2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) | 979-8-3503-0689-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ASRU57964.2023.10389757

TOWARDS MATCHING PHONES AND SPEECH REPRESENTATIONS

Gene-Ping Yang, Hao Tang

Centre for Speech Technology Research, University of Edinburgh

ABSTRACT

Learning phone types from phone instances has been a long-standing problem, while still being open. In this work, we revisit this problem in the context of self-supervised learning, and pose it as the problem of matching cluster centroids to phone embeddings. We study two key properties that enable matching, namely, whether cluster centroids of self-supervised representations reduce the variability of phone instances and respect the relationship among phones. We then use the matching result to produce pseudo-labels and introduce a new loss function for improving self-supervised representations. Our experiments show that the matching result captures the relationship among phones. Training the new loss function jointly with the regular self-supervised losses, such as APC and CPC, significantly improves the downstream phone classification.

***Index Terms*—** self-supervised learning, acoustic unit discovery, Gromov–Wasserstein distance

1. INTRODUCTION

Frame representation produced by self-supervised models have shown to work well at distinguishing phone identities at the frame level [1, 2, 3, 4] and at the segment level [5, 6, 7]. Although phones are thought to be largely separable in the space of speech representation, the separability is based on *instances*. Whether it is possible to learn representations of *types* as opposed to instances remains an open problem.

In this paper, we will focus on phone types, because phonetic properties are salient even after crude quantization down to 100 or even 50 codes [8, 9]. The problem of learning phone types has a long history and shows up under different names, such as acoustic unit discovery [10, 11, 12] and unsupervised speech recognition [13, 14, 15, 16, 17, 18]. It is also highly related to unsupervised phone segmentation [19, 11, 12] and lexical discovery [20].

Most, if not all, approaches assume that types arise from clustering of instances. Approaches based on dynamic time warping for acoustic unit discovery rely on clustering [10]. Bayesian models for acoustic unit discovery is another form of clustering [21]. Much of the recent progress focuses on adversarial approaches to distinguish sequences of cluster IDs and phone sequences observed in data sets [13, 14, 15, 16, 17, 22]. In this paper, we follow the same footsteps and study whether cluster centroids of instances can be a good representation for types in the context of self-supervised learning.

Even though cluster centroids have been used as types for almost all approaches to learning phone types, it is also true that the types discovered by these approaches are never one to one. There is abundant variability of phones within and across speakers, and we generally do not know the correct number of clusters to use. Recent work on analyzing representations of self-supervised predictive coding finds that phonetic and speaker information are largely represented in orthogonal subspaces [23]. We will use this as a tool to measure and reduce the variability of phone instances among speakers.

For the second problem where there is not always a one-to-one mapping between the centroids and phone types, we take a different route and assume access to a set of phone embeddings. We introduce an additional *matching* step to find the correspondence between the centroids and phone embeddings. This is reminiscent to the line of work in unsupervised machine translation [24, 25, 26]. We will adopt a similar approach, using optimal transport, in particular, optimizing the Gromov-Wasserstein distance [27, 28] to match the centroids and phone embeddings.

We see several applications once we match the centroids and phone embeddings. One immediate application is to use the matching result to provide pseudo-labels on unlabeled speech. The pseudo-labels can be used to further improve the speech representations. We introduce a loss function predicting pseudo-labels as self-supervised learning, similar to [8, 29].

Our experiments show that centroids of self-supervised frame representation provide a good starting point for learning types. We will show quantitative and qualitive results for matching centeroids to phone embeddings. In addition, by predicting the pseudo-labels produced by the matching result gives a sizable improvement in downstream phone classification.

2. PROPOSED APPROACH

An overview of our proposed approach is illustrated in Figure 1. The goal is to match type vectors derived from a self-supervised model to phone embeddings (without using paired transcriptions). Vectors produced by self-supervised mod-

979-8-3503-0689-7/23/$31.00 ©2023 IEEE

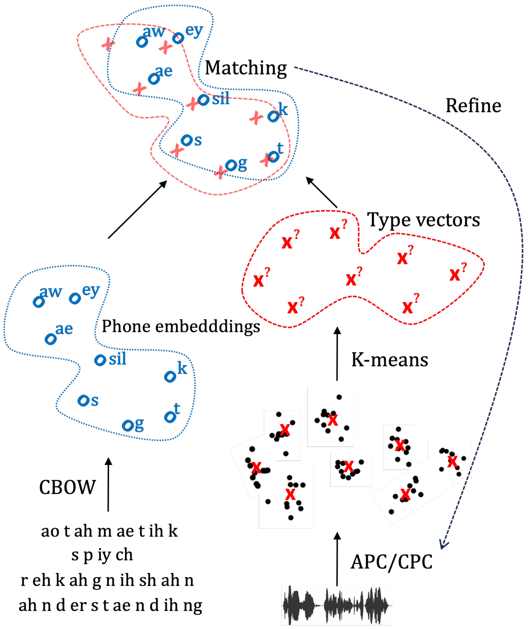


Fig. 1: An overview of our proposed approach. Phone embeddings are learned from phone sequences with CBOW. Type vectors are derived from clustering self-supervised frame representations. Matching assigns phone labels to each centroid and can be used to provide pseudo-labels for unlabeled speech. Based on the matching result, an additional objective is introduced for self-supervised learning that predicts the pseudo-labels.

els are contextualized, meaning that different phone instances would have different sequences of vectors. Since vectors produced by self-supervised models tend to be sufficiently clustered by phones [8], we make the first assumption that a type can be represented by the mean of its instances. As a result, we define **type vectors** to be centroids after clustering frame vectors produced by self-supervised models.

We further make a second assumption that the derived type vectors respect the relationship among phones, for example, similar pronouncing phones are closer in space. If we have access to phone embeddings that also respect the relationship of phones, the two sets of vectors have the potential to be matched. In this work, we use phone embeddings from a continuous bag-of-word (CBOW) model [30, 31] trained on phone sequences extracted from texts and a pronunciation dictionary.

To match the two sets of vectors, we use optimal transport, in particular, optimizing the Gromov-Wasserstein distance [27, 28]. Finally, with the goal of improving speech representations, we infer the matched phone labels for every frame, and treat them as pseudo-labels for self-supervised learning.

Our approach is unsupervised, in the sense that it does not require manual transcriptions paired with speech. The dependency on texts and a pronunciation dictionary may seem stringent. However, as we will show in the experiments, we observe improvements even when the matching result is only approximate. This suggests that the requirement on texts and a pronunciation dictionary can be relaxed, and we leave it as future work.

3. ASSUMPTIONS AND VALIDATION

There are two main assumptions in our approach: one is that the mean of instances would be representative of the instances and the other is that type vectors respect the relationship of phones. In the first assumption, whether the mean is representative of the instances depends on the variability of frame representations. In the second assumption, whether the relationship of phones are respected depends on how they are organized in space and their nearest neighbors. We will study both assumptions in this section.

3.1. Variability of Phone Instances

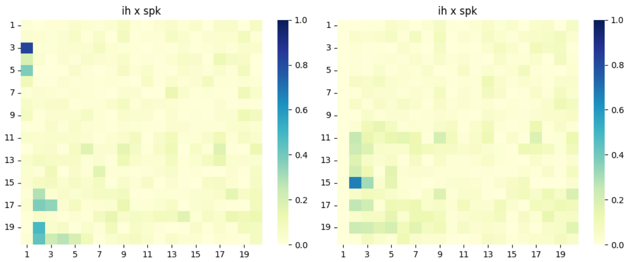
To study the variability of phone instances, inspired by [23], we collect all frame vectors produced by a self-supervised model of the same phone label into a matrix and compute PCA on it. Specifically, suppose all frames produced by a self-supervised model are forced aligned to phones. We compute PCA on the matrix M^p whose column vectors are frames labeled as phone p in a data set. In other words, the set of eigenvectors we get spans the subspace of a particular phone and captures the variance of the instances of that particular phone.

Similarly, we define the speaker subspace as the span of speaker vectors, each of which is an average of frame vectors that have the same speaker label. We compute PCA on $M^{\text{spk}} = [m_{s_1}, \dots, m_{s_J}]$ for a set of J speakers $\{s_1, \dots, s_J\}$ where $m_s = \text{avg}(\{h : \text{the frame } h \text{ is from speaker } s\})$. We compute PCA on the speaker vectors as well to obtain eigenvectors for the speaker subspace. Given two sets of eigenvectors (one from phone instances and one from speakers), we can study how correlated they are.

We conduct our analysis on the 100-hour subset of Librispeech with 3-layer LSTMs trained with APC [3] and CPC [1] (more details in the experimental section). In Figure 2, we focus on APC and use the phone [ih] as an example and show the absolute of dot products of the two sets of eigenvectors. In Figure 2 (a), the first speaker direction (ranked third) contributes significantly to the variability of the phone instances.[1] We adopt the idea in [23], projecting a frame vector h to a subspace orthogonal to the first speaker direction v (called collapsing in [23])

$$h_c = h - (h^\top v)v, \tag{1}$$

[1] The first speaker direction likely indicates the average fundamental freqeuencies, separating males and females [23].



(a) before collapsing (b) after collapsing

Fig. 2: The absolute of dot products among the top 20 eigenvectors from the APC speaker subspace (x-axis) and those from the frame representation of the phone [ih] (y-axis). (a) The absolute of dot products before collapsing the first speaker direction. (b) The same measures after collapsing the first speaker direction.

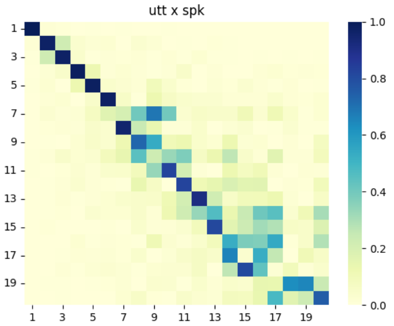


Fig. 3: The absolute of dot products comparing the top 20 principle directions of the APC speaker subspace (x-axis) and those from the utterance subspace (y-axis).

where h_c is the collapsed frame vector. The resulting absolute of dot products is shown in Figure 2 (b). Collapsing effectively reduces the variability along the speaker directions. We could continue collapsing but the second speaker direction (ranked 15th) contributes significantly less than the first.

The collapsing approach relies on speaker labels that are not always at hand. We discover an alternative approach to find the speaker direction without using speaker labels. Instead of averaging frame vectors based on speakers, we average frame vectors based on utterances and compute PCA on it. The absolute of dot products of the two approaches are shown in Figure 3, and the eigenvectors are surprisingly well aligned for the first few directions.

The findings hold for most phones and for both APC and CPC. Later in the experiments, we will approximate speaker directions with utterances and collapse the first speaker dimension before clustering. Controlling variability of instances is deemed essential in unsupervised automatic speech recognition. For example, [16] relies on silence removal, dimensionality reduction with PCA after k-means, and mean pooling.

Table 1: Top 3 nearest neighbors of phone types from APC and CBOW. Phone types from APC are the averages of frame representation (of the same phone label) based on forced alignments. Nearest neighbors that appear in both sets are underlined.

Phone	APC	CBOW	Phone	APC	CBOW
ae	eh $\underline{\text{aw}}$ ay	ih ow $\underline{\text{aw}}$	k	g, $\underline{\text{t}}$, p	m, $\underline{\text{t}}$, f
ah	spn, $\underline{\text{ih}}$, uh	er, $\underline{\text{ih}}$, ow	m	n, spn, b	t, s, l
aw	aa, $\underline{\text{ae}}$, eh	$\underline{\text{ae}}$, ay, ih	p	b, k, $\underline{\text{t}}$	s, $\underline{\text{t}}$, m
er	r, $\underline{\text{spn}}$, $\underline{\text{ah}}$	$\underline{\text{ah}}$, $\underline{\text{spn}}$, ih	t	$\underline{\text{d}}$, spn, jh	$\underline{\text{d}}$, m, z
ey	iy, ih, $\underline{\text{eh}}$	$\underline{\text{eh}}$, aa, ah	th	f, dh, $\underline{\text{t}}$	$\underline{\text{t}}$, z, ch
ih	$\underline{\text{ah}}$, spn, eh	$\underline{\text{ah}}$, ow, ae	z	$\underline{\text{s}}$, spn, $\underline{\text{t}}$	d, $\underline{\text{t}}$, $\underline{\text{s}}$
ch	jh, $\underline{\text{t}}$, sh	$\underline{\text{t}}$, m, z	s	$\underline{\text{z}}$, spn, t	m, $\underline{\text{z}}$, b
d	$\underline{\text{t}}$, spn, n	$\underline{\text{t}}$, z, m	sh	$\underline{\text{ch}}$, zh, jh	t, s, $\underline{\text{ch}}$
g	$\underline{\text{k}}$, b, d	$\underline{\text{k}}$, f, b	v	$\underline{\text{b}}$, spn, f	$\underline{\text{b}}$, m, n
jh	ch, $\underline{\text{t}}$, d	$\underline{\text{t}}$, v, b	zh	sh, $\underline{\text{jh}}$, ch	$\underline{\text{jh}}$, t, f

3.2. Nearest Neighbors of Phone Types

To study whether the type vectors respect the relationship of phones, we use forced alignments (only for analysis) and compute the means of frame vectors that have the same phone label. Table 1, based on APC, shows the nearest neighbors of the means and the phone embeddings learned from CBOW. We observe that the vectors in the two spaces share similar nearest neighbors. In addition, vowels share nearest neighbors with vowels, while consonants share nearest neighbors with consonants. Manners and places of articulation for consonants also form groups. This provides us evidence that there is enough signal to match the two spaces. It also suggests that an accurate matching is not required (if possible at all). An approximate match would provide training signals to distinguish vowels and consonants or even the manners and places.

4. MATCHING

There are many algorithms that aim to match two sets of vectors. Based on the analyses in the previous section, we choose to optimize the Gromov–Wasserstein distance [27, 28] as it respects distances among vectors and does not assume a linear transformation between the two spaces. Suppose the set of vectors $\{c_1, \dots, c_n\}$ are the centroids after clustering the frame vectors produced by a self-supervised model, and the set of vectors $\{y_1, \dots, y_m\}$ are the phone embeddings learned from CBOW. Note that n and m are not required to be the same. We compute two distances matrices S and S' on the two spaces, where $S_{ij} = \|c_i - c_j\|_2^2$ and $S'_{ij} = \|y_i - y_j\|_2^2$. The goal is to find Γ where Γ_{ij} indicates the probability of matching c_i to y_j. When c_i is matched to y_j and c_k is matched to y_l, we incur a cost $(S_{ik} - S'_{j\ell})^2$ for not respecting the distances. Based on these definitions, the Gromov-Wasserstein distance is defined as

$$\min_{\Gamma} \quad \sum_{ij} \sum_{k\ell} (S_{ik} - S'_{j\ell})^2 \Gamma_{ij} \Gamma_{k\ell} - \epsilon H(\Gamma)$$
$$\text{s.t.} \quad \Gamma \mathbf{1} = p, \quad \Gamma^\top \mathbf{1} = q, \quad \Gamma \geq 0$$

where $\mathbf{1}$ is an all-one vector, $H(\Gamma)$ is the sum of entropy for all rows in Γ, and ϵ is the entropy regularization parameter. The objective additionally has two unigram distribution vectors p and q. In our case, p is the unigram distribution of centroids, and q is the unigram distribution of phones in the texts for training CBOW. Both can be computed without relying on manual transcriptions or forced alignments.

To optimize the Gromov–Wasserstein distance, following [28], we iteratively compute a loss matrix L and the matching matrix Γ as follows.

$$L \leftarrow S^2 p \mathbf{1}^\top + \mathbf{1} q^\top S'^{2\top} - 2 S \Gamma S'^\top \tag{2}$$
$$a \leftarrow p \oslash Kb, \quad b \leftarrow q \oslash K^\top a \tag{3}$$
$$\Gamma \leftarrow \text{diag}(a) K \text{diag}(b) \tag{4}$$

where $K = e^{-L/\epsilon}$. We technically do not need to learn a linear transformation between the two spaces (if the two spaces can be matched at all with a linear transformation). If this is necessary, we can use Procrustes analysis to find the linear transformation between the two spaces. Specifically, if the SVD of $C\Gamma Y^\top$ is $U\Sigma V^\top$, the linear transformation between the two spaces is $UV^\top$.

5. PSEUDO-LABELS FROM MATCHING

Once matching is done, every centroid is matched to a phone embedding. We can label each frame vector based on which centroid it belongs and what phone embedding it gets matched. This process creates sequences of pseudo-labels for unlabeled speech without relying on manual transcriptions. With the goal of improving speech representations, we can create a new self-supervised loss function, predicting the pseudo-labels discovered with this process.

More formally, we assign each frame at t a pseudo-label

$$s_t = \operatorname*{argmax}_i \Gamma_{z_t, i} \tag{5}$$

where z_t is the centroid that the frame belongs. In other words, the pseudo-label is the matched phone given the centroid that the frame belongs. Predicting pseudo-labels is a common approach to self-supervised learning. For example, wav2vec 2.0 uses the quantized output of the convolution layers as pseudo-labels [32], while HuBERT uses quantized MFCCs or one of its own layers as pseudo-labels [8]. There is even evidence that randomly quantizing log Mel features (also known as BEST-RQ in [29]) could serve as pseudo-labels for self-supervised learning.

When predicting the pseudo-labels, we have two options: one is to use cross entropy loss and the other is to get close to the target embedding with the ℓ_2 loss. For cross entropy loss, we add the loss

$$\sum_{t=1}^{T-k} \mathbf{1}_{s_{t+k}}^\top \log \text{softmax}(W h_t), \tag{6}$$

to measure the prediction quality, where $\mathbf{1}_i$ is a one-hot vector with the i-th dimension set to 1, h_t is the frame vector produced by a self-supervised model at time t, W is for linear prediction, and s_{t+k} is the pseudo-label. As an aside, if s_{t+k} is the cluster ID of clustering log Mel features with a random codebook, this approach becomes a special case of BEST-RQ [29].

In the second case, more formally, recall that y_i is the i-th phone embedding from CBOW. To predict the target embedding, we will add

$$\sum_{t=1}^{T-k} \|y_{s_{t+k}} - W h_t\|_2^2 \tag{7}$$

to the training objective where W is for linear regression, and $y_{s_{t+k}}$ is the embedding looked up with the pseudo-label s_{t+k}. Note that we have applied a time shift of k on the targets to be consistent with APC and CPC.

6. EXPERIMENTS

Following the protocol of [3], we evaluate our proposed approach by pre-training on the 360-hour subset of LibriSpeech and performing phone classification on Wall Stree Journal (WSJ). The pre-trained models are frozen after pre-training, as we are interested in the quality of speech representations after pre-training. Ten percent of the training set (si284) in WSJ is used as a development set for choosing the best phone classifier. We report the phone error rates of the selected probing models on dev93 and eval92. A phone set with 41 phones are used in our study, including one label (sil) for silence and one (spn) for spoken noise. To compare against prior work, we use 3-layer uni-directional LSTMs with a hidden size of 512 dimension in all experiments. To test the generalization across self-supervised losses, we compare both APC [3] and CPC [1]. We choose a time shift of 5 for both APC and CPC, and train them with a batch size of 32 and a step size of 10^{-3} using the Adam optimizer. For CPC, following [2], we do not use convolution layers and directly contrast on log Mel features. A set of 100 negative samples are uniformly drawn from the log Mel frames of the same utterance. All models are trained with 15 epochs for pre-training, and another 15 epochs for phone classification.

For CBOW on phone sequences, we use the texts on the 100-hour subset of Librispeech and CMUdict version 0.7. We train CBOW with a window size of 5, a step size of 0.005, a

Table 2: Analyses on clusters and matching with forced alignments. The evaluation metrics are phone purity (PP), cluster purity (CP), and frame and type phone error rates (PER). We compare APC, CPC, and their counterpart APC$_c$ and CPC$_c$ after collapsing the first speaker direction. HuBERT and random projection are also listed.

frame vectors	PP (↑)	CP (↑)	frame PER (↓)	type PER (↓)
APC	50.1	37.3	50.1	93.1
APC$_c$	47.5	36.6	50.1	76.8
CPC	51.1	37.7	48.9	73.0
CPC$_c$	50.6	38.2	47.8	72.6
HuBERT	50.3	37.8	48.3	-
Rand Proj	22.1	29.5	73.0	-

batch per utterance, and gradient clipping to norm 5.[2] Given the relatively small number of classes, we consider all classes and do not use negative samples.

6.1. Matching with Gromov-Wasserstein Distance

Before matching, we first evaluate the quality of clustering frame vectors produced by self-supervised models. Following [8, 9], we analyze the phone purity (PP), cluster purity (CP) and phone error rates (PER) with forced alignments. Note that phone purity first computes a purity for each individual cluster and averages them, while frame phone error rates is the total number of errors summed over all clusters divided by the total number number of frames.

We run Lloyd's algorithm for k-means clustering with 50 clusters on the 100-hour subset of LibriSpeech for 20 epochs. We choose the second LSTM layer for APC and CPC, as well as the ninth transformer layer for HuBERT, because the phonetic information is most prominent in these layers [3, 8]. For APC and CPC, we also include the ones after collapsing the first speaker direction (denoted APC$_c$ and CPC$_c$). We include random projection on log Mel features [29], initializing the projection matrix using the Glorot initialization [33] and the codebook with a unit Gaussian.

Results are shown in Table 2. Surprisingly, similar performance has been observed among APC, CPC and HuBERT, despite that HuBERT is much larger in size and trained with more data. There is a drastic difference in phone purity and frame PER for random projection. This is due to large imbalance in cluster sizes. We will leave HuBERT and random projection for future work and focus on APC and CPC for the rest of the paper.

Next, we evaluate the quality of matching centroids to phone embeddings with Gromov–Wasserstein distance. We compute type phone error rates, i.e., how often an incorrect phone label is assigned to a centroid. The ground truth label of a centroid is determined with the majority vote of frames based on forced alignments. Prior to computing the distance matrices, we center and normalize the vectors to unit norm, following [25, 26]. We use entropy regularization of 0.0005 for APC and 0.01 for CPC. The iterative algorithm is run for 1,000 iterations. The unigram distribution for both the centroids and the phones are computed on the respective data sets that they are trained on. Results are shown in the last column of Table 2. Collapsing the first speaker direction improves both frame and type PERs.

To evaluate the matching result qualitatively, we use Procrustes analysis to learn the linear transformation of the two spaces and plot the two spaces together with tSNE. The visualization for APC is shown in Figure 4. Though the type PERs are high, the matching is qualitatively successful in grouping vowels, consonants, and silence. Within consonants, fricatives, stops, and nasals also form groups.

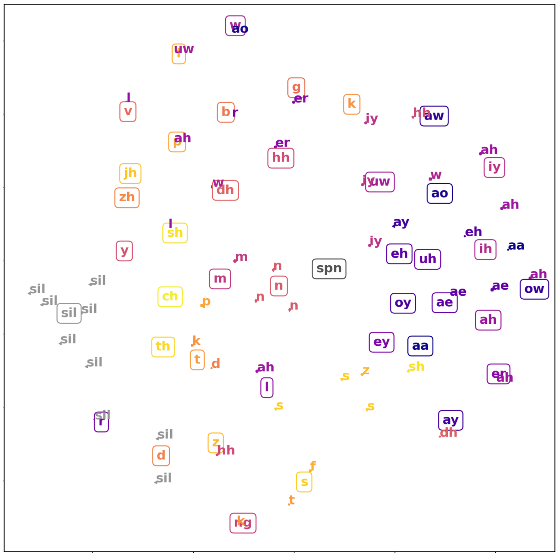


Fig. 4: A t-SNE plot of the matching result. The type vectors are projected onto the space of phone embeddings using a linear transformation learned with Procrustes analysis. Phone embeddings are boxed, while the centroids are not. The color uses the first phone direction in [23] that correlates with sonority.

6.2. Self-Supervised Learning with Pseudo-Labels

Though matching between centroids and phone embeddings produces high type PERs, there are relationship among phones recovered from matching is still valuable for speech representation learning. We produce pseudo-labels for the 360-hour subset of LibriSpeech by looking up the centroid each frame belongs to and the phone embedding it is matched.

[2] Incidentally, there is little prior work on CBOW for learning phone embeddings. There was a blog post on this by Gabriel Synnaeve written in 2014, and an unpublished manuscript by Michael Hammond in 2020.

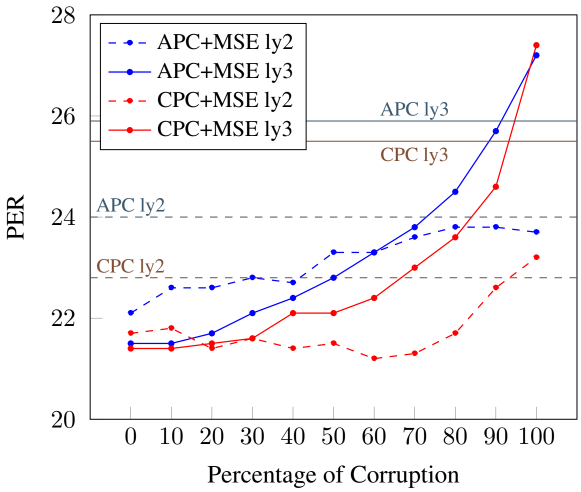


Fig. 5: Phone error rates (PER) for evaluating joint self-supervised and supervised pre-training under different percentages of corruption to forced alignments. Both layer 2 (ly2) and layer 3 (ly3) of LSTMs are reported. We measure the prediction of phone embeddings with ℓ_2 loss, hence the label MSE.

Table 3: Phone error rates (PER) for the downstream phone classification on WSJ. The additional loss function is annotated with MSE for the ℓ_2 against the target embeddings and CE for the cross entropy loss against the codes. The settings for forced alignments with 77% corruption (FA @ 77%) are also listed.

objective	pseudo-labels	layer 2		layer 3	
		dev93	eval92	dev93	eval92
APC	-	22.0	22.0	23.8	24.2
APC + MSE	Matching	**20.5**	**19.9**	22.5	22.2
APC + MSE	Rand Proj	22.5	22.3	26.8	28.5
APC + CE	Rand Proj	21.3	21.1	24.5	24.8
CPC	-	20.7	20.3	23.2	23.4
CPC + MSE	Matching	**20.0**	**19.6**	21.9	21.8
CPC + MSE	Rand Proj	22.5	22.7	28.0	29.5
CPC + CE	Rand Proj	20.8	20.4	24.2	24.8
APC + MSE	FA @ 77%	21.4	21.2	22.3	22.2
CPC + MSE	FA @ 77%	19.5	19.2	21.2	20.9

We introduce a new self-supervised loss function by predicting the target embeddings of the pseudo-labels.

Note that if we obtain a perfectly accurate match between the two spaces, the task of predicting pseudo-labels falls back to supervised phone classification. Based on this observation, we first conduct a topline of jointly training with a self-supervised loss function and a supervised loss function [34, 35]. We further conduct a set of experiments, adding corruption to phone labels to simulate joint training with matching. Note that when the percentage of corruption is 100%, we randomly sample a phone embedding to predict, so it is different from BEST-RQ [29] that predicts codeword vectors from a random codebook.

Results of adding corruption to phone labels during joint self-supervised and supervised training are shown in Figure 5. In particular, since we get about 77% type errors in Table 2 for APC, we are interested in the case where the pseudo-labels have high type errors. We observe improvements in phone classification even at a high percentage of corruption. This suggests that even when the matching result has a type PER of 77%, we can still expect improvements.

Results of predicting pseudo-labels derived from matching are shown in Table 3. We observe a 1.5% and 2.1% absolute improvement over regular APC on dev93 and eval92. The improvements are observed for both layer 2 and layer 3. The results are also similar when training with CPC. We include random projection on log Mel features as another set of baseline. We follow [29], quantizing log Mel features with a random codebook sampled from unit Gaussian. We also include a variant where we use ℓ_2 (MSE) as the loss and the codeword vectors as targets, to compare with the matching result using the same loss function. Our proposed approach is better than random projection. Finally, we list the result of corrupting 77% of the phone labels based on forced alignments. Jointly predicting pseudo-labels and training with APC actually performs better, while the CPC counterpart is not far behind.

7. CONCLUSION

In this paper, we have studied the space of self-supervised representations and presented a novel approach to match frame representations to phone embeddings without paired manual transcriptions. While the type error rates are high after matching, qualitative analysis shows promising results. Using the matching result to generate pseudo-labels as prediction targets, we observe a sizeable improvement on the downstream phone classification. The approach can be further refined by iterating between matching and self-supervised learning. Our positive result also suggests that it might be possible to jointly perform matching while optimizing a self-supervised loss function.

8. REFERENCES

[1] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, “Representation learning with contrastive predictive coding,” *arXiv:1807.03748*, 2018.

[2] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, “An unsupervised autoregressive model for speech representation learning,” in *Interspeech*, 2019.

[3] Gene-Ping Yang, Sung-Lin Yeh, Yu-An Chung, James Glass, and Hao Tang, “Autoregressive predictive cod-

ing: A comprehensive study," *IEEE Journal of Selected Topics in Signal Processing*, 2022.

[4] Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Ko tik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung yi Lee, "SUPERB: Speech Processing Universal PERformance Benchmark," in *Interspeech*, 2021.

[5] Jan Chorowski, Grzegorz Ciesielski, Jarosław Dzikowski, Adrian Lańcucki, Ricard Marxer, Mateusz Opala, Piotr Pusz, Paweł Rychlikowski, and Michał Stypułkowski, "Aligned contrastive predictive coding," in *Interspeech*, 2021.

[6] Saurabhchand Bhati, Jesús Villalba, Piotr Żelasko, Laureano Moro-Velazquez, and Najim Dehak, "Unsupervised speech segmentation and variable rate representation learning using segmental contrastive predictive coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[7] Santiago Cuervo, Adrian Lancucki, Ricard Marxer, Paweł Rychlikowski, and Jan K Chorowski, "Variable-rate hierarchical CPC leads to acoustic unit discovery in speech," in *NeurIPS*, 2022.

[8] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[9] Dan Wells, Hao Tang, and Korin Richmond, "Phonetic analysis of self-supervised representations of english speech," in *Interspeech*, 2022.

[10] Alex S. Park and James R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[11] Maarten Versteegh, Roland Thiollière, Thomas Schatz, Xuan Nga Cao, Xavier Anguera, Aren Jansen, and Emmanuel Dupoux, "The zero resource speech challenge 2015," in *Interspeech*, 2015.

[12] Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux, "The zero resource speech challenge 2017," in *ASRU*, 2017.

[13] Da-Rong Liu, Kuan-Yu Chen, Hung yi Lee, and Lin shan Lee, "Completely Unsupervised Phoneme Recognition by Adversarially Learning Mapping Relationships from Audio Embeddings," in *Interspeech*, 2018.

[14] Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-Yi Lee, and Lin shan Lee, "Completely Unsupervised Phoneme Recognition by a Generative Adversarial Network Harmonized with Iteratively Refined Hidden Markov Models," in *Interspeech*, 2019.

[15] Da-rong Liu, Po-chun Hsu, Yi-chen Chen, Sung-feng Huang, Shun-po Chuang, Da-yi Wu, and Hung-yi Lee, "Learning phone recognition from unpaired audio and phone sequences based on generative adversarial network," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2022.

[16] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, "Unsupervised speech recognition," in *NeurIPS*, 2021.

[17] Alexander H Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski, "Towards end-to-end unsupervised speech recognition," in *SLT*, 2022.

[18] Chih-Kuan Yeh, Jianshu Chen, Chengzhu Yu, and Dong Yu, "Unsupervised speech recognition via segmental empirical output distribution matching," in *ICLR*, 2019.

[19] Felix Kreuk, Joseph Keshet, and Yossi Adi, "Self-supervised contrastive learning for unsupervised phoneme segmentation," in *Interspeech*, 2020.

[20] Herman Kamper, Aren Jansen, and Sharon Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech and Language*, 2017.

[21] Chia-ying Lee and James Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *ACL*, 2012.

[22] Amitay Sicherman and Yossi Adi, "Analysing discrete self supervised speech representation for spoken language modeling," in *ICASSP*, 2023.

[23] Oli Danyi Liu, Hao Tang, and Sharon Goldwater, "Self-supervised predictive coding models encode speaker and phonetic information in orthogonal subspaces," in *Interspeech*, 2023.

[24] Guillaume Lample, Alexis Conneau, Marc'Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou, "Word translation without parallel data," in *ICLR*, 2018.

[25] David Alvarez-Melis and Tommi Jaakkola, "Gromov-Wasserstein alignment of word embedding spaces," in *EMNLP*, 2018.

[26] Jean Alaux, Edouard Grave, Marco Cuturi, and Armand Joulin, "Unsupervised hyper-alignment for multilingual word embeddings," in *ICLR*, 2019.

[27] Facundo Mémoli, "Gromov–wasserstein distances and the metric approach to object matching," *Foundations of computational mathematics*, 2011.

[28] Gabriel Peyré, Marco Cuturi, and Justin Solomon, "Gromov-wasserstein averaging of kernel and distance matrices," in *ICML*, 2016.

[29] Chung-Cheng Chiu, James Qin, Yu Zhang, Jiahui Yu, and Yonghui Wu, "Self-supervised learning with random-projection quantizer for speech recognition," in *ICML*, 2022.

[30] Tomás Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean, "Efficient estimation of word representations in vector space," in *ICLR*, 2013.

[31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean, "Distributed representations of words and phrases and their compositionality," in *NeurIPS*, 2013.

[32] Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[33] Xavier Glorot and Yoshua Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *AISTATS*, 2010.

[34] Chengyi Wang, Yu Wu, Yao Qian, Kenichi Kumatani, Shujie Liu, Furu Wei, Michael Zeng, and Xuedong Huang, "UniSpeech: Unified speech representation learning with labeled and unlabeled data," in *ICML*, 2021.

[35] Junwen Bai, Bo Li, Yu Zhang, Ankur Bapna, Nikhil Siddhartha, Khe Chai Sim, and Tara N. Sainath, "Joint unsupervised and supervised training for multilingual ASR," in *ICASSP*, 2022.