INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Language Recognition Based on Unsupervised Pretrained Models

Haibin Yu[1*], *Jing Zhao*[1*], *Song Yang*[2], *Zhongqin Wu*[2], *Yuting Nie*[1], *Wei-Qiang Zhang*[1^]

[1]Beijing National Research Center for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]TAL Education, 18 Zhongguancun Avenue, Beijing 100080, China

{yhb18,zhaoj20,nyt19}@mails.tsinghua.edu.cn, {yangsong1,wuzhongqin}@tal.com, wqzhang@tsinghua.edu.cn

Abstract

Unsupervised pretrained models have been proven to rival or even outperform supervised systems in various speech recognition tasks. However, their performance for language recognition is still left to be explored. In this paper, we construct several language recognition systems based on existing unsupervised pretraining approaches, and explore their credibility and performance to learn high-level generalization of language. We discover that unsupervised pretrained models capture expressive and highly linear-separable features. With these representations, language recognition can perform well even when the classifiers are relatively simple or only a small amount of labeled data is available. Although linear classifiers are usable, neural nets with RNN structures improve the results. Meanwhile, unsupervised pretrained models are able to gain refined representations on audio frame level that are strongly coupled with the acoustic features of the input sequence. Therefore these features contain redundant information of speakers and channels with few relations to the identity of the language. This nature of unsupervised pretrained models causes a performance degradation in language recognition tasks on cross-channel tests.

Index Terms: language recognition, unsupervised pretrained model

1. Introduction

Language Recognition (LRE) is a branch of speech signal processing in rapid growth. The task of language identity recognition is to solve a multi-target classification problem: what natural language is spoken in the utterance. Feature extraction has always been the first and most focused topic in LRE tasks. Since a few decades ago, many algorithms focusing on LRE tasks have been proposed. They are roughly classified into two types: phonotactic approaches and acoustic-phonetic approaches [1].

In the methods of phonotactics, a speech sequence is converted to representations of phonemes typically, and models operating at phoneme level are referred to as phone recognizer. For example, hidden Markov models (HMMs) and universal phone recognition (UPR) are widely used to model phonemes[2, 3]. Another method to convert speech utterances, attribute recognizer, models speech sequences into attributes of relatively smaller speech units [4]. Afterwards, the relations in the token sequences are summarized with language models such as phone n-Gram models [5, 6] or vector space models [3].

In the methods of acoustic-phonetics, the acoustic features are extracted from speech sequences. Mel-frequency cepstral coefficients (MFCCs) [7] are designed to utilize the human auditory patterns, and work well to capture dynamic features in speech recognition. During the recent years, shifted delta cepstral (SDC) [8] and time-frequency cepstral (TFC) features [9] are evaluated to be more effective. Statistic modeling is another effective method deserved to be mentioned, which includes universal-background-based GMM (GMM-UBM) and its variations [10]. The acoustic features can also be modelled in vector space by support vector machines (SVMs) [11, 12].

Recently, deep learning methods have been adopted to language recognition [13]. Except supervised learning methods, an increasing number of works have focused on pretrained feature extraction and unsupervised representation learning. With similar architecture and training strategies in BERT [14], Mockingjay [15], Tera [16] and wav2vec series [17] have achieved promising results in speech recognition as well. However, there is almost no research on their application to language recognition.

In our work, we mainly investigate several unsupervised representation learning methods, Contrastive Predictive Coding (CPC) [18], Autoregressive Predictive Coding (APC) [19], Vector-Quantized APC (VQ-APC) [20] and Non-Autoregressive Predictive Coding (NPC) [21]. The purpose of our work is to evaluate the effectiveness of unsupervised pretrained models for the LRE task in different settings. First, we observe that all of these pretrained features can easily adapt to downstream classifiers for LRE task. Systems based on unsupervised pretrained models work well, though the performance shows slight difference as the structure of the classifier varies. Second, we find out that self-supervised pretraining can obtain expressive features, allowing downstream classifiers to be relatively simple or trained at a low cost of data.

2. Related work

2.1. Exploration of unsupervised features in ASR

Unsupervised pretrained models are applied to various downstream tasks in speech processing with generalized representations, for example phoneme classification, speaker recognition, and a variety of other tasks [19, 22, 23]. Apart from carrying out specific tasks, several works have investigated other properties of pretrained features [17]. For example, Riviére et al. have shown that CPC features are well transferable across different languages [24]. Other works have evaluated the performance of several unsupervised models compared with traditional feature extraction methods by conducting multiple downstream tasks at the same time [25, 26].

[*] Equal contribution.
[^] Corresponding author.

http://dx.doi.org/10.21437/Interspeech.2021-807

2.2. Recent works on feature extraction for LRE task

In the past few years, many feature extraction approaches for language recognition have been proposed, and we mainly pay attention to methods related to neural networks. Radek et al. train bottleneck neural networks to produce multilingual bottleneck features [27]. Snyder et al. propose x-vector for deep neural network training [28], which is applied to LRE task soon [29]. Recently, a BERT-like feature extractor trained in a multi-object weakly supervised fashion has been proposed, which shows advantages over MFCC in LRE task [30].

3. Method

In this section, we first briefly introduce the principles of all the mentioned predictive coding models proposed recently. More detailed information can be found in our references. Next, we demonstrate our practice of applying unsupervised features to LRE task.

3.1. Related unsupervised pretrained methods

3.1.1. Contrastive predictive coding (CPC) [18]

CPC is an unsupervised method based on contrastive loss. In detail, given a sequence enframed into T frames, at a current time step t, CPC encodes the input frame x_t with an encoder g_{enc}, i.e. $z_t = g_{\text{enc}}(x_t)$. After that, CPC applies an autoregressive model g_{ar}, summarizing $c_t = g_{\text{ar}}(z_1, \ldots, z_t)$ from the current frame and its history.

CPC model does not aim to predict the future inputs of the sequence x_{t+k} directly, therefore not optimizing the conditional probability $p(x_{t+k}|c_t)$. Instead, it is trained to optimize InfoNCE loss given by following equations [18]:

$$f(x_{t+k}, c_t) = \exp(z_{t+k}^{\mathrm{T}} W_k c_t) \tag{1}$$

$$\mathcal{L} = -\mathbb{E}\left\{\log \frac{f(x_{t+k}, c_t)}{\sum_{x_j \in N} f(x_j, c_t)}\right\} \tag{2}$$

where N is a set of "negative" samples composed of distant frames from t. x_{t+k} is a nearby sample whose representation is to be predicted. f is a log-bilinear model where W_k is a linear transform matrix, and $\mathcal{L}$ is the overall loss. It is proven by the original author of CPC that optimizing this loss is equivalent to lifting the lower bound of mutual information between x_{t+k} and x_t [18].

3.1.2. Autoregressive predictive coding (APC) [19]

APC is an unsupervised speech representation method which mainly focuses on predicting a sequence of spectrum in the future. Given the input sequence $(x_1, x_2, ..., x_T)$, the context network (for example, an RNN) is optimized to predict the frame ahead of the current x_t by minimizing the L1 loss between the input sequence and the predicted sequence $(y_1, y_2, ..., y_T)$:

$$\sum_{i=1}^{T-n} |x_{i+n} - y_i| \,. \tag{3}$$

The pretrained APC model should preserve information for a wide range of downstream tasks with enough generality.

3.1.3. Vector-quantized APC (VQ-APC) [20]

On the basis of APC, VQ-APC is introduced with additional vector quantization layer(s) between the context network layers. The discrete codebook variables are selected by Gumbel-Softmax in a fully differentiable way. Meantime, the training objective remains the same as APC. The quantization architecture obtains better representations with the amount of information controlled explicitly.

3.1.4. Non-autoregressive predictive coding (NPC) [21]

Different from the autoregressive model in APC or VQ-APC, NPC relies only on local dependencies of speech in a non-autoregressive manner to learn speech representations. Masked Convolution Blocks are utilized to implement some of the Masked Language Modeling (MLM) properties by reconstructing the masked frames. Specially, the high-level representations are restricted to observe certain unmasked frames containing no information about the corresponding masked input. So the method gets rid of incorporating global dependency, which leads to significant speedup for deriving speech representations. Besides, the representation extraction model is composed by stacked Convolution Blocks with Vector-Quantization. For the input sequence $(x_1, x_2, ..., x_T)$, the optimization objective is still L1 loss between the input and prediction sequence with minor differences:

$$\sum_{i=1}^{T} |y_i - x_i| \tag{4}$$

The NPC model performs comparably on a series of downstream tasks to the above architectures with more efficient inference.

3.2. Performing LRE task with pretrained models

Previous researches have shown that unsupervised pretrained features are effective for speaker recognition and phoneme classification. Unlike these two tasks, since language identity is dependent on the arrangement and composition of phonemes, LRE task should follow a high-level modelling principle.

During our experimental procedure, we first build unsupervised pretrained models with open-source datasets, and then use them to extract features on multi-lingual datasets for language recognition. Generally speaking, these pretrained features are sequences of vectors with fixed lengths.

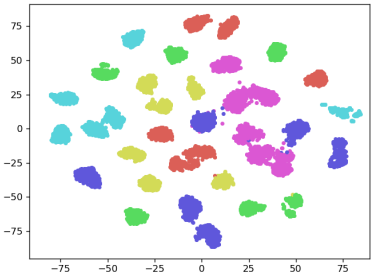

Figure 1: *T-SNE diagram of CPC features of AP20-OLR-channel-test*

Before designing concrete downstream classifiers, we reduce the dimension of the feature vectors, and observe their clustering results and separability qualitatively. More specifically speaking, we use the CPC model to extract the latent feature representations from AP20-OLR-channel-test dataset [31], and map them to 2-dimensional plane with the T-SNE tool [32]. The results are shown in figure 1, which illustrates the viability

of these features for language recognition. Each color stands for one single identity of language in the figure. We observe that the pretrained features are well separable after dimensionality reduction, providing evidence for their excellence in linear separability. Moreover, within a single class, the features are clustered into separate blocks, which indicates that the acoustic qualities irrelevant to the LRE task exist in these unsupervised features, such as speaker information.

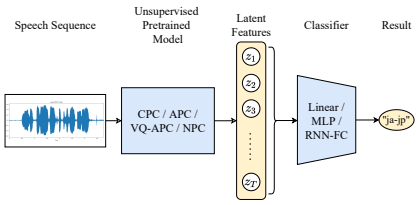

Figure 2: *Overall experimental system architecture*

To carry out the language recognition task that is almost not explored or presented systematically in the unsupervised pretraining field before, we develop multiple classifiers to handle these features. First considering the high separability of unsupervised features shown in the dimensional reduction visualization result, we are motivated to select simple linear classifiers in the following evaluations. Besides, we also explore several classification schemes apart from this. For example, multi-layer perceptrons (MLPs) are taken into account. Additionally, to take full advantage of pretrained features, we should utilize the time sequence relations argued previously. Therefore we employ a recurrent neural net (RNN) to build sequence models for LRE task, and linear layers are attached to the back end of sequence models.

All the classifiers are trained with unsupervised features. The detailed architectures of the classifiers as well as their corresponding performances with various pretrained architectures are described in the following sections.

4. Experiment setting

4.1. Data usage

In our experiments, we mainly use three sets of databases: LibriSpeech [33], CSS10[34], and AP-OLR datasets including AP16-OL7, AP17-OL3 and AP20-OLR-channel-test [31]. CSS10 consists of 10 languages spoken by 10 speakers respectively, with a total duration of about 140 hours. The combination of AP16-OL7 and AP17-OL3 consists of 10 oriental languages, and the duration of training data for each language is about 10 hours. CSS10 shares 3 languages (Japanese, Mandarin and Russian) with AP16-OL7 and AP17-OL3. AP-20-OLR-channel-test is composed of 6 languages within the 10 languages in AP16-OL7 and AP17-OL3, and the audios are recorded in unknown different channels.

4.2. Pretraining

We pretrain the CPC model on LibriSpeech train-clean-100 and CSS10. In our practice, the encoder is a convolutional network with 5 layers, and the auto-regressive model is a single layer LSTM, closely following the implementation by Facebook AI Research [24]. During the training process, negative samples are retrieved from other sequences spoken by the same speaker. Given one speech sequence, pretrained CPC model first splits it into windows of 20480 points, and cut each window to frames of 160 points, meaning that each feature vector is a latent embedding of 10ms of audio.

We build the following 3 models after the implementation [21] of Liu et al. For APC and VQ-APC, only train-clean-100 and train-clean-360 subsets in LibriSpeech are utilized in pretraining period. As for the model architecture, a 3-layer unidirectional GRU with hidden dimension of 512 is adopted as the feature extraction network in APC and VQ-APC. Besides, in VQ-APC the codebook size as well as the embedding dimension of each code is 512. During training, the model predicts a target frame that is 5 steps ahead the current one in the future.

The NPC model is trained on LibriSpeech, the train-clean-100 and train-clean-360 subsets as well. To keep up with the models mentioned earlier, the dimension of each layer in the convolution blocks is set to 512. The convolution blocks are 6 in total and the input mask size is 11 with a receptive field size of 37.

4.3. Downstream models

For all the 4 models mentioned above, a simple linear classifier is adhered on the bottom of the pretrained models, which means the extracted features are averaged over time extension and then fed into the classifier to predict the final results. We also explore another two classification modules: one is MLP and the other is RNN-FC pipeline. For the MLP scheme, we apply a 1-layer or 2-layer MLP with 256-dim before the linear layer. For the RNN-FC scheme, the representations are arranged by time, and then put into a 1-layer LSTM followed by 2 linear layers. We collect the hidden state on every time step, average them over time, and input to the linear layers for classification.

Our results are evaluated mainly by two metrics, the accuracy of language classification and the C-avg score related to false alarm rates and missing rates. We adopt the rule in OLR20 to compute the C-avg score [31].

5. Results

In our sets of experiments, we first compare the predictive coding models pretrained on LibriSpeech 100 with the x-vector with extended TDNN [31] and phone n-Gram-SVM [1] baseline systems on AP20-OLR-channel-test. All of the downstream classifiers are trained on randomly-divided 10% of AP20-OLR-channel-test, and evaluated on the remaining 90%.

Table 1: *Results on AP20, Libri100 pretrained*

Pretrained Model	Classifier	Accuracy	C-avg
CPC	Linear	0.9674	0.0192
NPC	Linear	0.9647	0.0194
APC	Linear	0.9657	0.0186
VQ-APC	Linear	0.9449	0.0313
NPC	MLP	0.9744	0.0145
APC	MLP	0.9732	0.0145
VQ-APC	MLP	0.9587	0.0225
CPC	RNN-FC	0.9923	0.0042
x-vector baseline		–	0.1279
n-Gram-SVM baseline		0.9268	0.0418

First, in Table 1, we compare the accuracy and C-avg of the CPC, APC, VQ-APC and NPC systems as well as the baseline systems. We are glad to find that systems based on pretrained models achieves significantly lower false alarm rates and miss-

ing rates, and the gaps between different pretrained models are relatively small. Note that all the classifiers are trained on a small amount of data. This indicates unsupervised features allow LRE classifiers to be trained at a relatively low cost.

Although representations extracted by unsupervised pretrained models show strong linear separability, non-linear factors still exist, which is suggested by the improved results generated by the MLP and RNN-FC models. The LSTM layer enables the downstream system to generalize the feature series over time, and MLP provides non-linear discriminant abilities.

Table 2: *Results on AP16 and AP17, Libri100 pretrained*

Pretrained Model	Classifier	Accuracy	C-avg
CPC	Linear	0.9606	0.0255
NPC	Linear	0.9408	0.0414
NPC	MLP	0.9680	0.0214
CPC	RNN-FC	0.9848	0.0104
x-vector baseline		–	0.2020
n-Gram-SVM baseline		0.8079	0.1200

With the full view of the results on AP20-OLR-channel-test, we then transfer our experiments to larger datasets. In the following set of experiments, we replace our downstream dataset to the combination of AP16-OL7 and AP17-OL3 while keeping the train-test data distribution style. The results are reported in Table 2. We find that the classifiers keep working well with features extracted by the same CPC and NPC models pretrained on mono-lingual datasets.

After these tests, we explore other behaviors of unsupervised pretrained features by altering pretraining and testing conditions in the following sets of experiments.

Table 3: *Results on AP16 and AP17, CSS10 pretrained*

Pretrained Model	Classifier	Accuracy	C-avg
CPC	Linear	0.9561	0.0291
CPC	RNN-FC	0.9820	0.0129

Here we replace the pretraining dataset to CSS10. The downstream classifiers are trained and tested on AP16-OL7 and AP17-OL3 as in the second set of experiments. We find an interesting result in Table 3 that multi-language pretrained CPC model does not have much inferiority to models pretrained on single language datasets. This result again shows that CPC features are transferable across languages.

Table 4: *Cross-Channel results on AP20, Libri100 pretrained*

Pretrained Model	Classifier	Accuracy	C-avg
APC	Linear	0.5329	0.2657
VQ-APC	Linear	0.5825	0.2376
NPC	Linear	0.5225	0.2719
APC	MLP	0.3439	0.3727
VQ-APC	MLP	0.4941	0.2900
NPC	MLP	0.4950	0.2881
x-vector baseline		–	0.2663

However, the performance of unsupervised pretrained models declines in cross-channel tests. In our next collection of experiments, we use APC, VQ-APC and NPC models pretrained on LibriSpeech train-clean-100 and train-clean-360, train and validate the classifiers on AP16-OL7 (except Mandarin), and test the systems on AP20-OLR-channel-test. As is compared in Tables 4 and 5, during our tests, expanding the range of pretraining data slightly improves the performance. Although the VQ-APC system has a small gap compared to other pretrained systems in previous evaluations, it obtains a leading role in cross-channel tests, while none of the systems reach as high performance as within-channel tests.

Table 5: *Cross-channel results on AP20, Libri360 pretrained*

Pretrained Model	Classifier	Accuracy	C-avg
APC	Linear	0.5571	0.2495
VQ-APC	Linear	0.5814	0.2360
NPC	Linear	0.5666	0.2466

Similarly, the performance also suffers a decline in our tests on a totally speaker-independent dataset we build from AP16-OL7 and AP17-OL3 with training data of about 39000 utterances and test data of about 10000. The best accuracy we gain is just below 92% with CPC finetuned model, which suggests the speaker-related factors remain in pretrained features, and the classifiers find them as language distinction to a certain extent.

Generally, the unsupervised pretrained models are designed to capture low-level features related to the acoustic qualities of speech sequences. By nature, these features are sensitive to audio and speaker conditions, and establish strong bonds to the classifier during the downstream training process. When transferring to unknown channels, mismatch occurs between pretrained features and classifier parameters, and therefore these systems achieve poor performance in cross-channel tests.

6. Conclusion

Unsupervised pretrained models can extract feature representations that are easily utilized in downstream language recognition tasks. Unsupervised features can be produced from a large range of unlabeled and accessible speech data, whether on mono-lingual datasets or multi-lingual ones. These features are able to transfer across languages and thus credible for language recognition. Furthermore, they show strong linear separability in our tests, which allows us to build simple classifiers and train them at a minimal cost of time and data consumption.

Though unsupervised pretrained features have an obvious tendency to show performance degradation in cross-channel tests, they are still promising means to perform LRE tasks. In this work, we basically explore the feasibility of this application to discriminating from a close set of natural languages, and obtain preliminary results from several classifiers. Future works are to find how the cross-channel and cross-domain robustness can be enhanced, and how the pretrained features can be exploited in a more effective way to provide solid solution to increasing the performance of downstream classifiers.

7. Acknowledgements

This work was supported by National Key R&D Program of China under Grant No. 2020AAA0104500, and in part supported by the National Natural Science Foundation of China under Grant No.U1836219.

8. References

[1] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[2] S. Nakagawa, Y. Ueda, and T. Seino, "Speaker-independent, text-independent language identification by hmm," in *Second International Conference on Spoken Language Processing*, 1992.

[3] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 271–284, 2006.

[4] S. M. Siniscalchi, J. Reed, T. Svendsen, and C.-H. Lee, "Exploring universal attribute characterization of spoken languages for spoken language recognition," in *Tenth Annual Conference of the International Speech Communication Association*, 2009.

[5] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling," in *Proceedings of ICASSP'94. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 1994, pp. I–305.

[6] J.-L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices," in *Eighth International Conference on Spoken Language Processing*, 2004.

[7] M. A. Zissman, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on speech and audio processing*, vol. 4, no. 1, p. 31, 1996.

[8] P. A. Torres-Carrasquillo, E. Singer, M. A. Kohler, R. J. Greene, D. A. Reynolds, and J. R. Deller Jr, "Approaches to language identification using gaussian mixture models and shifted delta cepstral features," in *Seventh international conference on spoken language processing*, 2002.

[9] W.-Q. Zhang, L. He, Y. Deng, J. Liu, and M. T. Johnson, "Time–frequency cepstral features and heteroscedastic linear discriminant analysis for language recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 266–276, 2010.

[10] E. Wong and S. Sridharan, "Methods to improve gaussian mixture model based language identification system," in *Seventh International Conference on Spoken Language Processing*, 2002.

[11] W. M. Campbell, J. P. Campbell, D. A. Reynolds, E. Singer, and P. A. Torres-Carrasquillo, "Support vector machines for speaker and language recognition," *Computer Speech & Language*, vol. 20, no. 2-3, pp. 210–229, 2006.

[12] W. Zhang, B. Li, D. Qu, and B. Wang, "Automatic language identification using support vector machines," in *2006 8th international Conference on Signal Processing*, vol. 1. IEEE, 2006.

[13] F. Richardson, D. Reynolds, and N. Dehak, "Deep neural network approaches to speaker and language recognition," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1671–1675, 2015.

[14] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jun. 2019.

[15] A. T. 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 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6419–6423.

[16] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *arXiv preprint arXiv:2007.06028*, 2020.

[17] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Proc. Interspeech 2019*, pp. 3465–3469, 2019.

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

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

[20] Y.-A. Chung, H. Tang, and J. Glass, "Vector-Quantized Autoregressive Predictive Coding," in *Proc. Interspeech 2020*, 2020, pp. 3760–3764.

[21] A. H. Liu, Y.-A. Chung, and J. Glass, "Non-autoregressive predictive coding for learning speech representations from local dependencies," *arXiv preprint arXiv:2011.00406*, 2020.

[22] Y. A. Chung and J. Glass, "Generative pre-training for speech with autoregressive predictive coding," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 3497–3501.

[23] R. Fan, A. Afshan, and A. Alwan, "Bi-apc: Bidirectional autoregressive predictive coding for unsupervised pre-training and its application to children's asr," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 7023–7027.

[24] M. Rivière, A. Joulin, P.-E. Mazaré, and E. Dupoux, "Unsupervised pretraining transfers well across languages," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 7414–7418.

[25] L. van Staden and H. Kamper, "A comparison of self-supervised speech representations as input features for unsupervised acoustic word embeddings," in *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021, pp. 927–934.

[26] M. A. C. Blandón and O. Räsänen, "Analysis of predictive coding models for phonemic representation learning in small datasets," *ICML Workshop on Self-supervision in Audio and Speech*, 2020.

[27] R. Fér, P. Matějka, F. Grézl, O. Plchot, and J. Černockỳ, "Multilingual bottleneck features for language recognition," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[28] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[29] D. Snyder, D. Garcia-Romero, A. McCree, G. Sell, D. Povey, and S. Khudanpur, "Spoken language recognition using x-vectors." in *Odyssey*, 2018, pp. 105–111.

[30] S. Ling, J. Salazar, Y. Liu, K. Kirchhoff, and A. Amazon, "Bertphone: Phonetically-aware encoder representations for utterance-level speaker and language recognition," in *Proc. Odyssey 2020 The Speaker and Language Recognition Workshop*, 2020, pp. 9–16.

[31] Z. Li, M. Zhao, Q. Hong, L. Li, Z. Tang, D. Wang, L. Song, and C. Yang, "Ap20-olr challenge: Three tasks and their baselines," in *2020 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*. IEEE, 2020, pp. 550–555.

[32] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of machine learning research*, vol. 9, no. 11, 2008.

[33] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP 2015 - 2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[34] K. Park and T. Mulc, "Css10: A collection of single speaker speech datasets for 10 languages," *Proc. Interspeech 2019*, pp. 1566–1570, 2019.