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

Analyzing Speaker Information in Self-Supervised Models to Improve Zero-Resource Speech Processing

Benjamin van Niekerk, Leanne Nortje, Matthew Baas, Herman Kamper

E&E Engineering, Stellenbosch University, South Africa

{benjamin.l.van.niekerk,nortjeleanne,matthew.baas}@gmail.com, kamperh@sun.ac.za

Abstract

Contrastive predictive coding (CPC) aims to learn representations of speech by distinguishing future observations from a set of negative examples. Previous work has shown that linear classifiers trained on CPC features can accurately predict speaker and phone labels. However, it is unclear how the features actually capture speaker and phonetic information, and whether it is possible to normalize out the irrelevant details (depending on the downstream task). In this paper, we first show that the per-utterance mean of CPC features captures speaker information to a large extent. Concretely, we find that comparing means performs well on a speaker verification task. Next, probing experiments show that standardizing the features effectively removes speaker information. Based on this observation, we propose a speaker normalization step to improve acoustic unit discovery using K-means clustering of CPC features. Finally, we show that a language model trained on the resulting units achieves some of the best results in the ZeroSpeech2021 Challenge.
Index Terms: unsupervised speech processing, self-supervised learning, acoustic unit discovery, spoken language modeling.

1. Introduction

A core goal of *zero-resource speech processing* is to develop methods that can learn robust representations of speech without supervision [1–4]. These representations can be used to bootstrap training in downstream speech systems and reduce requirements on labeled data [5–7]. While a range of self-supervised methods have been developed for speech [6–11], in this paper we focus on contrastive predictive coding (CPC) [12].

CPC models are trained to distinguish future observations from a set of negative examples. The idea is that to accurately identify future speech segments, the model must learn meaningful phonetic contrasts while being invariant to low-level details such as background noise. Recent studies [12, 13] show that separate linear classifiers trained on CPC features can accurately predict both speaker and phone categories. Features that capture either phonetic or speaker information can be useful depending on the downstream task. However, it is unclear whether we can disentangle or discard either component e.g. if speaker-invariance is required for a specific task.

In this paper, we investigate how speaker information is represented in CPC features. We qualitatively (Section 2) and quantitatively (Section 3) show that the per-utterance mean over CPC features captures a large degree of the speaker information. Based on this observation, we propose a simple speaker normalization step that effectively removes speaker information (Section 5). We then show that speaker normalization improves performance on two downstream tasks: acoustic unit discovery [13–16], and spoken language modeling [17, 18]. Specifically, we improve an acoustic unit discovery system based on K-means clustering of CPC features (Section 6) and show that an LSTM-based language model trained on the discovered units achieves some of the best scores in the ZeroSpeech2021 challenge [19] (Section 6.2).

Our speaker normalization approach is very simple, making it easy to incorporate into current and future CPC speech models.

2. Analysis of CPC features

2.1. Contrastive predictive coding

CPC models consist of two components: an encoder, and a context network. First, the encoder maps input audio into a sequence of embeddings $(z_1, \ldots, z_T)$. Next, the autoregressive context network summarizes the embeddings (up until time t) into a context vector c_t. Using this context, the model is trained to discriminate actual future embeddings from a set of negative examples drawn from other utterances. Specifically, we minimize the constrastive loss:

$$\mathcal{L}_t := -\frac{1}{M}\sum_{m=1}^{M}\log\left[\frac{\exp(z_{t+m}^{\intercal}W_m c_t)}{\sum_{\tilde{z}\in\mathcal{N}_{t,m}}\exp(\tilde{z}^{\intercal}W_m c_t)}\right],$$

where M is the prediction horizon, W_m is a linear classifier, and $\mathcal{N}_{t,m}$ is a set containing the negative examples along with the correct future embedding z_{t+m}.

In this paper, we use the CPC-big model from [17] trained on the LibriLight `unlab-6k` set [5]. The encoder consists of five convolutional layers each with 512 channels, kernel sizes $\langle 10, 8, 4, 4, 4\rangle$, and strides $\langle 5, 4, 2, 2, 2\rangle$. Given raw audio sampled at 16 kHz, the encoder extracts embeddings with a hop length of 10 ms. The context network is a stack of four LSTM layers with 512 hidden units each. Finally, the linear classifier W_m is replaced with a single-layer transformer. We use the outputs of the second LSTM layer as speech features since they gave the best ABX phone discrimination results in [17]. In the remainder of the paper we refer to these as the *CPC features*.

2.2. A visual exploration of the CPC features

Previous work [12, 13] has shown that CPC features capture both phonetic and speaker information. However, it is unclear how the representation structures this information. We hypothesize that the per-utterance mean of the features captures a large degree of the speaker information. This is reasonable under the assumption that speaker identity remains constant over an utterance with phonetic content varying over shorter time scales [20].

As a first step towards validating this hypothesis, we explore the CPC features using UMAP [21]. Figure 1(a) shows the per-utterance mean of CPC features for six speakers selected from the LibriSpeech `dev-clean` set [22]. The different speakers are clearly separated, showing that the mean does indeed capture speaker information. In Figure 1(b) we zoom in, visualizing the CPC features as individual frames for two speakers (colored

http://dx.doi.org/10.21437/Interspeech.2021-1182

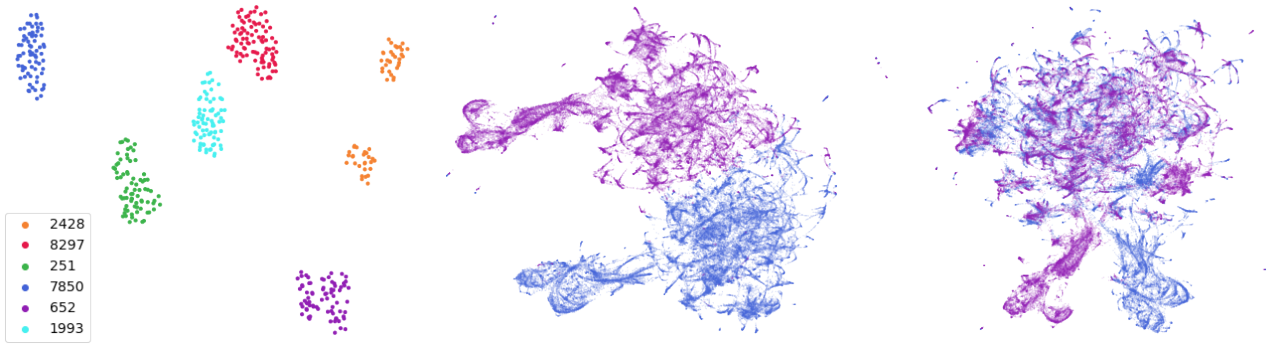

Figure 1: *UMAP visualizations of CPC features. (a) The per-utterance means of CPC features for six speakers. (b) Per-frame CPC features for the blue and purple speakersin (a). (c) Per-frame CPC features (standardized per utterance) for the same speakers.*

blue and purple). Although the UMAP embeddings for the two speakers exhibit similar structure, they are still separated based on speaker identity. This contrasts with Figure 1(c) which shows the same features after standardization, i.e. per-utterance mean and variance normalization of the CPC features. Here the structures are more aligned and no longer separated by speaker.

3. Speaker verification

In this section, we verify quantitatively that the per-utterance means of the CPC features capture speaker information. We show that simply comparing the means performs well on a speaker verification task. Given a set of enrollment utterances, the goal of speaker verification is to determine whether a new utterance belongs to a specific speaker. To set up the task, we randomly select five enrollment utterances for each speaker in the LibriSpeech `dev-clean` set, reserving the remainder for testing. We compare three systems across two metrics: classification accuracy and equal error rate (EER).

The first system is based on the means of the CPC features. In the enrollment step, we extract CPC features and compute the mean for each utterance. The means are then aggregated to find a single speaker embedding. At test time, we use Euclidean distance to compare the CPC feature mean of an utterance to the reference speaker embeddings. For classification accuracy, we select the closest speaker as the prediction. For EER, we threshold the distance to decide if the test utterance matches a given speaker.

The second system is a naive baseline that follows the same approach, but uses Mel-frequency cepstral coefficients (MFFCs) instead of CPC features. This system should provide a lower bound on the performance of the CPC-based approach.

The third system is a supervised topline based on the GE2E loss [23]. We use an open-source implementation trained on more than 8k speakers.[1] This system was specifically trained for speaker verification using a discriminative loss on a much larger dataset. Therefore it serves as an upper bound on expected performance.

Table 1 shows the results for the three approaches. The CPC-based system clearly outperforms the baseline. While there is a gap in performance compared to the topline, our goal was to demonstrate that the per-utterance mean of CPC features results in discriminative speaker embeddings.

[1] `https://github.com/resemble-ai/Resemblyzer`

Table 1: *Speaker verification results for the supervised topline and the CPC- and MFCC-based systems.*

	EER (%)	Accuracy (%)
Topline: GE2E	1.6	98.8
Proposed: Mean of CPC	6.7	95.8
Baseline: Mean of MFCCs	19.8	59.8

4. Speaker normalization

Based on the above observations, we propose standardizing the CPC features as a simple speaker normalization step. Given an utterance (or set of utterances) from a single speaker, we remove speaker information from the CPC features by subtracting the mean and scaling to unit variance. In the remainder of the paper, we analyze this speaker normalization step and apply it to two downstream tasks: acoustic unit discovery, and spoken language modeling (see Figure 2).

4.1. Acoustic unit discovery

In contrast to continuous representation learning, acoustic unit discovery involves finding a set of discrete units corresponding to the phonetic inventory of a language [4, 15]. We incorporate speaker normalization into a baseline acoustic unit discovery system built on K-means clustering applied to CPC features [17]. Concretely, we cluster the speaker-normalized features using K-means with 50 clusters. The cluster means are estimated on speech from a subset of 35 speakers in the LibriSpeech `train-clean-100` set.

4.2. Spoken language modeling

To determine whether the discovered acoustic units capture structure beyond just the acoustics, we consider the task of spoken language modeling [17, 18]. For this task, we train an LSTM language model on the units (top of Figure 2). We use the language model architecture of [17]: three LSTM layers each with 1024 hidden units. We train for 100k steps with a batch size of 32k acoustic unit tokens. To evaluate the quality of the language model, we use several metrics developed specifically for the goal of language modeling on speech (Section 6.2). Most of these metrics rely on a score for how probable a spoken segment is under the language model. To score a spoken segment, we first encode it into a sequence of acoustic units $(q_1, \ldots, q_N)$, based on the trained K-means model. We then compute the log probability of the sequence using the chain rule, $\log P(q_1, \ldots, q_N) = \sum_{k=1}^{N} \log P(q_k|q_1, \ldots, q_{k-1})$, where $P(q_k|q_1, \ldots, q_{k-1})$ is the output of the LSTM.

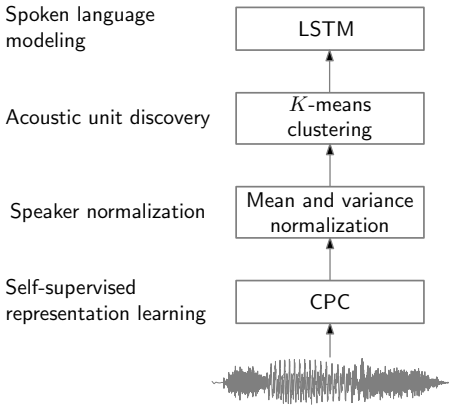

Figure 2: *We propose a speaker normalization method for CPC features. We incorporate speaker normalization into an acoustic unit discovery system (based on K-means clustering) and spoken language model (trained on the clustered codes).*

5. Probing experiments

To evaluate the speaker normalization step and further analyze the information captured by the CPC features, we conduct a series of probing experiments. We train a set of classifiers to predict either phone, speaker, or gender labels given a single CPC feature frame as input. The classifiers are trained on the `dev-clean` subset of LibriSpeech with classification accuracy reported on a held-out set of 400 utterances (10 per speaker). For the phone classifier, we use the Montreal Forced Aligner [24] to extract time-aligned phone labels (from 41 phone classes).

We first train a set of *linear* classifiers. The results (averaged over 10 runs) are shown at the top part of Table 2. These classifiers assess the degree to which phone, speaker, and gender classes are linearly separable. In the first row we repeat the experiments in [12, 13], showing that CPC features linearly separate phonetic, gender and speaker information. The second row shows that standardizing the features degrades speaker and gender classification accuracy, but slightly improves phone classification. This is reasonable given our observation that the mean captures speaker identity.

How does standardization affect results for *clustered* CPC features? This question is of interest for acoustic unit discovery (see Section 6). In the third and fourth rows of Table 2, we repeat the linear classifier experiments but on clustered features (using K-means with 50 clusters). We see that the clustering step reduces accuracy across all tasks. In particular, speaker and gender accuracy is reduced almost to the level of chance (2.5% and 50% respectively). This shows that the clusters primarily capture phonetic information with standardization improving performance.

The experiments above investigated linear separability, but to what extent do the features capture phone, speaker and gender properties more generally? To answer this question, we train non-linear classifiers on the CPC features. Specifically, we use multi-layer perceptrons with one hidden ReLU-layer containing 1024 units. Results are shown at the bottom of Table 2. Overall the non-linear classification scores are higher than their linear counterparts. However, standardization still reduces speaker classification accuracy (by over 10% absolute). On the other hand, phone and gender accuracies remain similar. This indicates that while standardization removes speaker and gender information in the linear case, these characteristics are still present in the features (albeit non-linearly).

Table 2: *Probing experiments where phone, speaker and gender classifiers are trained on CPC features. Clustering is performed on the CPC features using K-means with 50 clusters.*

		Accuracy (%)		
Standardized	Clustered	Phone	Speaker	Gender
Linear classifiers:				
✗	✗	75.7	93.4	96.7
✓	✗	77.0	14.8	55.3
✗	✓	46.6	3.4	53.5
✓	✓	48.5	3.1	50.9
Non-linear classifiers:				
✗	✗	80.1	99.5	99.8
✓	✗	79.7	89.0	98.1

6. Results on downstream tasks

6.1. Acoustic unit discovery

ABX phone discrimination tests. In this section, we use ABX phone discrimination tests [25] to evaluate the acoustic unit discovery system. These tests ask whether triphone X is more similar to triphone A or B. Here A and X are instances of the same triphone (e.g. "beg"), while B differs in the middle phone (e.g. "bag"). For the *within-speaker* test, A, B, and X, are all taken from the same speaker. The *across-speaker* test aims to measure speaker-invariance by taking A and B from the same speaker, but X from a different speaker. ABX is reported as an aggregated error rate over pairs of triphones. For the similarity metric between encoded segments, we use the average cosine distance along the dynamic time warping alignment path.

Table 3 shows ABX results on the `dev-clean` and `dev-other` subsets of LibriSpeech. Without clustering (rows one and two), speaker normalization slightly improves ABX scores. For the clustered CPC features (rows three and four), ABX is performed over one-hot encoded cluster codes. In this case, speaker normalization improves ABX by more than 13% relative on both the within and across speaker tests.

Clustering metrics. To further analyze the discovered acoustic units, we compute four metrics used to evaluate clustering quality. By mapping each unit to the overlapping phone label in the forced alignment, we evaluate the clustering quality in terms of the adjusted rand index (ARI), adjusted mutual information (AMI) [26], homogeneity, and completeness [27]. All these metrics are in the range $[0, 1]$, where higher is better.

Table 4 shows the results on the `dev-clean` subset of LibriSpeech. Standardization gives consistent improvements across the metrics. However, the clustering scores are relatively low

Table 3: *ABX error rates for CPC features and MFCCs.*

		Within (%)		Across (%)	
Standardized	Clustered	clean	other	clean	other
CPC features:					
✗	✗	3.41	4.85	4.18	7.64
✓	✗	3.41	4.81	4.12	7.49
✗	✓	6.38	10.22	8.26	14.86
✓	✓	5.38	8.80	6.56	12.79
Baseline: MFCCs		10.95	13.55	20.94	29.4

Table 4: *Clustering metrics calculated on the K-means clustered CPC features, with and without prior standardization.*

Standardized	ARI	AMI	Homogeneity	Completeness
✗	0.221	0.450	0.477	0.425
✓	0.255	0.488	0.517	0.462

overall. This indicates that despite good phone discrimination scores, there is still a large gap between the discovered acoustic units and the ground-truth phonetic transcriptions.

Number of clusters. Next, we study the effect of the number of clusters on ABX score. Table 5 reports ABX score (averaged over `dev-clean` and `dev-other`) for different numbers of clusters. In contrast to the findings in [17] (where 50 clusters gave the best results), we observe that increasing the number of clusters can improve ABX error rates.

Feature selection. Finally, we investigate feature selection to improve acoustic unit discovery. The idea is that speaker information might primarily be captured in a few specific dimensions of the CPC features. To test this, we train a random forest to predict speaker labels for each frame of the CPC features. We then prune the dimensions according to their importance ranking i.e., removing the dimensions that are most predictive of the speaker first. Table 6 shows ABX results (averaged over `dev-clean` and `dev-other`) as a function of the number of retained dimensions. We can see that ABX scores improve while pruning up to half of the feature dimensions.

6.2. Spoken language modeling

Lexical: Spot-the-word. To evaluate language models at the lexical level, we use the spot-the-word task from [30]. In this task, models are presented with pairs comprising of an existing word and a similar non-word (e.g., "brick" and "blick"). The goal is to distinguish the word from the non-word by assigning it a higher probability. An average classification accuracy is calculated over all word/non-word pairs. Table 7 reports spot-the-word results on the sWUGGY [17] test set. The set consists of 40k word/non-word pairs, generated using WUGGY [31] and synthesized using Google Cloud Text-to-Speech.

Syntactic: Acceptability judgments. At the syntactic level, we use grammar acceptability judgments to test the language models. This is similar to the spot-the-word task, but the goal is to distinguish grammatical from ungrammatical sentences (for example, "the dogs eat meat" versus "the dogs eats meat"). Table 7 reports classification accuracy on sBLIMP [17], a spoken version of the BLIMP [32] benchmark. The sBLIMP test set consists of 64k sentence pairs covering 12 grammar categories, e.g. anaphor agreement, island effects, and subject-verb agreement.

Table 5: *ABX results for different numbers of K-means clusters.*

# clusters	50	100	150	200
Within	7.09	6.73	6.74	**6.68**
Across	9.68	9.15	9.2	**9.09**

Table 6: *ABX results after pruning the CPC dimensions that are least informative for predicting speaker.*

# features	64	128	192	256	320	384	512
Within	8.88	7.74	7.05	6.88	**6.79**	7.04	7.09
Across	11.90	10.62	9.97	9.64	**9.44**	9.49	9.68

Table 7: *Results on the lexical, syntactic, and semantic spoken language modeling tasks.*

			Semantic	
	Lexical	Syntactic	Synth.	Libri.
Topline:				
Forced Align	92	63	8.5	2.4
Phone	**98**	67	12.2	20.2
RoBERTa	96	**82**	**33.2**	**27.8**
High budget:				
BERT baseline	68	56	6.3	2.5
Low budget:				
LSTM baseline	61	53	7.4	**2.4**
LSTM speaker-norm	**65**	**54**	**9.2**	-1.1
Chorowski et al. [28]	64	53	5.2	-0.9
Maekaku et al. [29]	61	**54**	7.0	-1.2

Semantic: Similarity judgments. We use human similarity judgments between word pairs to assess the semantic information captured by the language models. First, human evaluators score pairs of words (e.g. "abduct" and "kidnap") based on their semantic similarity. Next, we extract a fixed-dimensional representation for each word by pooling the outputs of a hidden layer of the language model. Specifically, we follow [17] by applying min-pooling to the outputs of the second LSTM layer. Finally, we compute the cosine similarity between the two representations and evaluate how well it compares to the human similarity scores with results reported as the Spearman's rank correlation coefficient. Table 7 reports semantic similarity scores on the sSIMI benchmark [17], a combination of 13 existing semantic similarity and relatedness tests including both synthetic and natural speech.

Results summary. In the bottom section of Table 7, we compare our approach (LSTM speaker-norm) to the three low-budget models submitted to the ZeroSpeech2021 challenge [17, 28, 29]. In the low-budget category, the LSTM language model trained on clustered speaker normalized CPC features scores the best on the lexical and syntactic tasks. This shows that better ABX scores (through speaker normalization) translate into better results on spoken language modeling. However, there remains a large gap in performance compared to the supervised topline systems. While the toplines score well on the lexical task, the syntactic and semantic results show that there is still room for improvement (despite access to ground-truth transcriptions). This suggests that syllable- or word-like units may be required for these tasks. Finally, while our approach doesn't match the performance of the high-budget BERT baseline, we expect the speaker normalization step to benefit this model as well.

7. Conclusion

We proposed a simple speaker normalization method for contrastive predictive coding (CPC) models. By analyzing a CPC model, we found that speaker information is largely captured by the per-utterance mean of the features. Based on this observation, we showed that standardizing the features effectively removes speaker details. We incorporated this speaker normalization step into systems for acoustic unit discovery and spoken language modeling, improving the ZeroSpeech 2021 Challenge baselines.

Acknowledgements. This work is supported in part by the South African NRF (grant no. 120409), a Google PhD Scholarship for BvN, a DeepMind Scholarship for LN, and a Google Faculty Award for HK.

8. References

[1] A. Jansen, E. Dupoux, S. J. Goldwater, M. Johnson, S. Khudanpur, K. Church, N. Feldman, H. Hermansky, F. Metze, R. Rose, M. Seltzer, P. Clark, I. McGraw, B. Varadarajan, E. Bennett, B. Borschinger, J. Chiu, E. Dunbar, A. Fourtassi, D. Harwath, C.-y. Lee, K. Levin, A. Norouzian, V. Peddinti, R. Richardson, T. Schatz, and S. Thomas, "A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition," in *Proc. ICASSP*, 2013.

[2] E. Dunbar, X. N. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The Zero Resource Speech Challenge 2017," in *Proc. ASRU*, 2017.

[3] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black *et al.*, "The Zero Resource Speech Challenge 2019: TTS without T," in *Proc. Interspeech*, 2019.

[4] E. Dunbar, J. Karadayi, M. Bernard, X.-N. Cao, R. Algayres, L. Ondel, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2020: Discovering discrete subword and word units," in *Proc. Interspeech*, 2020.

[5] J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P.-E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen *et al.*, "Libri-light: A benchmark for asr with limited or no supervision," in *Proc. ICASSP*, 2020.

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

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

[8] M. Heck, S. Sakti, and S. Nakamura, "Feature optimized DPGMM clustering for unsupervised subword modeling: A contribution to ZeroSpeech 2017," in *Proc. ASRU*, 2017.

[9] A. Baevski, S. Schneider, and M. Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *Proc. ICLR*, 2020.

[10] J. Chorowski, R. J. Weiss, S. Bengio, and A. van den Oord, "Unsupervised speech representation learning using WaveNet autoencoders," *IEEE Trans. Audio, Speech, Language Process.*, vol. 27, no. 12, pp. 2041–2053, 2019.

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

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

[13] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," in *Proc. Interspeech*, 2020.

[14] B. Varadarajan, S. Khudanpur, and E. Dupoux, "Unsupervised learning of acoustic sub-word units," in *Proc. ACL*, 2008.

[15] C.-y. Lee and J. R. Glass, "A nonparametric Bayesian approach to acoustic model discovery," in *Proc. ACL*, 2012.

[16] L. Ondel, L. Burget, and J. Černockỳ, "Variational inference for acoustic unit discovery," *Procedia Comput. Sci.*, vol. 81, pp. 80–86, 2016.

[17] T. A. Nguyen, M. de Seyssel, P. Rozé, M. Rivière, E. Kharitonov, A. Baevski, E. Dunbar, and E. Dupoux, "The zero resource speech benchmark 2021: metrics and baselines for unsupervised spoken language modeling," in *arXiv preprint arXiv:2011.11588*, 2020.

[18] K. Lakhotia, E. Kharitonov, W.-N. Hsu, Y. Adi, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed *et al.*, "Generative spoken language modeling from raw audio," *arXiv preprint arXiv:2102.01192*, 2021.

[19] E. Dunbar, M. Bernard, N. Hamilakis, T. A. Nguyen, M. de Seyssel, P. Rozé, M. Riviere, E. Kharitonov, and E. Dupoux, "The interspeech zero resource speech challenge 2021: Spoken language modelling," in *Proc. Interspeech*, 2021.

[20] W.-N. Hsu, H. Tang, and J. Glass, "Unsupervised adaptation with interpretable disentangled representations for distant conversational speech recognition," in *Proc. Interspeech*, 2018.

[21] L. McInnes, J. Healy, N. Saul, and L. Großberger, "Umap: Uniform manifold approximation and projection," *Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[22] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015.

[23] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," *arXiv preprint arXiv:1710.10467*, 2020.

[24] M. McAuliffe, M. Socolof, S. Mihuc, M. Wagner, and M. Sonderegger, "Montreal forced aligner: trainable text-speech alignment using kaldi," in *Proc. Interspeech*, 2017.

[25] T. Schatz, V. Peddinti, F. Bach, A. Jansen, H. Hermansky, and E. Dupoux, "Evaluating speech features with the minimal-pair ABX task: Analysis of the classical MFC/PLP pipeline," in *Proc. Interspeech*, 2013.

[26] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Is a correction for chance necessary?" in *Proc. ICML*, 2009.

[27] A. Rosenberg and J. Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure," in *Proc. EMNLP-CoNLL*, 2007.

[28] J. Chorowski, G. Ciesielski, J. Dzikowski, A. Lancucki, R. Marxer, M. Opala, P. Pusz, P. Rychlikowski, and M. Stypułkowski, "Information retrieval for zerospeech 2021 the submission by university of wroclaw," in *Proc. Interspeech*, 2021.

[29] T. Maekaku, X. Chang, Y. Fujita, L.-W. Chen, S. Watanabe, and A. Rudnicky, "Speech representation learning combining conformer cpc with deep cluster for the zerospeech challenge 2021," in *Proc. Interspeech*, 2021.

[30] G. Le Godais, T. Linzen, and E. Dupoux, "Comparing character-level neural language models using a lexical decision task," in *EACL*, 2017.

[31] E. Keuleers and M. Brysbaert, "Wuggy: A multilingual pseudoword generator," *Behavior Research Methods*, 2010.

[32] A. Warstadt, A. Parrish, H. Liu, A. Mohananey, W. Peng, S.-F. Wang, and S. R. Bowman, "BLiMP: The benchmark of linguistic minimal pairs for english," *TACL*, 2020.