SIMILARITY ANALYSIS OF SELF-SUPERVISED SPEECH REPRESENTATIONS

Yu-An Chung[1], *Yonatan Belinkov*[2*], *James Glass*[1]

[1]MIT Computer Science and Artificial Intelligence Laboratory, Cambridge, MA 02139, USA
[2]Technion Henry and Marilyn Taub Faculty of Computer Science, Haifa 3200003, Israel

{andyyuan,glass}@mit.edu, belinkov@technion.ac.il

ABSTRACT

Self-supervised speech representation learning has recently been a prosperous research topic. Many algorithms have been proposed for learning useful representations from large-scale unlabeled data, and their applications to a wide range of speech tasks have also been investigated. However, there has been little research focusing on understanding the properties of existing approaches. In this work, we aim to provide a comparative study of some of the most representative self-supervised algorithms. Specifically, we quantify the similarities between different self-supervised representations using existing similarity measures. We also design probing tasks to study the correlation between the models' pre-training loss and the amount of specific speech information contained in their learned representations. In addition to showing how various self-supervised models behave differently given the same input, our study also finds that the training objective has a higher impact on representation similarity than architectural choices such as building blocks (RNN/Transformer/CNN) and directionality (uni/bidirectional). Our results also suggest that there exists a strong correlation between pre-training loss and downstream performance for some self-supervised algorithms.

Index Terms— Self-supervised learning, speech representation learning, unsupervised pre-training, comparative analysis

1. INTRODUCTION

Self-supervised learning is a form of unsupervised learning that treats the input or modifications of the input as learning targets. Thanks to this property, self-supervised learning can leverage large-scale unlabeled data for training, and has enjoyed success in learning high-level representations of data from different modalities [1, 2, 3].

Recently, self-supervised approaches for learning speech representations have received great research attention. Methods like contrastive predictive coding [4], autoregressive predictive coding [5], masked predictive coding [6, 7, 8], and problem-agnostic speech encoder [9] have been shown to be capable of learning representations that capture high-level properties of speech that are not easily accessible from surface features such as audio waveforms and spectrograms. These methods have been further extended or improved for tackling a wide range of speech applications, including speech recognition [10, 11, 12, 13, 14], speech translation [15, 16], speaker verification [17], unsupervised unit discovery [18], and unsupervised phoneme segmentation [19], to name a few.

Despite the recent progress in self-supervised speech representation learning, most of the effort is made to develop new algorithms or adapt existing methods to particular tasks, and only a few studies focus on reviewing existing approaches. In this work, we aim to provide a comparative study on some of the most representative self-supervised algorithms: contrastive predictive coding (CPC), autoregressive predictive coding (APC), and masked predictive coding (MPC). Our analysis focuses on the following two aspects. First, we hope to understand the similarity of different self-supervised representations. To carry out this study, we adopt two similarity measures for quantifying the similarity of two given representations. Although such a similarity analysis approach cannot discern absolute facts about the representations, it allows us to compare representations without subscribing to any specific type of information, and helps us answer questions like: Given the same input, how similar are different self-supervised representations? Which modeling choices, e.g., building blocks (RNN/Transformer/CNN) and directionality (uni/bidirectional), have a higher impact on similarity? How much does a model change when it is trained on more data?

Our second area of investigation examines, for each self-supervised algorithm, how well its pre-training loss correlates with downstream performance. We use phonetic and speaker classification as probing tasks to measure the amount of phonetic and speaker information contained in the representations as a function of pre-training loss. This study could be useful for model selection if there exists a strong correlation between them.

Only a few studies have focused on analyzing self-supervised models. Chung et al. (2020) proposed to incorporate vector quantization layers to restrict model capacity during pre-training so as to uncover a model's preference in preserving speech information for achieving a maximal self-supervised objective [20]. Blandón and Räsänen (2020) studied the correlation between the self-supervised loss of APC and CPC and their performance on a phoneme discrimination task [21], which has the same goal as our second study. However, neither of these two works investigated the similarity between different self-supervised representations. For the correlation study, we also consider more self-supervised models with diverse modeling choices as compared to previous work [21].

Our analysis yields the following insights:

- The objective has a higher impact on representation similarity than model architecture.

- Under the same objective, a model's directionality (uni/bidirectional) affects representation similarity more than its building blocks (RNN/Transformer/CNN).

- Both APC and MPC have a stronger correlations between pre-training loss and phonetic and speaker classification performance than CPC does.

- While all models benefit from increasing the size of unlabeled training data, CPC is found to make use of these additional data more efficiently than APC and MPC.

---

2. ANALYSIS METHODS

We study two aspects of self-supervised speech representation learning: (1) the similarity between representations learned by various models, and (2) how well their self-supervised pre-training loss correlates with downstream performance. We describe our methods for analyzing these two aspects in Sections 2.1 and 2.2, respectively.

2.1. Approaches for measuring representation similarity

Consider a pre-trained self-supervised model M. For an acoustic feature sequence (in our case, a log Mel spectrogram) $\mathbf{x} = (\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^{80}$, from a dataset D, the model M transforms $\mathbf{x}$ into a representation $M(\mathbf{x}) = (\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_T)$, where $\mathbf{m}_t \in \mathbb{R}^{512}$. Given two representations extracted by two self-supervised models $M^{(1)}$ and $M^{(2)}$, a similarity measure outputs $\text{sim}(M^{(1)}(\mathbf{x}), M^{(2)}(\mathbf{x})) \in \mathbb{R}$ that quantifies their similarity. Note that this approach does not require D to be annotated.

Existing similarity measures are proposed to capture different similarity notions. Some focus on capturing the localization of information of two representations, which is usually done by comparing the behaviors of two individual elements $\mathbf{m}_i^{(1)}$ and $\mathbf{m}_j^{(2)}$ from $M^{(1)}(\mathbf{x})$ and $M^{(2)}(\mathbf{x})$, respectively [22]. Other measures emphasize distributivity of information and find correlations between two representations $M^{(1)}(\mathbf{x})$ and $M^{(2)}(\mathbf{x})$ directly [23, 24, 25, 26]: if two representations behave similarly over all of their elements, their similarity will be high even if no two individual elements have similar behaviors. In this work we focus on the latter case and adopt linear centered kernel alignment (`lincka`; [24]) and singular vector canonical correlation analysis (`svcca`; [25]) as our similarity measures. We choose these two since they are found to be comparable or better than other measures in prior studies for analyzing contextual word representation models [27].

2.2. Probing tasks for measuring phonetic and speaker content

We consider phonetic and speaker classification for measuring the amount of phonetic and speaker content contained in a representation [28, 29]. Given a self-supervised model M pre-trained on an unlabeled dataset D_1, we use M to extract features $M(\mathbf{x}) = (\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_T)$, where $\mathbf{m}_t \in \mathbb{R}^{512}$ for another dataset D_2, and train a linear classifier using the extracted features as input.

For phonetic classification, the goal is to predict the phone identity of each frame in an input utterance. For speaker classification, the extracted features are first averaged before being fed to the classifier, and the goal is to predict the speaker identity of the utterance. The frame-level phone error rate and utterance-level speaker error rate on the test set of D_2 indicate the amount of phonetic and speaker content contained in the representation, respectively.

3. EXPERIMENTAL SETUP

3.1. Self-supervised models

We consider some of the most representative models, including contrastive predictive coding (CPC) [4], autoregressive predictive coding (APC) [5], and masked predictive coding (MPC) [6, 7, 8].

While there are additional models that have successfully been applied to speech applications, most of them are more or less an extension of the above models. For example, Rivière et al. (2020) improved CPC by modifying its batch normalization mechanism

Table 1: Information about various implementations of APC, MPC, and CPC to be compared in this work. All RNN and Transformer models have a hidden size of 512 (256 for forward and 256 for backward if bidirectional). For CPC, `cpc-mixed_spk-rnn` draws negative samples across speakers, while `cpc-within_spk-rnn` and `cpc-within-spk-cnn` draw negative samples from the same utterance as the target future frame. All building blocks have 3 layers unless otherwise stated. TRF stands for Transformer.

Model notation	Objective	Building block	Uni- or Bi-dir
`apc-fw-rnn`	APC	GRU	Uni-dir
`apc-fw+bw-rnn`	APC	GRU	Bi-dir
`apc-fw-trf`	APC	TRF decoder	Uni-dir
`apc-fw+bw-trf`	APC	TRF decoder	Bi-dir
`mpc-birnn`	MPC	GRU	Bi-dir
`mpc-trf`	MPC	TRF encoder	Bi-dir
`cpc-mixed_spk-rnn`	CPC	GRU	Uni-dir
`cpc-within_spk-rnn`	CPC	GRU	Uni-dir
`cpc-within_spk-cnn`	CPC	Same as [30]	-

and replacing the linear prediction head with a 1-layer Transformer network [31]. Kawakami et al. (2020) made CPC bidirectional [32]. wav2vec [30] is CPC with a convolutional architecture. DeCoAR [12] could be viewed as a bidirectional version of APC. Chung and Glass (2020) proposed an auxiliary loss serving as a regularizer to help APC generalize better [33]. Liu et al. (2020) applied SpecAugment [34] to improve MPC's masking techniques [35]. Jiang et al. (2020) combined APC and MPC to form a unified pre-training objective [13]. We leave the explorations of these extensions for future work. Below we briefly review CPC, APC, and MPC.

CPC & APC Contrastive predictive coding (CPC) and autoregressive predictive coding (APC) share a similar methodology as both use an autoregressive model to learn representations through conditioning on the past context to make predictions of future information. Their main difference lies in the manner in which they optimize the autoregressive model: while APC attempts to predict a future frame via L1 regression, CPC incorporates a proposal distribution for drawing negative samples, and learns representations containing information that most discriminates the future frame from the negative samples using a loss based on noise-contrastive estimation [36]. We mainly follow the original papers [4, 5] for implementing the models with small modifications described in [5].

Since the objectives of APC and CPC are based on the notion of future prediction, bidirectional architectures are not applicable. A simple method for making these models have access to context from both directions is to separately train a forward and backward APC/CPC model and concatenate their output representations as the final representations (similar to how ELMo [37] is trained for learning contextualized word embeddings). This method has been explored for APC and CPC in [12] and [32], respectively.

MPC Inspired by the masked language modeling technique from BERT [2], masked predictive coding (MPC) directly trains a bidirectional architecture by first masking parts of the input signals and then predicting them through conditioning on context from both directions. Similar to APC, MPC is optimized by minimizing the framewise L1 distance between the predicted output and the original input before masking. Transformer encoder [6, 8] and bidirectional RNN [7] have both been used to implement MPC.

To account for multiple factors in model design (objective, RNN/Transformer/CNN, uni/bidirectional), we consider the implementations of APC, MPC, and CPC as listed in Table 1.

3.2. Pre-training datasets

We use LibriSpeech [38], which contains about 1k hours of speech audio, for pre-training all self-supervised models. We also use the `unlab-6k` subset from Libri-Light [39], which contains about 6k hours of speech audio, for additional experiments in Section 4.3. We use 80-dimensional log Mel spectrograms as input acoustic features, i.e., $\mathbf{x}_t \in \mathbb{R}^{80}$. All models are trained for 10 epochs using Adam with a batch size of 32 and a learning rate of 10^{-3}. During pre-training, only the speech portion from the dataset is used.

3.3. Probing datasets

Representation similarity measures For calculating representation similarity with `lincka` and `svcca` (introduced in Section 2.1), we use the `si284` subset from Wall Street Journal (WSJ) and the train set from TIMIT.

Phonetic and speaker classification We carry out both classification tasks on WSJ. For phonetic classification, there are a total of 42 phone categories, and we follow the standard split of WSJ, using 90% of `si284` for training, 10% for validation, and reporting frame-level phone error rate on `dev93`. The phone alignments are generated with a speaker adapted GMM-HMM model. For speaker classification, we follow [11] and consider a 259-class classification task where each class corresponds to an unique speaker, using 80% of `si284` for training, the other 10% for validation, and reporting utterance-level speaker error rate on the rest 10%. We note that speaker classification is not a typical task for WSJ, and only serves as a sanity check for the presence of speaker information. For both tasks, the classifier is a linear logistic regression trained for 10 epochs using SGD with a batch size of 32 and a fixed learning rate of 10^{-4}. All reported error rates are an average of 5 runs, of which variances are negligibly small and not included.

4. RESULTS AND ANALYSIS

4.1. Similarity of different self-supervised representations

Figure 1 shows the heatmap of similarities between representations learned by various self-supervised models. Brighter colors indicate higher similarity between two representations. We also include the similarity between each self-supervised representation and the surface feature, i.e., log Mel spectrogram. Due to space limit, we only show the result according to similarity measure `lincka` on WSJ.[1] We find all heatmaps exhibiting consistent patterns regardless of the probing dataset and similarity measure, and all self-supervised representations are very different from the surface feature. The heatmaps reveal the following insights.

Objective affects similarity more than architecture. The most evident pattern from the heatmaps is that there is always a greater similarity within an objective than across objectives, indicated by the bright block diagonal. For example, `apc-fw-rnn` is always more similar to `apc-fw+bw-rnn`, `apc-fw-trf`, and `apc-fw+bw-trf` than to any MPC and CPC variants, even when `apc-fw-rnn` and `cpc-mixed_spk-rnn/cpc-within_spk-rnn` share the same building block and directionality. This conclusion also holds for the MPC- and CPC-family. Representations learned by generative-based objectives, i.e., variants of APC and MPC, are also more similar to one another than to the CPC variants.

---
[1]More similarity heatmaps are available at `https://github.com/iamyuanchung/ICASSP21-Similarity-Supplementary`
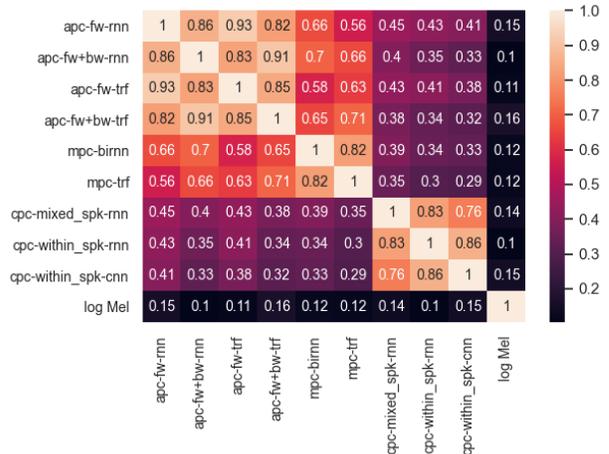
Fig. 1: Similarity heatmap of various self-supervised representations on WSJ according to `lincka`. Similarity values are also annotated.

Directionality affects similarity more than building block. When the objective is the same, we find that model's directionality (uni/bidirectional) has a higher impact on representation similarity than its building block (RNN/Transformer/CNN). For instance, the similarity between `apc-fw-rnn` and `apc-fw-trf`, which are both unidirectional while the former uses RNNs and the latter uses Transformers, is higher than that between `apc-fw-rnn` and `apc-fw+bw-rnn`, which both use RNNs while the former is unidirectional and the latter is bidirectional. Furthermore, as may be expected, making APC bidirectional reduces its difference with MPC, which is indicated by the fact that `mpc-birnn` is more similar to `apc-fw+bw-rnn` than to `apc-fw-rnn`, and `mpc-trf` is more similar to `apc-fw+bw-trf` than to `apc-fw-trf`.

Source of negative samples affects similarity more than architecture. When focusing on the CPC-family, we find that the proposal distribution of where the negative samples are drawn from is more impactful on representation similarity than the building block. This is indicated by the fact that `cpc-within_spk-rnn` is more similar to `cpc-within_spk-cnn` than to `cpc-mixed_spk-rnn`, where the two models in the former case share the same proposal distribution but use different building blocks, and the two models in the latter case share the same building block but incorporate different proposal distributions.

4.2. Correlation between self-supervised loss and phonetic & speaker classification performance

Experiments so far have only revealed the similarities between different self-supervised representations. We further uncover the correlation between self-supervised loss during pre-training and the amount of phonetic and speaker information contained in the representations, measured by their performance on phonetic and speaker classification (Section 2.2). We only consider `apc-fw-rnn`, `mpc-birnn`, `cpc-mixed_spk-rnn`, and `cpc-within_spk-rnn` in this experiment for a comparison only in terms of their objectives (except `mpc-birnn`, which has to be bidirectional). We calculate the Pearson correlation coefficients r between loss value and both phone and speaker error rates, as listed in Table 2.

Table 2: Pearson correlation coefficients between the self-supervised loss and the phone and speaker error rates. $*$ denotes statistical significance at $\rho < 0.05$.

Model	Phone	Speaker
`apc-fw-rnn`	0.989*	0.950*
`mpc-birnn`	0.885*	0.847*
`cpc-mixed_spk-rnn`	0.643*	0.762*
`cpc-within_spk-rnn`	0.675*	-0.071

Overall, APC and MPC are found to have a stronger correlation between the self-supervised loss and both their phonetic and speaker classification performance than CPC. In particular, `apc-fw-rnn` features the strongest correlation among the four considered self-supervised models. Our finding aligns with [21], where the autoregressive loss of APC is found to be more correlated with the ABX-score of a phone discrimination task than the InfoNCE loss of CPC.

It is noteworthy that the loss of `cpc-within_spk-rnn` has almost no correlation with speaker classification performance. This result seems natural since the model always draws negative samples from the same utterance as the positive sample, so speaker information is never found to be useful for distinguishing them and thus not learned by the representation. On the other hand, the proposal distribution of `cpc-mixed_spk-rnn` allows the model to learn from negative samples coming from both the same and different utterances as the positive sample, meaning that both phonetic and speaker information could be relevant for discriminating them. Therefore, we find the loss of `cpc-mixed_spk-rnn` is still correlated with the speaker error rate to some degree.

We emphasize that our findings here are not meant to claim any self-supervised approach to be the best, but aim to provide some results for other researchers for future reference. For example, APC and MPC's strong correlation between their self-supervised loss and phonetic and speaker classification performance could be useful for model selection even during the pre-training stage, since a lower pre-training loss would indicate a richer phonetic and speaker representation. CPC, though exhibiting a smaller correlation between its self-supervised loss and phonetic and speaker classification performance, could still be extremely powerful when the downstream task is known and thus the pre-training proposal distribution can be determined aforehand, as shown by its recent impressive performance on semi-supervised speech recognition [3].

4.3. Effect of increasing unlabeled data for pre-training

One of the biggest advantages of self-supervision is its capability to leverage large-scale unlabeled data for representation learning. Here we train `apc-fw-rnn`, `mpc-birnn`, `cpc-mixed_spk-rnn`, and `cpc-within_spk-rnn` on 2k, 4k, and 6k hours of speech audio, all sampled from the `unlab-6k` subset of the Libri-Light corpus, and calculate the similarities between each of these variants and their counterpart trained on the original 960 hours LibriSpeech audio according to `lincka`. Results are shown in Table 3.

For all models, their representations become more dissimilar when more data are used for pre-training. We also find that CPC's representations change more than those of APC and MPC when increasing the data size. For instance, the similarity "only" drops from 0.957 to 0.923 for `apc-fw-rnn` when increasing the data size from 2k hours to 6k hours, while for `cpc-mixed_spk-rnn`, the similarity drops from 0.911 to 0.837.

Table 3: Representation similarity between self-supervised models pre-trained on $\sim$1k hours of audio and their counterparts pre-trained on increasing amounts of audio according to `lincka`.

Model	Hours of pre-training audio		
	$\sim$2k	$\sim$4k	$\sim$6k
`apc-fw-rnn`	0.957	0.935	0.923
`mpc-birnn`	0.940	0.939	0.925
`cpc-mixed_spk-rnn`	0.911	0.883	0.837
`cpc-within_spk-rnn`	0.920	0.896	0.861

Changes in representation similarity can be attributed to encoding details of speech other than phonetic and speaker information that might be unnecessary, such as background noises. To confirm whether such changes in representation similarity correspond to an actual richer phonetic and speaker representation, we again use phonetic and speaker classification performance to quantify the amount of phonetic and speaker information contained in the representation.

Encouragingly, we observe that most self-supervised models' performance on both tasks is improved when being trained on more data. The only exception is `cpc-within_spk-rnn` on speaker classification, which is expected as speaker information is never found relevant for discriminating positive and negative samples during its training. However, its performance on phonetic classification obtains the largest gain among all considered self-supervised models. Concerning `cpc-mixed_spk-rnn`, in addition to showing improvement on both tasks, the drop of its speaker error rate is also the largest among all models. Intuitively, having more data means that CPC models are provided with more comparisons of negative and positive samples to learn from, and our results seem to suggest that this is a more effective way for learning representations when large amounts of unlabeled data are available, as opposed to attempting to reconstruct details of the speech signals as APC and MPC models do. That being said, both generative- and contrastive-based objectives also benefit from having more unlabeled training data.

5. CONCLUSIONS

We have analyzed representations learned by contrastive predictive coding (CPC), autoregressive predictive coding (APC), and masked predictive coding (MPC) through the lens of similarity analysis. Extensive experiments have been conducted to study the impact of different modeling choices for training self-supervised models, the effect of the size of unlabeled training data, and how well the self-supervised loss correlates with phonetic and speaker classification performance. We have found that the self-supervised objective has a much higher impact on representation similarity than architectural choices such as building blocks (RNN/Transformer/CNN) and directionality (uni/bidirectional). We have also observed that APC has the strongest correlation between its self-supervised loss and phonetic and speaker classification performance, which is useful for model selection. Finally, while all self-supervised models benefit from having more training data, CPC is found to learn from the additional data more efficiently than APC and MPC.

6. ACKNOWLEDGMENTS

7. REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," in *ICLR*, 2020.

[2] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional Transformersfor language understanding," in *NAACL-HLT*, 2019.

[3] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[4] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[5] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," in *Interspeech*, 2019.

[6] A. Liu, S.-W. Yang, P.-H. Chi, P.-C. Hsu, and H.-Y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional Transformer encoders," in *ICASSP*, 2020.

[7] W. Wang, Q. Tang, and K. Livescu, "Unsupervised pre-training of bidirectional speech encoders via masked reconstruction," in *ICASSP*, 2020.

[8] D. Jiang et al., "Improving Transformer-based speech recognition using unsupervised pre-training," *arXiv preprint arXiv:1910.09932*, 2019.

[9] S. Pascual, M. Ravanelli, J. Serra, A. Bonafonte, and Y. Bengio, "Learning problem-agnostic speech representations from multiple self-supervised tasks," in *Interspeech*, 2019.

[10] A. Baevski and A. Mohamed, "Effectiveness of self-supervised pre-training for ASR," in *ICASSP*, 2020.

[11] Y.-A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP*, 2020.

[12] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," in *ICASSP*, 2020.

[13] D. Jiang et al., "A further study of unsupervised pre-training for Transformer based speech recognition," *arXiv preprint arXiv:2005.09862*, 2020.

[14] X. Song et al., "Speech-XLNet: Unsupervised acoustic model pretraining for self-attention networks," in *Interspeech*, 2020.

[15] H. Nguyen, F. Bougares, N. Tomashenko, Y. Estève, and L. Besacier, "Investigating self-supervised pre-training for end-to-end speech translation," in *Interspeech*, 2020.

[16] A. Wu, C. Wang, J. Pino, and J. Gu, "Self-supervised representations improve end-to-end speech translation," in *Interspeech*, 2020.

[17] V. Ravi, R. Fan, A. Afshan, H. Lu, and A. Alwan, "Exploring the use of an unsupervised autoregressive model as a shared encoder for text-dependent speaker verification," in *Interspeech*, 2020.

[18] S. Feng and O. Scharenborg, "Unsupervised subword modeling using autoregressive pretraining and cross-lingual phone-aware modeling," in *Interspeech*, 2020.

[19] F. Kreuk, J. Keshet, and Y. Adi, "Self-supervised contrastive learning for unsupervised phoneme segmentation," in *Interspeech*, 2020.

[20] Y.-A. Chung, H. Tang, and J. Glass, "Vector-quantized autoregressive predictive coding," in *Interspeech*, 2020.

[21] M. A. C. Blandón and O. Räsänen, "Analysis of predictive coding models for phonemic representation learning in small datasets," in *ICML SAS Workshop*, 2020.

[22] A. Bau, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass, "Identifying and controlling important neurons in neural machine translation," in *ICLR*, 2019.

[23] N. Kriegeskorte, M. Mur, and P. Bandettini, "Representational similarity analysis-connecting the branches of systems neuroscience," *Frontiers in Systems Neuroscience*, vol. 2, pp. 4, 2008.

[24] S. Kornblith, M. Norouzi, H. Lee, and G. Hinton, "Similarity of neural network representations revisited," in *ICML*, 2019.

[25] M. Raghu, J. Gilmer, J. Yosinski, and J. Sohl-Dickstein, "SVCCA: Singular vector canonical correlation analysis for deep learning dynamics and interpretability," in *NIPS*, 2017.

[26] G. Andrew, R. Arora, J. Bilmes, and K. Livescu, "Deep canonical correlation analysis," in *ICML*, 2013.

[27] J. Wu, Y. Belinkov, H. Sajjad, N. Durrani, F. Dalvi, and J. Glass, "Similarity analysis of contextual word representation models," in *ACL*, 2020.

[28] Y. Belinkov and J. Glass, "Analyzing hidden representations in end-to-end automatic speech recognition systems," in *NIPS*, 2017.

[29] S. Wang, Y. Qian, and K. Yu, "What does the speaker embedding encode?," in *Interspeech*, 2017.

[30] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," in *Interspeech*, 2019.

[31] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP*, 2020.

[32] K. Kawakami, L. Wang, C. Dyer, P. Blunsom, and A. van den Oord, "Unsupervised learning of efficient and robust speech representations," 2020.

[33] Y.-A. Chung and J. Glass, "Improved speech representations with multi-target autoregressive predictive coding," in *ACL*, 2020.

[34] D. Park et al., "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Interspeech*, 2019.

[35] A. Liu, S.-W. Li, and H.-Y. Lee, "TERA: Self-supervised learning of Transformer encoder representation for speech," *arXiv preprint arXiv:2007.06028*, 2020.

[36] M. Gutmann and A. Hyvärinen, "Noise-contrastive estimation: A new estimation principle for unnormalized statistical models," in *AISTATS*, 2010.

[37] M. Peters et al., "Deep contextualized word representations," in *NAACL-HLT*, 2018.

[38] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[39] J. Kahn et al., "Libri-Light: A benchmark for ASR with limited or no supervision," in *ICASSP*, 2020.