER18.6%). Here, the CER/WER on the paired si84 and paired si284 were the upper (CER15.8%, WER44.2%) and lower bounds (CER6.3%, WER18.3%) for the semi-supervised training. We observed a clear improvement in CER/WER by using semi-supervised training with L_{KL} (CER14.4%, WER39.5%) and without the inter-domain loss while RNNLM and $\mathrm{L}_{\mathrm{GAN}}$ degrade the baseline model. By comparing $\mathrm{L}_{\mathrm{GAN}}$ and $\mathrm{L}_{\mathrm{GAN}}(si84)$, which are retrained on the unpaired si284 and the paired si84, respectively, we confirmed that the inter-domain loss minimization on large unpaired datasets could reduce the ASR errors.

To investigate the effects of these different inter-domain losses, we adjusted the ratio of the paired and unpaired training data using si84. Figure 2 shows the CER/WER for the several paired ratios. In all cases, L_{KL} results better for smaller amounts of paired data than $\mathrm{L}_{\mathrm{GAN}}$ that requires additional tuning of the discriminator.

However, the WER for si84 is still relatively high because of the very limited amount of training data. Indeed, end-to-end systems usually require a relatively large amount of data to achieve high performance [4], [20], [27]. In future work, we would like to confirm whether the promising relative improvement observed in this experiment remains for tasks with a larger amount of paired and unpaired data.

[1] https://github.com/ShigekiKarita/espnet-semi-supervised

Table 1: *Character error rates (%) on the WSJ "eval92".*

si84	si284	RNNLM	$\mathrm{L}_{\mathrm{text}}$	$\mathrm{L}_{\mathrm{dom}}$	CER
pair	-	-	-	-	15.8
pair	-	✓	-	-	16.6
pair	unpair	-	✓	-	15.6
pair	unpair	-	✓	$\mathrm{L}_{\mathrm{GAN}}$(si84)	15.0
pair	unpair	-	✓	$\mathrm{L}_{\mathrm{GAN}}$	17.9
pair	unpair	-	✓	L_{KL}(si84)	15.0
pair	unpair	-	✓	L_{KL}	**14.4**
-	pair	-	-	-	6.3
-	pair	✓	-	-	6.1

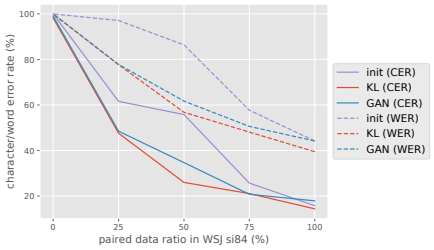

Figure 2: *Character/word error rates (%) on the WSJ eval92 for multiple paired data ratios on the WSJ si84.*

5.3. t-SNE visualization of inter-domain features

Figure 3 shows the common inter-domain features of speech and text encoded by the models retrained on unpaired si284 without (left) and with (right) the inter-domain-loss L_{KL}. As the shared encoder outputs z (=320) dimensional features, we apply dimensionality reduction to the two-dimensional plane using t-distributed stochastic neighbor embedding (t-SNE) [28]. These results support our expectation in Section 2.3 that our inter-domain loss regularizes the encoded features of speech and text so that the become similar because the points of retrained speech (blue) and text (orange) are more mixed when the inter-domain loss is combined.

6. Conclusions and future work

In this study, we introduced the first end-to-end semi-supervised method for ASR. It can exploit large unpaired speech and text data simultaneously unlike the conventional methods. Through experiments, our Gaussian KL-based inter-domain loss improved the CER from 15.8% to 14.4% by semi-supervised training on a 14-hour paired dataset and an 81-hour unpaired dataset of the WSJ. In future work, we plan to extend the proposed approach for fully unsupervised ASR.

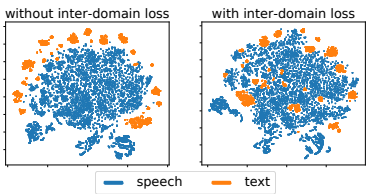

Figure 3: *t-SNE visualization of inter-domain features from a model retrained without (left) and with the* L_{KL} *inter-domain loss (right).*

7. References

[1] D. Bahdanau, J. Chorowski, D. Serdyuk, and Y. Bengio, "End-to-End Attention-based Large Vocabulary Speech Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 4945–4949, 2016.

[2] M. Mohri, F. Pereira, and M. Riley, "Speech Recognition with Weighted Finite-State Transducers," *Springer Handbook of Speech Processing*, pp. 559–584, 2008.

[3] G. Hinton, L. Deng, D. Yu, *et al.*, "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[4] D. Amodei, S. Ananthanarayanan, R. Anubhai, *et al.*, "Deep Speech 2 : End-to-End Speech Recognition in English and Mandarin," in *International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, vol. 48, 2016, pp. 173–182.

[5] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a Resource for the Next Generations of Speech-to-Text," *International Conference on Language Resources and Evaluation*, vol. 4, pp. 69–71, 2004.

[6] K. Vesely, M. Hannemann, and L. Burget, "Semi-Supervised Training of Deep Neural Networks," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 267–272.

[7] E. Dikici and M. Saraçlar, "Semi-supervised and Unsupervised Discriminative Language Model Training for Automatic Speech Recognition," *Speech Communication*, vol. 83, pp. 54–63, 2016.

[8] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep Neural Network Features and Semi-Supervised Training for Low Resource Speech Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 6704–6708.

[9] F. Grézl and M. Karafiát, "combination of multilingual and semi-supervised training for under-resourced languages."

[10] K. Veselý, L. Burget, and J. Černocký, "Semi-Supervised DNN Training with Word Selection for ASR," in *Interspeech 2017*, vol. 2017-Augus, 2017, pp. 3687–3691.

[11] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A Fast Learning Algorithm for Deep Belief Nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[12] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, "Recurrent Neural Network based Language Model," *Interspeech*, no. September, pp. 1045–1048, 2010.

[13] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised Image-to-Image Translation Networks," in *Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, *et al.*, Eds., 2017, pp. 700–708.

[14] M. Artetxe, G. Labaka, E. Agirre, and K. Cho, "Unsupervised Neural Machine Translation," *International Conference on Learning Representation*, 2018.

[15] G. Lample, L. Denoyer, and M. Ranzato, "Unsupervised Machine Translation Using Monolingual Corpora Only," *International Conference on Learning Representation*, 2018.

[16] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *International Conference on Learning Representations*, 2015.

[17] G. E. Hinton, "Reducing the Dimensionality of Data with Neural Networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[18] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to Sequence Learning with Neural Networks," *Neural Information Processing Systems*, pp. 3104–3112, 2014.

[19] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell: A neural network for large vocabulary conversational speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016.

[20] S. Kim, T. Hori, and S. Watanabe, "Joint CTC-Attention based End-to-End Speech Recognition Using Multi-Task Learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2017, pp. 4835–4839.

[21] I. Goodfellow, J. Pouget-Abadie, M. Mirza, *et al.*, "Generative Adversarial Nets," in *Neural Information Processing Systems*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680.

[22] S. Kullback, "Multivariate Analysis: Other Hypothesis," in *Information Theory and Statistics*, 1959, p. 298.

[23] A. Mohamed, G. Dahl, and G. Hinton, "Deep Belief Networks for phone recognition," *Neural Information Processing Systems Workshop on Deep Learning for Speech Recognition*, 2009.

[24] I. Sutskever, G. Hinton, and G. Taylor, "The Recurrent Temporal Restricted Boltzmann Machine," *Neural Information Processing Systems*, vol. 21, no. 1, pp. 1601–1608, 2008.

[25] T. Hori, S. Watanabe, Y. Zhang, and W. Chan, "Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM," in *Interspeech*, 2017, pp. 949–953.

[26] S. Watanabe, T. Hori, S. Karita, *et al.*, "ESPnet: End-to-End Speech Processing Toolkit," manuscript submitted for publication.

[27] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid CTC/Attention Architecture for End-to-End Speech Recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[28] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.