Domain Adversarial Self-Supervised Speech Representation Learning for Improving Unknown Domain Downstream Tasks

Tomohiro Tanaka, Ryo Masumura, Hiroshi Sato, Mana Ihori, Kohei Matsuura, Takanori Ashihara, Takafumi Moriya

NTT Corporation
tomohiro.tanaka.ht@hco.ntt.co.jp

Abstract
In this paper, we propose novel self-supervised speech representation learning method that obtains domain invariant representations by using a domain-adversarial neural network. Recently, self-supervised representation learning has been actively studied in the speech field. Since self-supervised learning requires large-scale unlabeled data, we need to effectively use data collected from a variety of domains. However, existing methods cannot construct valid representations in unknown domains because they cause overfitting to the domains in the training data. To solve this problem, our proposed method constructs contextual representations that cannot identify the domains from input speech by using domain-adversarial neural networks. The domain-adversarial training can improve robustness for data in unknown domains because the model trained by our proposed method can construct domain invariant representations. In addition, we investigate multi-task learning of representation construction and domain classification to consider domain information. Experimental results show that our proposed method outperforms the conventional training method of wav2vec 2.0 in unknown domain downstream automatic speech recognition tasks.

Index Terms: speech representation learning, self-supervised learning, automatic speech recognition, Transformer

1. Introduction
There has been increasing progress in end-to-end automatic speech recognition (ASR) that directly converts a speech into textual tokens. End-to-end ASR is an approach with simpler training and faster decoding compared with traditional deep-neural-network hidden-Markov-model hybrid ASR. Various end-to-end ASR methods including connectionist temporal classification (CTC) [1, 2], recurrent neural network (RNN) encoder-decoder models [3, 4], RNN transducers [5, 6], and Transformer encoder-decoder models [7, 8], have been investigated. These ASR models are generally trained from labeled speech data.

Self-supervised representation learning (SSL) is known as a method that uses unlabeled data (text, speech, image, etc.). A typical SSL model is trained from a large amount of unlabeled data. Then, the trained parameters are transferred into a model for specific tasks called downstream tasks. This procedure has been shown to be effective even when the amount of training data for downstream tasks is limited. SSL has been making great progress [9–11] in natural language processing. The most successful model is Bidirectional Encoder Representations from Transformers [9], which trains the bi-directional Transformer model from a large amount of unlabeled text data.

In speech processing, there have been several attempts to improve ASR performance on the basis of SSL that use unlabeled

speech data [12–18]. Transformer-based models for learning contextual representations have been proposed [14, 15]. To better handle continuous input, vector quantization was introduced to convert continuous speech into discrete values [17]. Furthermore, joint training of quantization of speech and constructing speech representations was proposed [18]. These models enable us to use unlabeled speech data for improving ASR performance by transferring the trained model parameters.

To construct a highly accurate SSL model, we need to use as much data as possible for the training. A large amount of text data of a single domain can be collected from the web, but it is difficult to do the same for speech. Thus, it is necessary to collect data from various domains (meetings, telephone calls, lectures, etc.) when using a large amount of speech data. In other words, it is important to effectively utilize data from various domains obtained from various services and corpora. However, existing methods cannot obtain valid representations in unknown domains because they do not consider domain information of input speech. Actually, it was shown that ASR performance of a certain domain does not improve depending on the domain of training data for self-supervised learning [19].

To address this problem, we propose self-supervised speech representation learning methods that reduces domain dependencies. By reducing the dependencies, the self-supervised speech representation model obtains robust representations for unknown domain data. Therefore, we introduce a domain-adversarial neural network [20] to the self-supervised speech representation learning. Our proposed method is achieved by extending wav2vec 2.0 [18]. The proposed method performs quantization of speech and acquisition of contextual representations, while simultaneously regularizing the representations so that they cannot identify the domain. We verify the effectiveness of our proposed models with Japanese ASR tasks with multiple domain datasets. The results show that our proposed model outperforms a conventional training method of wav2vec 2.0. In addition, we explore using a domain classifier for multi-task learning to compare with domain-adversarial speech representation learning.

2. Related Work

**wav2vec 2.0:** Our proposed method is closely related to wav2vec 2.0 [18], which is composed of three components: speech encoder, quantization module, and transformer encoder. During training, the model simultaneously learns vector quantization and acquires context representations from unlabeled speech. The conventional wav2vec 2.0 did not consider domain information if multiple domain datasets are used for training. Our proposed models explicitly take into account the domain difference in the training by using domain labels of training datasets.
Adversarial training: Adversarial training was firstly proposed in the computer vision fields [21]. This approach was implemented with gradient reversal layer (GRL) [20]. By making the model unable to identify domains, it is expected to obtain domain invariant representations. In ASR, training with GRL has been proposed and shown to be effective [22, 23]. Our proposal introduce GRL in self-supervised speech representation learning.

3. Domain-Adversarial Self-Supervised Speech Representation Learning

We use two types of datasets: unlabeled speech dataset \( \mathcal{D}_{\text{SSL}} \) used for self-supervised speech representation learning and labeled speech dataset \( \mathcal{D}_{\text{FT}} \) for finetuning of downstream ASR tasks. The unlabeled speech datasets consist of pairs of speech \( X \) and domain label \( d \) as \( \mathcal{D}_{\text{SSL}} = \{ (X_1, d_1), \ldots, (X_N, d_N) \} \), where \( N \) is the number of these pairs. The labeled speech dataset consists of pairs of speech \( X \) and token label sequence \( Y \) as \( \mathcal{D}_{\text{FT}} = \{ (X_1, Y_1), \ldots, (X_M, Y_M) \} \), where \( M \) is the number of these pairs.

3.1. Self-supervised learning with multiple domain datasets

3.1.1. Model components

**Speech encoder:** Samples of raw waveform speech \( X = \{ x_1, \ldots, x_I \} \) are converted into latent speech representations \( Z = \{ z_1, \ldots, z_T \} \), where \( I \) and \( T \) are the number of samples and frames of latent speech representations, respectively. The latent speech representations \( Z \) are calculated as

\[
Z = \text{Convolution}(X; \theta_{\text{ssl}}),
\]

where Convolution(\( \cdot \)) consists of several blocks containing convolutional networks followed by layer normalization and an activation function, \( \theta_{\text{ssl}} \) is the trainable parameter. The output of the speech encoder is input to both the quantization module and the Transformer.

**Transformer encoder:** Masking operation is applied to the latent speech representations of time steps, and a masked sequence is input to the \( M \) Transformer blocks. We define the output of the \( m \)-th Transformer encoder block as \( F^m = \{ f^m_1, \ldots, f^m_T \} \) and the input of the first block as \( F^0 = \text{Mask}(Z) \), where \( \text{Mask}(\cdot) \) is the masking operation. The computational process in the \( m \)-th block of the Transformer encoder is defined as

\[
F^m = \text{Transformer}(F^{m-1}; \theta_{\text{ssl}}),
\]

where \( \text{Transformer}(\cdot) \) is a Transformer module including a scaled dot-product multi-head self-attention layer and a position-wise feed-forward network and \( \theta_{\text{ssl}} \) is a trainable parameter. The context representations \( C = \{ c_1, \ldots, c_T \} \) is written by the final output of the Transformer encoder as \( C = F^M \).

**Quantization module:** Quantized representations are constructed from the latent speech representations. Given multiple codebooks which have entries, one entry is selected from each codebook and concatenates the vectors. The Gumbel softmax function [24] is used to select codebook entries in a fully differentiable manner. The quantized representations \( Q = \{ q_1, \ldots, q_T \} \) are calculated with the Gumbel softmax function and a linear projection as

\[
h = \text{GumbelSoftmax}(Z; \theta_{\text{quant}}),
\]

\[
Q = \text{FeedForward}(h; \theta_{\text{quant}}),
\]

where \( \text{GumbelSoftmax}(\cdot) \) represents the multiple Gumbel softmax operations with the Gumbel noise and temperature, \( \text{FeedForward}(\cdot) \) is the linear projection layer, and \( \theta_{\text{quant}} \) is the model parameters.

**Domain classifier:** The domain classifier identifies the domain of input speech from the context representation \( C \). The conditional probabilities of domains from the input speech \( X \) are calculated as

\[
o = \text{FeedForward} \left( \sum_{t=1}^T c_t; \theta_{\text{dec}} \right),
\]

\[
P(d|X) = \text{Softmax}(o; \theta_{\text{dec}}),
\]

where \( \text{Softmax}(\cdot) \) represents the softmax function with a activation function and linear transformation, \( d \) is the domain label, and \( \theta_{\text{dec}} \) is the trainable parameters.

3.1.2. Training

During the training, we optimize parameters with two losses.

**Contrastive loss:** Contrastive loss \( l \) for an utterance \( X \) is com-
computed from the context representation $C$ and quantized representations $Q$ as

$$l(X) = - \sum_{t=1}^{T} \log \frac{\exp(\text{sim}(c_t, q_t))}{\sum_{q \in Q} \exp(\text{sim}(c_t, q))},$$  \hspace{1cm} (7)$$

where $\hat{Q}$ is the set of negative samples of quantized representations, $\kappa$ is the temperature value, and $\text{sim}(\cdot, \cdot)$ calculates the cosine similarity $\text{sim}(c, q) = e^c q / ||c|| ||q||$. The contrastive loss $L_C$ for optimization is computed as

$$L_C = \sum_{X' \in \mathcal{D}_{SSL}} l(X').$$  \hspace{1cm} (8)$$

**Domain loss:** The domain loss is calculated from the logits in Eq. (6). The cross entropy loss is calculated as

$$L_D = - \sum_{(d', X') \in \mathcal{D}_{SSL}} \log P(d'|X').$$  \hspace{1cm} (9)$$

**Weight updating:** We investigated two types of learning strategies: learning that uses GRL to make domains indistinguishable, and learning that does not use GRL to distinguish domains. The model parameters $\Theta_{asr} = \{\theta_{asr1}, \theta_{asr2}\}$, $\Theta_{qm} = \{\theta_{qm1}, \theta_{qm2}\}$ and $\Theta_{dc} = \{\theta_{dc1}, \theta_{dc2}\}$ are updated during the training. When GRL is used for training, the model parameters are updated as

$$\Theta_{qm} \leftarrow \Theta_{qm} - \epsilon \beta \frac{\partial L_C}{\partial \Theta_{qm}},$$  \hspace{1cm} (10)$$

$$\Theta_{dc} \leftarrow \Theta_{dc} - \epsilon \frac{\partial L_D}{\partial \Theta_{dc}},$$  \hspace{1cm} (11)$$

$$\Theta_{asr} \leftarrow \Theta_{asr} - \epsilon (\beta \frac{\partial L_C}{\partial \Theta_{asr}} - \alpha_{\text{adv}} \frac{\partial L_D}{\partial \Theta_{asr}}),$$  \hspace{1cm} (12)$$

where $\epsilon$ is the learning rate, $\alpha_{\text{adv}}$ is the hyper parameter to change the regularization strength, $\beta$ is the weight parameter for contrastive loss. We define the parameter based on training step $p = s_{\text{cut}} / s_{\text{max}}$, where $s_{\text{cut}}$ is the current training step and $s_{\text{max}}$ is the maximum training step. The $\alpha_{\text{adv}}$ is gradually increases as $\alpha_{\text{adv}} = \frac{2}{1 + \exp(-t / \text{step})} - 1$ [25]. When GRL is not used for training, the model parameters are updated as

$$\Theta_{qm} \leftarrow \Theta_{qm} - \epsilon \beta \frac{\partial L_C}{\partial \Theta_{qm}},$$  \hspace{1cm} (13)$$

$$\Theta_{dc} \leftarrow \Theta_{dc} - \alpha_{\text{cls}} \frac{\partial L_D}{\partial \Theta_{dc}},$$  \hspace{1cm} (14)$$

$$\Theta_{asr} \leftarrow \Theta_{asr} - \epsilon (\beta \frac{\partial L_C}{\partial \Theta_{asr}} + \alpha_{\text{cls}} \frac{\partial L_D}{\partial \Theta_{asr}}),$$  \hspace{1cm} (15)$$

where $\alpha_{\text{cls}}$ is the weight of the domain loss.

### 3.2. Finetuning for Automatic Speech Recognition

We use CTC-based models for the downstream ASR tasks. We add the bi-directional long short-term memory neural network (BLSTM) following the Transformer encoder from the trained self-supervised speech representation learning model. The parameters in the quantization module and domain classifier are not used. The first input of Transformer encoder and the outputs of all the $M$ Transformer blocks are added together with trainable weight parameters and input to the BLSTM. The hidden representations $u = \{u_1, \ldots, u_T\}$ is calculated as

$$\hat{F} = \text{WeightedSum}(F^{0:M}; \theta_{asr1}),$$  \hspace{1cm} (16)$$

$$u = \text{BLSTM}(\hat{F}; \theta_{asr2}),$$  \hspace{1cm} (17)$$

where WeightedSum(·) is the weighted sum with trainable parameters, $F^{0:M}$ are the first input of the Transformer encoder and all the outputs of $M$ Transformer blocks, $\theta_{asr1}$ and $\theta_{asr2}$ are the model parameters. The probabilities of the output token is calculated as

$$v_t = \text{Softmax}(u_t; \theta_{asr3}),$$  \hspace{1cm} (18)$$

where $\theta_{asr3}$ are the model parameters. The probability of a CTC path $\alpha = \{a_1, \ldots, a_T\}$ is computed as

$$P(\alpha | X) = \prod_{t=1}^{T} v_t^{a_t},$$  \hspace{1cm} (19)$$

where $v_t^{a_t}$ is the probability of label $a_t$ at frame $t$. The probability of label sequence $Y$ is computed as

$$P(Y | X) = \sum_{\alpha \in \phi(Y)} P(\alpha | X),$$  \hspace{1cm} (20)$$

where $\phi$ is the set of all the CTC paths that can be converted to the label sequence $Y$. The sum over all the possible paths are calculated using a forward-backward algorithm. The CTC loss is defined as

$$L_{\text{CTC}} = - \sum_{(X', Y') \in \mathcal{D}_\text{PT}} \log P(Y' | X').$$  \hspace{1cm} (21)$$

In this study, we fixed the parameters from self-supervised learning. Therefore, we optimize the additional parameters $\{\theta_{asr1}, \theta_{asr2}, \theta_{asr3}\}$ with the above CTC loss.

### 4. Experiments

#### 4.1. Setups

**4.1.1. Datasets**

Table 1 and 2 show the amount of datasets for representation learning and finetuning. We prepared four domain datasets for self-supervised learning. Lecture and TV caption datasets were composed of subsets from Corpus of Spontaneous Japanese

<table>
<thead>
<tr>
<th>Domain known/unknown</th>
<th>Train</th>
<th>Dev</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>296.4</td>
<td>3.0</td>
<td>5.1</td>
</tr>
<tr>
<td>Meeting</td>
<td>44.7</td>
<td>2.5</td>
<td>2.7</td>
</tr>
<tr>
<td>Dialog</td>
<td>26.3</td>
<td>1.1</td>
<td>1.1</td>
</tr>
</tbody>
</table>

**Table 1:** Amount (hours) of unlabeled datasets for self-supervised learning.

<table>
<thead>
<tr>
<th>Domain known/unknown</th>
<th>Train</th>
<th>Dev</th>
<th>Eval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lecture</td>
<td>222.0</td>
<td>2.2</td>
<td></td>
</tr>
<tr>
<td>TV caption</td>
<td>314.4</td>
<td>3.2</td>
<td></td>
</tr>
<tr>
<td>Voice search</td>
<td>196.7</td>
<td>1.9</td>
<td></td>
</tr>
<tr>
<td>Reading</td>
<td>84.4</td>
<td>0.8</td>
<td></td>
</tr>
</tbody>
</table>

**Table 2:** Amount (hours) of labeled datasets for finetuning and evaluation. We define “known” domain as the domain included in the SSL training data and “unknown” domain as the domain not included in the SSL training data.
Table 3: CERs on evaluation sets in different datasets and models.

<table>
<thead>
<tr>
<th>Model</th>
<th>SSL data</th>
<th>known Lecture</th>
<th>known Call</th>
<th>known Meeting</th>
<th>known Dialog</th>
</tr>
</thead>
<tbody>
<tr>
<td>w2v-base</td>
<td>Lecture</td>
<td>5.30</td>
<td>5.93</td>
<td>23.29</td>
<td>16.02</td>
</tr>
<tr>
<td>w2v-base</td>
<td>All</td>
<td>5.51</td>
<td>5.64</td>
<td>23.51</td>
<td>16.27</td>
</tr>
<tr>
<td>w2v-domain-cls</td>
<td>All</td>
<td>5.63</td>
<td>5.50</td>
<td>23.55</td>
<td>16.60</td>
</tr>
<tr>
<td>w2v-domain-adv</td>
<td>All</td>
<td>5.39</td>
<td><strong>5.35</strong></td>
<td><strong>22.28</strong></td>
<td><strong>15.43</strong></td>
</tr>
</tbody>
</table>

4.1.2. Models

w2v-base is the baseline model that does not use any domain information. The speech encoder consists of seven blocks and the temporal convolutions in each block. In the Transformer encoder, the number of blocks, the hidden state dimension, non-linear layer dimension, and the number of heads were 6, 512, 2048, and 4, respectively. We applied masking to the inputs of the Transformer encoder with the ratio of 0.65 and the subsequent of 10 time steps. The number of codebooks and the entries in a codebook were set to 2 and 320. We optimized the parameters using Adam [29] with warmup step of 32k.

w2v-domain-cls is the model with multi-task learning of domain classification and speech representation construction. The model topology is the same as w2v-base except for the domain classifier. The number of layers of the domain classifier was set to 1. The parameter $\alpha_{cls}$ was set to 0.1 for the multi-task learning.

w2v-domain-adv is the model with domain-adversarial training. The model topology is the same as w2v-domain-cls. The weight parameter of contrastive loss $\beta$ was set to 0.1, and the number of negative samples in the calculation of contrastive loss was set to 100. The maximum training step $s_{\text{max}}$ for $\alpha_{adv}$ in the adversarial training was set to 100K.

In the training of downstream ASR, the parameters from self-supervised learning models were fixed. We applied SpecAugment [30] to the training data. The additional network to the self-supervised speech representation learning model was 2-layer BLSTM with 1024 units in each layer. We used Adam optimizer. During decoding, the beam size was set to 5.

4.2. Results

Table 3 shows CERs on evaluation sets with different models. Comparing w2v-base performance with different SSL data, the results confirmed that simply increasing training data did not necessarily improve the accuracy of ASR. In lecture, meeting, dialog domains, the accuracy degraded although training data was increased. We found that the conventional w2v-base did not effectively use data from different domains to construct speech representations. In the case of our w2v-domain-cls, there were both results that increased and decreased accuracy depending on the domain. This indicates that the accuracy varied depending on the similarity between the training and evaluation data since the model is more specialized to the domains in the training data. Our w2v-domain-adv outperformed all the others in the unknown domains. This confirmed that domain invariant representations can be obtained by our method.

Figure 2 shows visualizations of distribution of context representations by t-SNE [31]. We randomly selected samples from the validation sets of four domains. The samples made many small clusters regardless of their domains. Focusing on the differences among the domains, we can confirm that the distributions are independently biased in (a), the reading and voice search domains (dotted line circle in (a) of Figure 2). In (b), however, they are more evenly distributed than in (a), regardless of the domain. This is because our method is able to learn so that the adversarial network could not classify the domains.

5. Conclusions

We proposed domain-adversarial self-supervised speech learning that use a domain-adversarial neural network. By learning speech representations so that the domains cannot be distinguished, our models are robust against unknown domains. We investigated multi-task learning with domain classification for constructing speech representations. The results indicate that our proposed method outperforms the conventional training method of wav2vec 2.0 in unknown domain downstream ASR tasks.

(CSJ) [26] and LaboroTVspeech [27]. For the domain of voice search, we prepared speech command data from our in-house dataset. Japanese Newspaper Article Sentences (JNAS) [28] was used as a reading domain dataset. We define “known” domain as the domain included in the SSL training data and “unknown” domain as the domain not included in the SSL training data. For the finetuning and evaluation, one known domain dataset was lecture domain from CSJ and three unknown datasets were call, meeting and dialog domains from our in-house data. The evaluation sets of CSJ were three standard sets. Sampling rate was 16k for all audio recordings.
6. References


