Recycle Your Wav2Vec2 Codebook: a Speech Perceiver for Keyword Spotting

Guillermo Cámbara
TALN Research Group
Universitat Pompeu Fabra
Barcelona, Spain
guillermo.cambara@upf.edu

Jordi Luque
Telefónica Research
Telefónica I+D
Barcelona, Spain
jordi.luque@telefonica.com

Mireia Farrús
Centre de Llenguatge i Computació
UBICS, Universitat de Barcelona
Barcelona, Spain
mfarrus@ub.edu

Abstract

Speech information in a pretrained wav2vec2.0 model is usually leveraged through its encoder, which has at least 95M parameters, being not so suitable for small footprint Keyword Spotting. In this work, we show an efficient way of profiting from wav2vec2.0's linguistic knowledge, by recycling the phonetic information encoded in its latent codebook, which has been typically thrown away after pretraining. We do so by transferring the codebook as weights for the latent bottleneck of a Keyword Spotting Perceiver, thus initializing such model with phonetic embeddings already. The Perceiver design relies on cross-attention between these embeddings and input data to generate better representations. Our method delivers accuracy gains compared to random initialization, at no latency costs. Plus, we show that the phonetic embeddings can easily be downsampled with k-means clustering, speeding up inference in 3.5 times at only slight accuracy penalties.

1 Introduction

Keyword Spotting (KWS) has benefited recently from the adoption of the Transformer architecture (Vaswani et al., 2017), as well as from recent advances in self-supervised learning proposals like wav2vec2.0 (Baevski et al., 2020).

Transformers are capable of capturing information from broader contexts, going beyond the locality of convolutional neural networks (CNN) (LeCun et al., 1989) and avoiding the vanishing/exploding gradients from recurrent neural networks (RNN) (Rumelhart et al., 1985). However, this comes at the quadratic cost of the self-attention mechanism (Bahdanau et al., 2014), which is even more pronounced in high-dimensional modalities like speech or vision. KWS models like the Keyword Spotting Transformer (KWT) (Berg et al., 2021) and the Audio Spectrogram Transformer (AST) (Gong et al., 2021), minimize such cost by downsampling the spectrogram space into patches, inspired by the Vision Transformer (ViT) (Dosovitskiy et al., 2020) proposal from computer vision.

In parallel, approaches like Wav2KWS (Seo et al., 2021) or the model from SUPERB (Yang et al., 2021) have successfully applied wav2vec2.0 to KWS. During training, wav2vec2.0 learns a latent codebook that codifies phonetic information, using each code as a target for training a feature encoder. This codebook is typically thrown away after training, only keeping the encoder for downstream tasks like automatic speech recognition (ASR) or KWS. Even though the encoder is capable of extracting rich features from raw waveforms, the size of it (at least 95M parameters for the BASE model) and its added latency time might discourage straightforward use for small footprint KWS classifiers.

In this short paper, we focus on exploring methods for recycling the phonetic information from the wav2vec2.0 latent codebook, showing that such information kickstarts the accuracy of a KWS model at initialization and leads to a better convergence, at virtually no cost in terms of inference time or model size.

The cornerstone of our proposal is a natural synergy that we have spotted between wav2vec2.0 and the recently proposed Perceiver (Jaegle et al., 2021) model. The latter's design relies on cross-attention between input data and a smaller latent bottleneck array, achieving smaller computational costs than pure self-attention over input data. We find that a pretrained wav2vec2.0 latent codebook can be used as an initialization for the Perceiver's latent bottleneck array, which boosts the model's accuracy with respect to random weight initialization. Furthermore, since many vectors from the wav2vec2.0 codebook contain similar phonetic information, we apply k-means clustering and average vectors belonging to the same clusters, yielding downsampled latent bottlenecks that provide faster inference at only slight accuracy costs.

Proceedings of the 29th International Conference on Computational Linguistics, pages 7166–7170
October 12–17, 2022.

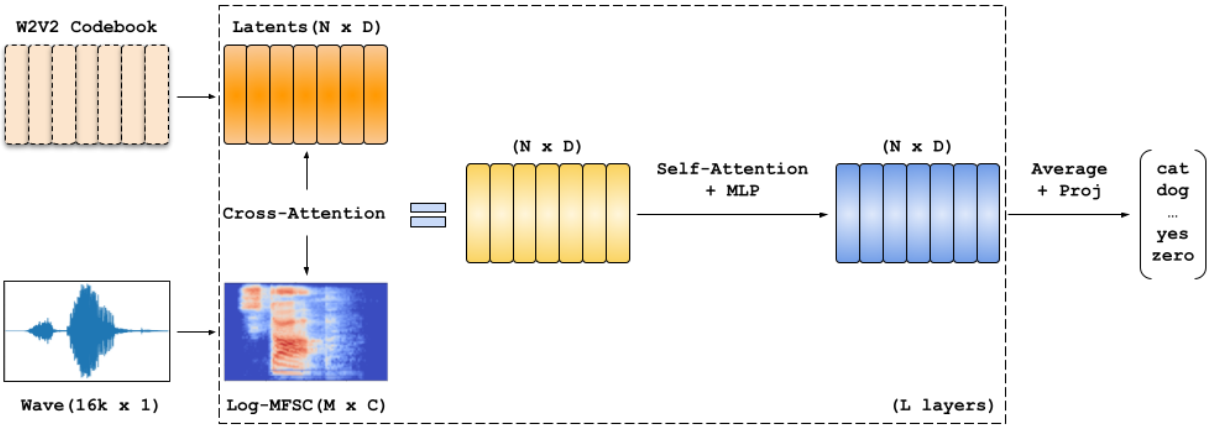

Figure 1: The Keyword Spotting Perceiver (KWP) model.

The focused contribution of this paper provides insight on efficient wav2vec2.0 transfer learning by latent codebook recycling, as well as showing the first application of the Perceiver model to a specific speech task like KWS, up to our knowledge.

2 Keyword Spotting Perceiver

Our Keyword Spotting Perceiver (KWP) is designed to take 1-second waveforms as inputs, converting these into log-mel spectrograms of $M = 100$ time steps and $F = 64$ frequency bins, which are linearly projected to a dimension of $C = 192$, resulting in a MxC data array. Fourier positional encodings are concatenated to the data array along the C dimension, using 64 frequency bands and a maximum resolution of 224, the best performing choice in the Perceiver paper. Cross-attention between such data array and a latent bottleneck array of NxD dimensions is done with a single head, whose output is further refined through a Transformer block containing self-attention with 8 heads and a multilayer perceptron (MLP) of hidden size 1024. The dimension for both self and cross attention heads is set to 64. Since the output is another latent array of NxD dimensions, we repeat cross-attention with the data array and the Transformer blocks for $L = 6$ layers, sharing the weights of cross-attends and Transformer across layers, in the style of a RNN. In earlier explorations we tried not sharing the weights but this led to performance degradation caused by overfitting. Finally, we average the latents in the D dimension, apply layer normalization and do a linear projection to get the class logits for prediction. A model depiction can be seen in Figure 1.

The latent array can be initialized randomly (KWP-BASE), or by transferring the weights from the latent codebook of a pretrained wav2vec2.0 model (KWP-W2V2). In this work, we recycle the latent weights of the wav2vec2.0 BASE model from the HuggingFace repo [1]. Such codebook consists of $N = 640$ vectors of dimension $D = 128$.

Since the complexity of cross-attention between latent and data arrays is $O(MN)$, we lose the efficiency gains from it with respect to self-attention over data array $O(N^2)$, since $O(MN) = O(100x640) = (6.4x10^4) > O(N^2) = O(100^2) = (10^4)$. To address this, we study three ways to downsample this latent space to lower dimensions $N = [320, 160, 80, 40, 20]$, by (1) sampling vectors randomly, (2) average pooling contiguous vectors and (3) clustering with k-means method and averaging vectors from the same cluster. According to the wav2vec2.0 paper, most of the codebook latents model specific English phonemes, being some phonemes represented by many latents. For instance, the silence phoneme is represented by 22% of the codebook. Being so, we expect k-means clustering to be the best downsampling method from the proposed ones, by clustering latents representing same or similar phonemes. Simple average pooling without k-means clustering might conserve phonetic information, although we expect it to be suboptimal given that we cannot guarantee that contiguous vectors in the codebook correspond to similar phonetics, thus potentially mixing information from different phonemes. Oppositely, random sampling guarantees that the individual information of each vector in the codebook is preserved, but as N gets lower many information is potentially lost, since most vectors are being dropped out.

[1] https://huggingface.co/facebook/wav2vec2-base

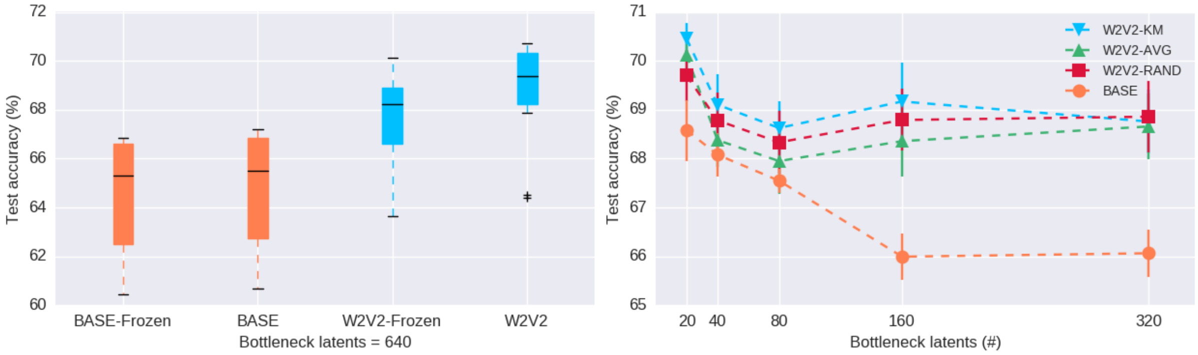

Figure 2: Test accuracy after a single training epoch, for a randomly initialized model (BASE), and another initialized with a wav2vec2.0 latent codebook weights (W2V2), with learnable or frozen weights (left). We also report the results of downsampling the bottleneck latents with k-means clustering (KM), averaging pooling (AVG) and random sampling (RAND) (right).

3 Experiments

We describe here the evaluations made for our Keyword Spotting Perceiver proposal. First, we assess the effects of transferring the wav2vec2.0 latent codebook to the Perceiver bottleneck at initialization. We also try different ways of downsampling this latent space, reporting accuracy comparisons between the baseline KWP-BASE model and the wav2vec2.0-initialized KWP-W2V2 model. Afterwards, we keep the best performing downsampling algorithm for the following round of experiments, where we let the system train until convergence. We report accuracy, model size and inference time metrics for KWP-BASE and KWP-W2V2 models with different latent number variants $N = [320, 160, 80, 40, 20]$.

Training, validation and testing phases are done with the standard partitions from the Google Speech Commands V2 dataset (Warden, 2018), obtaining the accuracy metrics from the 35-commands task. All timing metrics are obtained by doing inference over 1-second waveforms on CPU, warming up for 10 forward passes and averaging the time for 150 forward passes. We use the AdamW optimizer (Loshchilov and Hutter, 2018) with a step learning rate scheduler, decreasing the learning each epoch by a gamma factor of 0.98, starting with an initial learning rate of $1e^{-4}$. Batch size is set to 32, training for a single epoch in the initialization experiments and for 400 epochs in the convergence experiments. For the latter ones, we pick the top-10 checkpoints with the highest validation accuracy, averaging their weights to obtain the final checkpoint, which we use for test accuracy measurements. We open-source the PyTorch code [2] used for our experiments to the community.

Regarding data augmentation, we apply time shifting of ± 0.1 seconds with a probability of 60%. We also do resampling of the waveform signal between a $[0.85, 1.15]$ fraction of the input sampling rate, which is set to 16 kHz, with a probability of 100%. Background noise is also added in a range of $[5.0, 30.0]$ dBs and SpecAugment (Park et al., 2019) is done with 2 time masks of 25 frame size and 2 frequency masks of 7 frames each. The latter both data augmentation methods are also applied with a 100% probability during training. Even though, we relax the augmentation conditions for the shorter initialization experiments of a single epoch, to let the system learn a bit more in the early stages. Time shifting and resampling probabilities are lowered to a 30%, SpecAugment to a 70% and background noise addition to an 80%.

3.1 Initialization with Wav2vec2.0 Codebook

We check the impact of transferring the wav2vec2.0 codebook to KWP at initialization, by measuring the test accuracy after a single epoch, repeating training and test with 10 different seeds. Thus, we compare between KWP-BASE and KWP-W2V2 with all the $N = 640$ latent vectors, by making the latent bottleneck weights learnable (BASE and W2V2) and also freezing them (BASE-Frozen and W2V2-Frozen).

Figure 2(a) shows that both W2V2 and W2V2-Frozen have a significant performance advantage against BASE and BASE-Frozen. This suggests that the phonetic information transferred from the

[2] https://github.com/gcambara/speech-commands

wav2vec2.0 latent codebook kickstarts training, initializing the model already with useful information that the cross-attention mechanism can leverage. Furthermore, it is interesting to see how there is barely no performance differences between BASE and BASE-Frozen. We hypothesize that during the first epoch the randomly initialized BASE model has not learned enough phonetic information in its latent bottleneck, giving fewer chances for cross-attention to exploit relations with input data. The W2V2 model, oppositely, is able to leverage cross-attention early on, generating feedback between the phonetic information in the latent codebook and the cross-attention weights that are linked to input data.

To continue with, we repeat the same 10-seed experiment for $N = [320, 160, 80, 40, 20]$ latent vectors in the bottleneck. Different downsampling methods are tried: k-means clustering (W2V2-KM), average pooling (W2V2-AVG) and random sampling (W2V2-RAND). The results, as seen in Figure 2(b), highlight that the three latent downsampling methods are effective for boosting the performance with respect to the BASE model. W2V2-KM is the best performing model, confirming that averaging latents belonging to the same phonetic clusters is preferable, rather than simply averaging contiguous latents like in W2V2-AVG, or randomly sampling latent vectors like W2V2-RAND does, which loses representation power as N decreases.

3.2 Assessment at Convergence

To evaluate the accuracy of KWP-BASE and KWP-W2V2 models after convergence, we let the models train again for 400 epochs. This time, we only experiment with learnable latent weights and k-means clustering downsampling, given that the latter has reported the best initialization results. Training and test is done now with 3 seeds, varying the number of latents again with the same selection, $N = [640, 320, 160, 80, 40, 20]$, and comparing between BASE and W2V2 variants.

As Figure 3 depicts, W2V2 maintains significant advantage for all the numbers of latents, with a peak mean accuracy of $96.26 \pm 0.04\%$ at 640 latents, higher than BASE's top accuracy of $95.6 \pm 0.2\%$ at 80 latents. The W2V2 variant seems to scale well with the number of latents, as opposite to the BASE model, which might struggle to clusterize phonetic information in the latent space as it grows bigger. Still, KWP (1.5M parameters) is slightly behind to its self-attention counterparts, as the lightest KWT (0.6M parameters) scored a 96.8% accuracy, and AST scored 98.1%. Nevertheless, note that AST is pretrained with ImageNet (Deng et al., 2009), and is much larger (87M parameters). Even though, we motivate further research on fine-tuning KWP towards state-of-the-art performance.

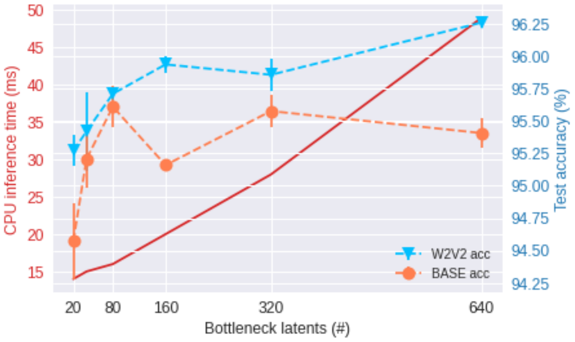

Figure 3: KWP-BASE (orange) and KWP-W2V2 (blue) test accuracies after convergence, for different numbers of bottleneck latents, with CPU inference time (red).

The inference time of the 640 latent model is 49 ± 5 ms, and 14 ± 2 ms for the smaller 20 latent model. Given that the accuracy is $95.3 \pm 0.1\%$ for the latter, only a 1% relative accuracy is lost with k-mean clustering downsampling, while increasing inference speed in 3.5 times. The accuracy of the BASE model at 20 latents is $94.6 \pm 0.3\%$, which is significantly below W2V2's. This confirms that even a hard downsampling of 640 to 20 latents of wav2vec2.0 information is still preferable to randomly initializing the latent space in KWP.

4 Conclusion

In this work, we have shown that phonetic information from the wav2vec2.0 latent codebook can be recycled, by transferring it to the latent bottleneck weights of a Keyword Spotting Perceiver. Accuracy gains are consistently significant with respect to random initialization of the latent bottleneck, both at early and late stages of training for the KWS task. Furthermore, we have studied easy-to-apply downsampling techniques for compressing the latent codebook, like averaging k-means clusters, having sped up the inference time of the model up to 3.5 times, at only a 1% accuracy drop.

We believe that our work motivates further research on efficient ways of profiting from the information in big self-supervised models like wav2vec2.0, as well as on applications for other tasks like ASR, for instance.

Acknowledgements

This work is a part of the INGENIOUS project, funded by the European Union's Horizon 2020 Research and Innovation Programme and the Korean Government under Grant Agreement No 833435.

References

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. *Advances in Neural Information Processing Systems*, 33:12449–12460.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Axel Berg, Mark O'Connor, and Miguel Tairum Cruz. 2021. Keyword transformer: A self-attention model for keyword spotting. *arXiv preprint arXiv:2104.00769*.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Yuan Gong, Yu-An Chung, and James Glass. 2021. Ast: Audio spectrogram transformer. *arXiv preprint arXiv:2104.01778*.

Andrew Jaegle, Felix Gimeno, Andy Brock, Oriol Vinyals, Andrew Zisserman, and Joao Carreira. 2021. Perceiver: General perception with iterative attention. In *International Conference on Machine Learning*, pages 4651–4664. PMLR.

Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. 1989. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551.

Ilya Loshchilov and Frank Hutter. 2018. Decoupled weight decay regularization. In *International Conference on Learning Representations*.

Daniel S Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D Cubuk, and Quoc V Le. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *Proc. Interspeech 2019*, pages 2613–2617.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1985. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science.

Deokjin Seo, Heung-Seon Oh, and Yuchul Jung. 2021. Wav2kws: Transfer learning from speech representations for keyword spotting. *IEEE Access*, 9:80682–80691.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Pete Warden. 2018. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*.

Shu-wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y Lin, Andy T Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, et al. 2021. Superb: Speech processing universal performance benchmark. *arXiv preprint arXiv:2105.01051*.