CONTRASTIVE PREDICTION STRATEGIES FOR UNSUPERVISED SEGMENTATION AND CATEGORIZATION OF PHONEMES AND WORDS

Santiago Cuervo[1], *Maciej Grabias*[1], *Jan Chorowski*[2], *Grzegorz Ciesielski*[1]
Adrian Łańcucki[3], *Paweł Rychlikowski*[1], *Ricard Marxer*[4]

[1] University of Wroclaw, Poland [2]NavAlgo, France [3]NVIDIA, Poland
[4] Université de Toulon, Aix Marseille Univ, CNRS, LIS, Toulon, France

ABSTRACT

We identify a performance trade-off between the tasks of phoneme categorization and phoneme and word segmentation in several self-supervised learning algorithms based on Contrastive Predictive Coding (CPC). Our experiments suggest that context building networks, albeit necessary for high performance on categorization tasks, harm segmentation performance by causing a temporal shift on the learned representations. Aiming to tackle this trade-off, we take inspiration from the leading approaches on segmentation and propose multi-level Aligned CPC (mACPC). It builds on Aligned CPC (ACPC), a variant of CPC which exhibits the best performance on categorization tasks, and incorporates multi-level modeling and optimization for detection of spectral changes. Our methods improve in all tested categorization metrics and achieve state-of-the-art performance in word segmentation.

Index Terms— self-supervised learning, Contrastive Predictive Coding, unsupervised phoneme segmentation, unsupervised word segmentation, phoneme classification

1. INTRODUCTION

Speech self-supervised learning (SSL) without linguistic labels is aimed at producing representations that are useful for downstream problems such as transcription, classification or understanding. The prior work focuses mainly on either automatic boundary detection of phonemes or words [1, 2, 3], or learning representations which expose phonemic information [4, 5], and facilitate phoneme prediction with linear transformations. Even though the tasks seem related, an approach which performs well on the former may do poorly on the latter. An extreme example is an encoding that alternates between only two labels at every phone change. This representation has the full information about the boundaries, yet no information about the sequence of phonemes. On the other hand, we may imagine an encoding with no abrupt changes at phoneme boundaries, in which every frame maps to the correct phoneme through some unknown linear projection.

Contrastive Predicting Coding (CPC) [4] and its variants are popular methods of approaching these tasks. CPC is an SSL algorithm which extracts latent representations from sequential data by learning to predict future states of the model. An encoder g_{enc} maps consecutive overlapping chunks of data to latent representations z, producing sequences of codes. An autoregressive model g_{ar} is then applied to the latent representations and trained to predict M upcoming latents. The model is trained using Noise-Contrastive Estimation (NCE): the prediction p_t of the latent code z_t at time t must be closer to z_t than to randomly sampled latent codes, termed negative samples. When applied to speech, CPC produces acoustic representations which are useful for phoneme prediction and low-resource speech recognition.

Different variations of CPC have been proposed in the literature and have shown improvements on various downstream tasks. Chorowski et al. [5] presented Aligned CPC (ACPC), in which rather than producing individual predictions for each future representation, the model emits a sequence of $K < M$ predictions which are aligned to the M upcoming representations. In this way, g_{ar} solves a simpler task of predicting the next symbols, but not their exact timing, while g_{enc} is incentivized to produce piece-wise constant latent codes. ACPC exhibits higher linear phone prediction accuracy and lower ABX error rates than CPC, while being slightly faster to train due to the reduced number of prediction heads.

CPC-based techniques have also been applied to unsupervised phoneme and word segmentation. Kreuk et al. [1] proposed a model that omits the prediction network g_{ar} and is instead trained to discriminate between adjacent and non-adjacent representations. The model learns to detect spectral changes in the signal and tends to produce piece-wise constant latent codes. Boundaries are obtained as peaks in cosine dissimilarity between consecutive latent representations.

Segmental CPC (SCPC) [2] improves upon [1] by adding a standard CPC feature extractor with $M=1$ to model the signal at the level of segments of frames. A differentiable imple-

The authors thank the Polish National Science Center for funding under the OPUS-18 2019/35/B/ST6/04379 grant and the PlGrid consortium for computational resources. We also thank the French National Research Agency for their support through the ANR-20-CE23-0012-01 (MIM) grant.

mentation of the boundary detector used by [1] is applied to the frame-level latent representations and those within boundaries are averaged to produce a sequence of segment representations that is fed as input to another feature extractor. The segment-level feature extractor is meant to operate roughly at phoneme level and act as a language model. At test time the boundary detector from [1] is used at the frame-level to predict phoneme boundaries and at the segment level for word boundaries. SCPC reported state-of-the-art performance in both phone and word unsupervised segmentation.

We investigate the performance of ACPC on phoneme segmentation, and of [1] and SCPC on phoneme classification accuracy and ABX. The results suggest a trade-off between segmentation and categorization performance. We then explore the variations to the standard CPC model to understand the causes of this conflict. We also propose a multi-level ACPC model aiming to obtain gains in segmentation performance similar to SCPC [1], and explore the effect of multi-level modelling on phoneme classification and the ABX task. Finally, we show that including an auxiliary contrastive loss between adjacent frames as in [1] in ACPC models consistently improves segmentation and categorization performance.

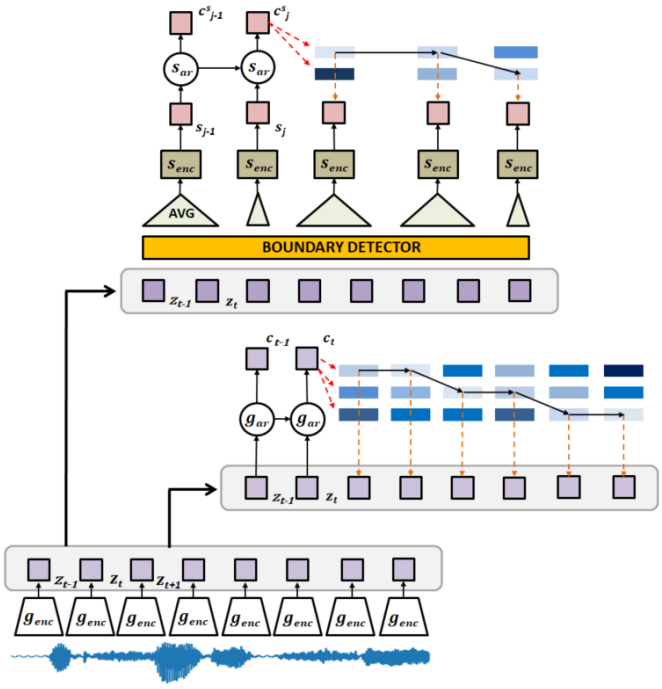

Fig. 1. The mACPC model has two main modules: frame-level and segment level. The frame-level module works on raw waveforms and extracts latent representations. These are processed by the boundary detector, which predicts boundaries and averages latents within those boundaries to produce segment representations. Finally, the segment-level module learns to predict higher-level features.

2. AUGMENTING ACPC FOR SEGMENTATION

We propose multi-level ACPC (mACPC; Fig. 1), which extends the ACPC architecture with a second ACPC feature extractor to model the signal at the level of frame segments, similarly to [2]. At the frame level, a strided convolutional encoder g_{enc} maps the input sequence $x_1, \ldots, x_T$ to a sequence of encoded frames $z_1, \ldots, z_{T'}$. The encoded frames are used to determine the position of the segment boundaries. Following [1], we calculate the score for placing a boundary at the position t as $-\text{sim}(z_{t-1}, z_t)$, where $\text{sim}(\cdot)$ denotes cosine similarity. Peaks in the dissimilarity scores indicate boundaries. Frames within two consecutive boundaries are considered as segments and their constituents are averaged. We obtain the final segment representations $s_1, \ldots, s_J$ by feeding the averages to a segment-level encoder s_{enc}. While the frame-level auto-regressive model g_{ar} summarizes all encoded frames up to time t into a context vector $c_t = g_{ar}(z_{\leq t})$, the auto-regressive model s_{ar} does it at the segment level $c^{s_j} = s_{ar}(s_{\leq j})$. Finally, K and K^s predictions are made at frame and segment levels conditioned on the corresponding context vectors, which are then aligned to M and M^s upcoming encoded frames and segments respectively. The ACPC prediction loss, as described in [5], is applied at both levels. The two prediction losses from frames and segments are summed into the total loss to be optimized.

Additionally, we also consider variations of ACPC and mACPC in which we add to their total loss the contrastive loss between adjacent representations proposed by Kreuk et al. in [1] to optimize for detection of spectral changes in the signal. The loss is applied to the output of g_{enc}.

3. EXPERIMENTS

We perform two phone classification evaluations: frame-wise classification, and an alignment-insensitive evaluation using Connectionist Temporal Classification (CTC) [6]. The classifiers are trained on the LibriSpeech train-clean-100 dataset [7] for 10 epochs. For the frame-wise case, a linear classifier is optimized with a cross-entropy loss and we report accuracy. The model used for the CTC evaluation is a single-layer bidirectional LSTM network with 256 hidden units, followed by a 1D convolution with 256 filters, kernel width 8 and stride of 4, trained using the CTC loss in which emission of the blank character is forbidden to force classifying each frame as a phoneme. Performance is evaluated using Phoneme Error Rate (PER).

Phoneme segmentation experiments are run on both TIMIT [8] and Buckeye [9]. Word segmentation is only run on Buckeye as in [2]. Segmentation quality was measured with precision, recall, F1-score and over-segmentation robust R-value [10], with a 20ms tolerance [1, 2].

We follow the methodology from [1, 2] for the train/test split and pre-processing of the corpora, and train our mod-

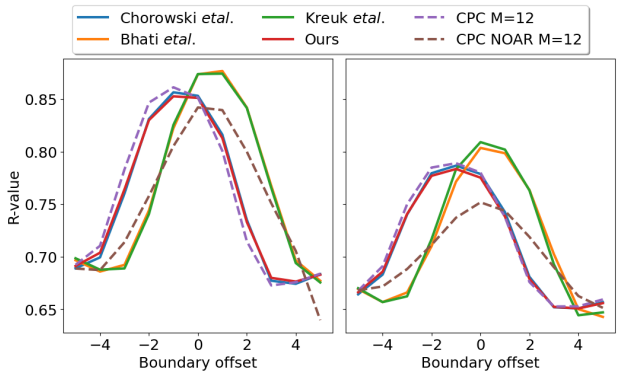

Fig. 2. Segmentation performance for different predicted boundary shifts on TIMIT (left) and Buckeye (right). Models with context builders perform better with an offset, indicating a representation shift.

els on the union of LibriSpeech *train-clean-100* and the *train* split of Buckeye. All models are trained for 50 epochs using a minibatch size of 32.

All models read single channel raw waveforms sampled at 16kHz, chunked into sequences of 20480 samples. Encoder g_{enc} applies five 1D convolutions with internal dimension 256 and filter widths (10; 8; 4; 4; 4). Convolutions are followed by channel-wise magnitude normalization and all but the last by ReLU activations. Convolutions are strided by (5; 4; 2; 2; 2), resulting in a 160-fold rate reduction, yielding a 256-dimensional latent vector extracted every 10ms. The segment is a network with a single fully connected hidden layer of 512 units and ReLU activation. Context-building models g_{ar} and s_{ar} are two-layer LSTM networks [11] with 256 units. We use $K=6$, $M=12$, $K^s=2$ and $M^s=4$ for (m)ACPC models. Each prediction head accesses all past contexts through a single transformer layer [12] with 8 scaled dot-product attention heads with internal dimension 2048 and dropout [13] with $p = 0.1$.

Our implementation is available at `https://github.com/chorowski-lab/CPC_audio`.

4. RESULTS AND DISCUSSION

4.1. Shifted predictions due to context building

Upon visual inspection of CPC and ACPC segmentations, we observed that predicted boundaries are often shifted in the same direction and by a similar amount. We hypothesize that the use of a context building network in the CPC model promotes pushing into the future the representation of the underlying signal. This would ease prediction of several steps ahead especially when the recurrent layer of the context builder is capable of exploiting the increased past context.

To test this possibility we re-evaluate the segmentation performance of the different methods by offsetting the predicted boundary positions by several fixed values. Note that each offset evaluates on different amount of data due to border effects, however scores are averaged across whole utterances limiting the consequences of this difference. Figure 2 shows how ACPC and mACPC exhibit an optimal offset value at 1 frame step (-10ms), while other methods peak at 0ms as expected from predictions without misalignment issues. To ensure this effect is due to the context builder and not other changes in ACPC (or mACPC), we also run a CPC model with and without it (dotted lines). It is worth noting that the optimal offset for both of these techniques is the same for the two datasets, however further investigation is needed to know whether this value is independent from the data. This consistent offset can be considered as a hyperparameter of the model. For comparison purposes, in the following experiments we report segmentation scores at a fixed offset of -10ms for models with a context builder.

Table 1. Phone segmentation on TIMIT (top) and Buckeye (bottom) test sets

Model	Precision	Recall	F1	R-val
Kreuk et al.	84.80	**85.77**	85.27	87.35
SCPC	**85.31**	85.36	**85.31**	**87.38**
ACPC	83.41	83.15	83.26	85.64
ACPC + Kreuk et. al. loss	83.68	84.74	84.69	86.86
mACPC	82.53	83.05	82.78	85.26
mACPC + Kreuk et. al. loss	84.63	84.79	84.70	86.86
Kreuk et al.	76.27	78.42	77.31	80.35
SCPC	**77.21**	**78.95**	**78.03**	**80.90**
ACPC	74.44	76.28	75.32	78.66
ACPC + Kreuk et. al. loss	74.68	76.59	75.59	78.88
mACPC	74.00	76.04	74.98	78.34
mACPC + Kreuk et. al. loss	74.70	76.81	75.72	78.97

Table 2. Phone classification on the test split of LibriSpeech train-clean-100

	On z vectors		On c vectors	
Model	Acc.	PER	Acc.	PER
Kreuk et al.	44.87	32.46	-	-
SCPC	43.79	31.62	-	-
ACPC	47.62	24.34	67.87	18.10
ACPC + Kreuk et. al. loss	47.82	25.93	67.99	18.15
mACPC	50.98	**21.15**	69.97	16.91
mACPC + Kreuk et. al. loss	**51.64**	21.69	**70.25**	**16.65**

4.2. Comparative study on segmentation and classification of phonemes

Tables 1 and 2 show the performance of the studied methods on phoneme segmentation and classification, respectively. Segmentation performance increases by adding to the model in [1] a second CPC at the segment level (as in [2]). Interestingly ACPC [5] and mACPC do not attain the same segmentation performance level despite their similarities and the offset correction. On the other hand they do achieve much better phoneme prediction rates, both frame synced (frame-wise

Table 3. Classification and phone segmentation performance of CPC variants. Frame-wise accuracy is calculated on encoded frames.

		R-value	
Model	Frame Acc.	Buckeye	TIMIT
CPC, $M=1$	17.28	66.98	70.77
CPC, $M=12$, no g_{ar}	38.11	75.17	84.20
CPC, $M=12$	47.90	78.89	86.11

accuracy) and through alignment (CTC PER). This points to an apparent compromise between ensuring proper boundaries and retaining phonemic information.

In order to assess which factors influence this trade-off, we perform a study in which each of the identified changes to CPC are tested* (Table 3). The one step ahead prediction used by [1] and SCPC is not in itself enough to improve segmentation performance. The key aspect, as hinted in [1], is the removal of the frame level context builder and the use of the encoded frames themselves as predictors.

Furthermore, the models in which we augment the ACPC loss with an auxiliary contrastive loss between adjacent representations show consistent improvements on segmentation and classification metrics, motivating its use as a method to tackle the performance trade-off. Moreover, the representations obtained by these models did not present a time shift even when using a context builder. This indicates that optimizing for detection of spectral changes penalizes representation shifting, and at least partially explains the better segmenting ability of [1] and SCPC.

4.3. The ZeroSpeech ABX task

Phoneme segmentation and classification are often targeted due to their potential use in applications such as the ABX task from the ZeroSpeech challenge [14], in which the objective is to match a speech example to its equivalent by choosing from two others that differ in a single phone. In this situation conserving the phonemic content is crucial since it is the discriminant factor. Results in Table 4 confirm that methods such as mACPC that excel in frame-wise accuracy and CTC PER also do so in ABX.

4.4. Integration of higher-level structure

We observe that the addition of higher level information using the second head in mACPC improves the results. SCPC showed an improvement in the case of phoneme segmentation, here we show the benefit of such strategy in phoneme discrimination with mACPC. Furthermore, it enables the detection of word boundaries using the technique from [2]. Word boundary experiment results in Table 5 indicate that

*We do not report variations for the number of negative samples because they did not have a significant effect on any of the metrics

Table 4. ABX scores on ZeroSpeech 2021 dev set. For models using context-networks the values are calculated on context vectors.

Model	ABX within	ABX across
Kreuk et al.	10.93	19.11
SCPC	20.18	16.26
ACPC	5.78	7.93
ACPC + Kreuk et. al. loss	5.67	7.78
mACPC	5.28	7.13
mACPC + Kreuk et. al. loss	**5.13**	**6.84**

Table 5. Word segmentation on Buckeye test set

Model	Precision	Recall	F1	R-val
SCPC	36.23	**32.75**	34.33	45.39
mACPC	**42.06**	30.32	**35.05**	**47.40**
mACPC + Kreuk et. al. loss	40.36	30.86	34.83	47.11

mACPC outperforms SCPC. We hypothesize that the superior phoneme representations of mACPC improve the quality of the pseudo-language model.

We also analyse topline performance in phoneme categorization and word-segmentation of SCPC and mACPC by using ground-truth phoneme boundaries during training. For mACPC, oracle segmentations improve PER on encoded frames from 21.15 (Table 2) to 20.12 and word segmentation R-value from 47.40 (Table 5) to 49.20. This contrasts with SCPC where use of ground-truth segments marginally affects results (31.62 to 31.93 PER and 45.39 to 46.1 R-value), suggesting that the representations obtained by the contrastive loss on adjacent frames [1] are insufficient for language modelling. The improved performance of mACPC when using ground-truth segmentation should motivate further work on improving estimated segmentation.

5. CONCLUSIONS AND FUTURE WORK

We investigate the applicability of CPC-based models to unsupervised phoneme segmentation and classification. As in [2] we propose a two-level model acting simultaneously on time synchronous frames and on variable-length segments, to capture linguistic regularities. We discover a performance compromise between phoneme segmentation and classification, and find that it stems from a roughly constant prediction shift induced by CPC's context modeling. We show that this issue can be alleviated by manually removing the offset from the representations or by using an auxiliary contrastive loss between consecutive latent representations. After accounting for the prediction shift, our model achieves competitive segmentation performance and outperforms existing approaches in phoneme classification, transcription, word boundary detection and ABX tests. Furthermore, the use of oracle phonemic alignments indicate that improvements on segment estimation may lead to even better performance in these tasks.

References

[1] F. Kreuk, J. Keshet, and Y. Adi, "Self-Supervised Contrastive Learning for Unsupervised Phoneme Segmentation," in *Proc. Interspeech 2020*, 2020, pp. 3700–3704.

[2] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak, "Segmental Contrastive Predictive Coding for Unsupervised Word Segmentation," in *Proc. Interspeech 2021*, 2021, pp. 366–370.

[3] H. Kamper and B. van Niekerk, "Towards unsupervised phone and word segmentation using self-supervised vector-quantized neural networks," *CoRR*, vol. abs/2012.07551, 2020. [Online]. Available: https://arxiv.org/abs/2012.07551

[4] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *CoRR*, vol. abs/1807.03748, 2018. [Online]. Available: http://arxiv.org/abs/1807.03748

[5] J. Chorowski, G. Ciesielski, J. Dzikowski, A. Łańcucki, R. Marxer, M. Opala, P. Pusz, P. Rychlikowski, and M. Stypułkowski, "Aligned Contrastive Predictive Coding," in *Proc. Interspeech 2021*, 2021, pp. 976–980.

[6] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06. New York, NY, USA: Association for Computing Machinery, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

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

[8] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, and N. L. Dahlgren, "Darpa timit acoustic phonetic continuous speech corpus cdrom," 1993.

[9] M. A. Pitt, K. Johnson, E. Hume, S. Kiesling, and W. Raymond, "The buckeye corpus of conversational speech: labeling conventions and a test of transcriber reliability," *Speech Communication*, vol. 45, no. 1, pp. 89–95, 2005.

[10] O. J. Räsänen, U. K. Laine, and T. Altosaar, "An improved speech segmentation quality measure: the r-value," in *Proc. Interspeech 2009*, 2009, pp. 1851–1854.

[11] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017.

[13] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 56, pp. 1929–1958, 2014.

[14] 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," 2020.