Analyzing acoustic word embeddings from pre-trained self-supervised speech models

ANALYZING ACOUSTIC WORD EMBEDDINGS FROM PRE-TRAINED SELF-SUPERVISED SPEECH MODELS

Ramon Sanabria, Hao Tang, Sharon Goldwater

The University of Edinburgh

ABSTRACT

Given the strong results of self-supervised models on various tasks, there have been surprisingly few studies exploring self-supervised representations for acoustic word embeddings (AWE), fixed-dimensional vectors representing variable-length spoken word segments. In this work, we study several pre-trained models and pooling methods for constructing AWEs with self-supervised representations. Owing to the contextualized nature of self-supervised representations, we hypothesize that simple pooling methods, such as averaging, might already be useful for constructing AWEs. When evaluating on a standard word discrimination task, we find that HuBERT representations with mean-pooling rival the state of the art on English AWEs. More surprisingly, despite being trained only on English, HuBERT representations evaluated on Xitsonga, Mandarin, and French consistently outperform the multilingual model XLSR-53 (as well as Wav2Vec 2.0 trained on English).

***Index Terms*—** acoustic word embedings, self-supervised learning, HuBERT, Wav2Vec2.0, XLSR-53, cross-lingual

1. INTRODUCTION

Speech tasks such as query-by-example, voice search, keyword spotting, and word discovery typically require measuring distances between speech segments [1, 2, 3]. To avoid the computational expense of the traditional Dynamic Time Warping method, recent papers often use *Acoustic word embeddings* (AWEs), which represent variable-length segments as fixed-dimensional vectors [4, 5]. These can then be compared quickly using measures such as cosine similarity.

An effective AWE algorithm will embed different instances of the same word close together in the vector space, and instances of distinct words further away. One of the main challenges is how to encode the sequential information from the speech signal into a vector space that has no inherent sequential structure. The representation should encode not just which phones are present, but their ordering—so that words like *task*, *stack*, *cast*, and *cats*, which all contain the same set of phones, will have distinct clusters in the representational space.[1] Various approaches have been developed that use supervision from known word pairs [6, 7, 8, 9], but here we focus on *unsupervised* learning of AWEs, where only raw audio is available [10, 11, 12]. This scenario is potentially important for speech applications in low-resource languages.

A common baseline that preserves sequential order while extracting a fixed-dimensional representation is *subsampling* (see, e.g., [12, 13, 14]): selecting a fixed number of (usually equally spaced) frames and concatenating them. Subsampling is simple and fast to compute, but, depending on the input frame size and number of samples, it can lead to prohibitively large embeddings and/or loss of phonetic information. Moreover, it does not perform as well on word discrimination tasks as newer learning-based approaches. Unsupervised learning-based methods typically work in two steps: first, apply an unsupervised term detection (UTD) system [15] to identify similar pairs of segments that are likely to be the same word or phrase, then use the pairs as a noisy set of positive examples to train a neural network. Network architectures vary, but the basic idea is to train the system's representations to make the positive examples closer together in the space [11, 12] (and in some models, also to separate additional negative example pairs [16, 17]).

Though effective, this learning-based approach relies on running UTD on the target language, which itself is computationally intensive and sensitive to differences in input features [12]. Here, we explore whether using newer self-supervised speech representations, available as pre-trained models [18, 19], may obviate both the UTD step and the need for specialized models to learn unsupervised AWEs. We hypothesize that the contextualized speech representations learned by these models will implicitly encode the sequential information needed for AWEs (e.g., by capturing within each frame the local acoustic effects of coarticulation, and/or information at a longer timescale that is needed to reconstruct the masked input during pretraining—where average mask span is nearly 300ms [19]). If so, then it should be possible to create effective AWEs with much smaller dimension than subsampling just by using simple pooling methods such as mean- or max-pooling. While pre-trained models and these pooling methods are widely used across many applications, as far as we know this paper is the first to compare and analyze them for creating AWEs.

We evaluate AWEs created using different pooling operations on the representations from two English pre-trained models—HuBERT (HB) and wav2vec 2.0 (W2V2)—and one multilingual pre-trained model (`XLSR-53`, which also uses the W2V2 architecture). Using a standard word discrimination task, we test on English, Xitsonga, Mandarin, and French—where the latter three better represent a low-resource target language scenario, where a large pre-trained model on that language is unlikely to be available.

In accordance with our hypothesis, we find that on our English test set, AWEs created by mean-pooling the HB representations perform almost as well as the state-of-the-art learned pooling model (MCVAE [20]) with equivalent dimensionality; and outperform subsampling, despite having a much lower dimensionality. Mean-pooled W2V2 representations underperform HB, but are still better than subsampled ones, and considerably better than the MFCC baseline.

Our experiments on other languages show that (1) the multilingual W2V2 representations work better than the monolingual English ones, but still underperform HB (for which only an English model is available); (2) unlike on English, mean-pooling does not outperform subsampling with the HB representations, although it comes close;

[1] Of course, coarticulation effects, which depend on the local ordering of phones, will mean that the phones in these words are not necessarily pronounced in the same way. We assume that good AWE models can effectively exploit this information to help capture sequential structure, though they may also model longer distance sequential information.

and (3) when equated on dimensionality, the HB representations are not quite as good as those learned by the best recent models, but perform surprisingly well given that (unlike these models) they require no training on the target language at all.

Overall, our results indicate that the right self-supervised model capture some sequential information needed for AWEs, making simpler pooling methods effective. While these contextualized representations don't generalize fully to other languages, they still work well with no training required.

2. OVERVIEW OF THE APPROACH

Our approach starts by encoding the corpus using a pre-trained model. The embedding for a given word is created by extracting the model's representations from the start to the end of that word and pooling these to create a fixed dimensional representation.

More formally, let us define two word segments $x^1_{s_1:t_1}$ and $x^2_{s_2:t_2}$ from utterances x^1 and x^2, where s_i and t_i are the start and end times of the word from x^i. We encode both utterances using a contextual self-supervised encoder f, yielding $z^1 = f(x^1)$ and $z^2 = f(x^2)$. We then pool the encoded representations of each word using a pooling function g, to obtain embeddings $c_1 = g(z^1_{s_1:t_1})$ and $c_2 = g(z^2_{s_2:t_2})$. We experiment with four different pooling functions: subsampling, argmax, sum, and mean.

Our encoder models are HB and W2V2, two recent self-supervised models based on Transformer architectures with latent states. In both models, CNN layers are used to encode the speech signal into audio features, followed by Transformer layers trained using a BERT-like masked language modeling objective (masking some of the input frames). Rather than using the context to predict exactly these frames, both models aim to predict quantized latent units. In W2V2 these are learned jointly in the neural model, while in HB training iterates between a separate clustering step (using K-means) and training the neural network to predict these clusters. Our pooling functions are applied to the frame-level representations from the Transformer layers.

To evaluate the AWEs, we use the same-different word discrimination task (henceforth, *same-diff*) [21]. For each pair of embeddings (c_1, c_2), we measure their cosine similarity and compare this value to a threshold to decide whether both embeddings belong to the same word type. We repeat this process across all possible pairs of a given set of word instances. By varying the similarity threshold across all possible values, we obtain an ROC curve; the final evaluation measure is the Average Precision (AP), or area under this curve.

3. EXPERIMENTS

We experiment with frame-level representations from three models: English W2V2 and HB, and multilingual W2V2. For the English models, we use `Wav2Vec 2.0 Large` and `HuBERT Large` from the official repository[2], which are both pre-trained on the 60k hour split from the Libri-light dataset [22] and have 317M and 316M parameters respectively. For the multilingual model, we use `XLSR-53` from the official repository, which is trained on 53 languages (including Mandarin and French, but not Xitsonga) and has 317M parameters. All models have a contextual representation with 1024 dimensions, so the c's also have 1024 dimensions when g is argmax, sum or mean. For subsampling, we concatenate 10 equally spaced frames, resulting in a 10240-dimensional embedding. Except where noted, frame-level representations, the z's, are normalized by subtracting the mean and dividing by the variance of the evaluated set.

All models have 23 Transformer layers. To save time and computation, we limit our study to layers 1, 11, 15, 19 and 23 and choose the best layer using the development set when available.

We test our AWEs on English, Mandarin, French, and Xitsonga (a low-resource Bantu language spoken in southern Africa). For the English experiments, we focus mainly on a cross-domain setting, testing on the Buckeye corpus of conversational speech [23] (whereas the pre-training corpus, Librispeech, is read speech). We use the dev and test splits (6h each) defined by [24][3]. We also include some in-domain results from Librispeech [25] `dev-clean` and `test-clean` (5.4h each). The Xitsonga data comes from the NCHLT corpus [26], which contains 2.5 hours of read speech and no dev/test split. For Mandarin and French we use the test sets from the ZeroSpeech Challenge 2017 [27][4], containing 2.5 and 24 hours of read speech respectively. With the larger French set, we created separate dev and test sets as described below.

To perform the *same-diff* evaluation, start and end timestamps for each word are needed. The Buckeye and Xitsonga corpora include manually corrected timestamps; for the French and Mandarin corpora we use the Kaldi forced alignments provided; and for Librispeech we obtained alignments using the Montreal Forced Aligner[28]. Following [24], we evaluate using the words from each split that are at least 5 characters and 0.5 seconds long.[5] Since the French data is larger, we created separate dev and test[6] sets from it by randomly sampling 4000 of the relevant word tokens for dev and another 4000 (without replacement) for test.[7]

Where possible, we compare to results from [12] and [20], two recent papers on unsupervised acoustic word embeddings who evaluated on English and Xitsonga. [20] appear to have the best published results on these languages using an architecture they call Maximal Sampling Correspondence VAE, while [12] use one of the most well-studied pooling architectures, the CAE-RNN [11], and provide several useful baselines. Both architectures learn from UTD pairs extracted from the target language data (as described in the Introduction), and [12] explore different frame-level features that are used to represent the UTD pairs as input to the CAE-RNN. Their best results are obtained using input features learned using Contrastive Predictive Coding (CPC) [29], a self-supervised model. Thus, those results combine self-supervision with a learned pooling function, whereas we focus on self-supervision alone (but using pre-trained W2V2 and HB instead of training CPC on the target language).

3.1. Monolingual setting

We first present results on English, focusing on the HB representations, which we found to work better than W2V2. (Selected W2V2 results are presented for comparison in Section 3.2). Preliminary experiments showed that representations from layer 19 worked best for English, so all results in this section use that layer from that model.

Figure 1 (top) compares the results of different pooling methods as well as the effects of frame-level normalization. Consistent with

[2]https://github.com/pytorch/fairseq

[3]https://github.com/kamperh/bucktsong_segmentalist/blob/master/features/

[4]https://download.zerospeech.com/

[5]We obtained the relevant words for Buckeye and Xitsonga from https://github.com/kamperh/recipe_bucktsong_awe_py3; for the other corpora we extracted the words using the same criteria.

[6]https://github.com/ramonsanabria/awe_ssl

[7]The number of word tokens/word types extracted from each set is as follows. Buckeye: 4054/2732 (dev), 4054/2121 (test); Librispeech: 7422/4535 (dev), 8162/4793 (test); Xitsonga: 6384/1795; Mandarin: 4132/3565; French: 4000/3043 (dev), 4000/3030 (test).

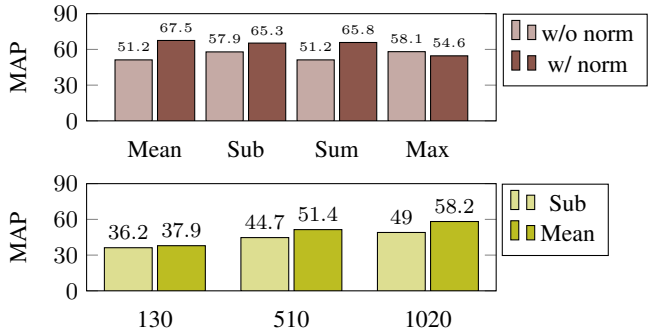

Fig. 1. Average Precision results on the Buckeye development set with pooled HB representations. Top: Results of frame-level normalization and different pooling functions. Subsampled embeddings have 10240 dimensions; the others have 1024. Bottom: Comparing reduced dimensionality embeddings, using mean pooling or subsampling followed by PCA. For comparison, subsampling MFCC representations yields a 130-dimensional embedding with AP of 19.4%.

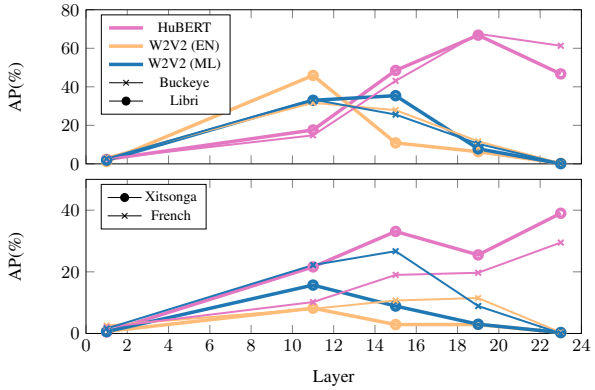

Fig. 2. Results on English datasets—Buckeye and Librispeech (up), and on non-English datasets—-Xitsonga and French (down). Both results use HB trained on English and W2V2 trained on English (EN) and Multiple languages (ML) and mean as pooling mechanism. All results are computed on the development set. We use mean pooling, but relative perfromance across layers is consistent with subsampling.

work on related tasks [30, 31], we find that normalization greatly improves performance, so we use it in all remaining experiments (including with W2V2, where it also helps). More importantly, we find evidence to support our hypothesis that contextualized representations from self-supervised models implicitly encode sequential information. Specifically, we see that the mean- and sum-pooling strategies work as well as sub-sampling, despite the latter having 10 times more dimensions and explicitly modeling sequential order. Since mean and sum work equally well, we focus on mean-pooling (as compared to subsampling) in the remainder of the paper.

So far, our AWEs have at least the same dimensionality as the pre-trained model representations, *i.e.* 1024 dimensions. However, some downstream tasks require fewer dimensions due to computational constraints [13], and many previous AWE systems focus on fewer dimensions. Therefore, for comparison with previous systems, we use PCA to reduce the dimensionality of the frame-level representations to 130 dimensions. For the subsample-pooled AWE, we perform a 13 dimensions PCA to the frame-level representations and then perform subsampling. For mean-pooling, we reduce the frame-level dimensionality to the target dimension (*e.g.*, 130). Figure 1 (bottom) shows that comparison, where we see that for embeddings of the same size down to 130 dimensions, mean pooling always outperforms subsampling, though the benefit is less for smaller embeddings. The figure shows development results, but we also computed test set results with 130 dimensions in order to compare to the best published results for an unsupervised AWE model (MCVAE) [20] (also with 130 dimensions). The AP score of the HB mean-pooled embeddings (35.2%) is close to the MCVAE (39.5%) despite having a much simpler pooling strategy—although we note that the MCVAE is trained using a much smaller amount of English data than the pre-training data for HB.

3.2. Cross-lingual setting

Although the results on English are promising, unsupervised AWEs are more likely to be useful for low-resource languages, where a large pre-trained model may not be available for the target language. In this section, we investigate how well AWEs obtained from English or multilingual pre-trained models can work on other languages. We test on two languages that are included in the multilingual pre-training data to differing degrees (Mandarin and French, with 12h and 1686h, respectively, included in the 56k total hours of pre-training data), and one truly low-resource language (Xitsonga), which is not included in pre-training (nor are any other closely related languages).

We start by determining the best layers to use and whether that differs across languages or models. Figure 2 (top) shows the development set performance on *same-diff* task across different layers on the English datasets: Librispeech and Buckeye. First, we observe that the result is consistent with previous work on W2V2 [32], which showed that phone and word identities were encoded most strongly in the middle layers. For HB, the trend is different: the later layers generally have better AP scores, suggesting that phone and word identities are better encoded there. In terms of the best performance, HB outperforms both versions of W2V2. Since the English models are pre-trained with the same amount of data and have similar numbers of parameters. This suggests that the difference in performance is due to differences in the pre-training objective.

Next, we repeat the layer-wise analysis on French (dev set) and Xitsonga. Since Xitsonga and Mandarin have no dev sets, we do this analysis on the Xitsonga test set but leave out Mandarin to avoid over-using test sets. As on English, we see that for W2V2 the best performance is near the middle, while for HB the later layers are better—though unlike in English, the final layer appears to be best. We can also see that, for the best-performing layer of each model, multilingual W2V2 works better for these non-English languages than English W2V2, but the improvement is not enough to outperform the English HB model. We hypothesize that a multilingually trained HB model might do even better, but no such model is currently available.

We now compute test set performance on all data sets using the best layers from previous analyses (HB: layer 19 for English, 23 for other languages; W2V2: layer chosen using dev set where available, otherwise layer 11). Figure 3 shows the AP scores for all models, using mean-pooling (1024 dims) and subsampling (10240 dims). For the best HB model, we also include subsampling after PCA to match the dimensions of mean-pooling. Unlike in English, subsampling outperforms mean-pooling for all other languages. This is likely because the English contextual information learned by the model doesn't generalize fully to languages with different phonotactic patterns (see Section 3.3 for analysis); therefore the explicit sequential modeling provided by subsampling can still help. Interestingly, reducing dimensionality prior to subsampling (white bars in Figure 3) hardly reduces the performance on the non-English languages, in contrast to the large drop for English. Therefore mean-pooling and subsampling yield similar performance on these languages at the same dimensionality,

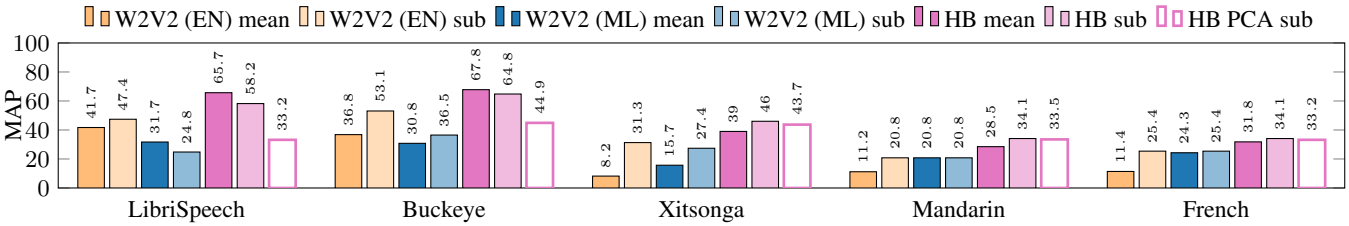

Fig. 3. A comparison of W2V2 and HB across different languages. Mean-pooling (1024 dims) is denoted with darker colors, and subsampling (10240 dims) with lighter colors. We also include a version of sampling (the white bars) that uses PCA to match the dimensions of averaging.

Table 1. Test set results on Xitsonga, compared to various baselines. We indicate the input representation (with training language: ENglish or XiTSonga), the pooling function, and the dimensionality. Note: CPC features were trained on only 2.5h (TS) or 6h (EN), whereas HB was trained on 60k hours (EN).

Input repr.	Pooling	Dims	AP (%)
MFCC	MCVAE (TS) [20]	130	44.4
CPC (TS)	CAE-RNN (TS) [12]	130	40.9
CPC (EN)	CAE-RNN (TS) [12]	130	41.8
CPC (TS)	Subsample [12]	356	18.7
MFCC	Subsample [12]	130	18.4
HB (EN)	PCA+Subsample	130	35.5
HB (EN)	PCA+mean	130	34.9
HB (EN)	PCA+Subsample	1020	43.7
HB (EN)	mean	1024	39.0
HB (EN)	Subsample	10240	46.0

with subsampling slightly ahead—though mean-pooling could still be preferable in practice since it does not require the extra PCA step.

Finally, we compare our results for Xitsonga to previous work [12, 20], using 130-dimensional embeddings (Table 1). While the cross-lingual HB embeddings are slightly worse than previous results using CAE-RNN and MCVAE pooling, they don't require any training on the target language. That said, the CPC features [12] and pooling architectures [12, 20] were trained on only a few hours of data, so in future it would be worth comparing CPC and HB representations trained on comparable amounts of data, and/or applying CAE-RNN or MCVAE to HB representations.

3.3. Qualitative analysis

We hypothesized that AWEs from self-supervised models would implicitly encode sequential information. Our quantitative results support this hypothesis and suggest that some of this information is language-specific. However, we wish to know whether the results are simply due to good encoding of coarticulation or whether longer-distance sequential information is also encoded. To explore this question, we looked at five words that (according to `CMUDict`) contain the same set of four phones in different order. Figure 4 (left) visualizes the mean-pooled HB AWEs for all instances of these words in Librispeech `test-clean`, `dev-clean`, and `train-clean-100`. We observe that the instances are clustered according to their respective word types, providing further evidence that the AWEs encode sequential ordering in some way.

More intriguingly, the most frequent word (*asked*) forms three sub-clusters, and when we examined them, we found evidence that they encode word-level context rather than just immediate phonetic context (as from coarticulation). To illustrate, Figure 4 (right) color-codes the instances of *asked* depending on the following word, and

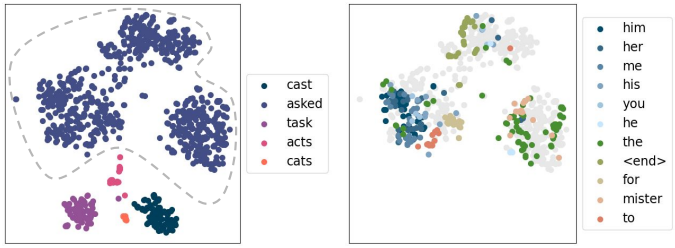

Fig. 4. t-SNE visualization of AWEs from normalized and mean-pooled layer 19 of HB. The left plot shows instances of five words that contain the same set of phones (according to `CMUDict`) but in different orders. The right plot shows only instances of *asked* (from inside the dotted line of the left plot), color-coded according to which of several frequent words follow that instance. Light gray dots are all instances that are followed by words not listed in the plot legend.

shows that instances followed by pronouns (shades of blue) cluster together. Critically, instances followed by *me* cluster with the other pronouns, rather than with the instances that are followed by *mister*, even though the latter shares its initial phone with *me* while the other pronouns do not. Overall, this analysis suggests that the HB AWEs encode much more than just coarticulation, which may explain why mean-pooling is less successful in the cross-lingual setting. It is also unclear whether sub-clustering frequent words by word-level context is desirable for AWEs; this may depend on the task.

4. CONCLUSION

We hypothesized that self-supervised frame-level representations contain sufficient context so we do not need to model sequential order to construct AWE. Our results on conversational English confirm that this can be true, but depends on the model: for HuBERT (but not W2V2) AWEs constructed using mean-pooling outperform subsample-pooling despite having 10 times fewer dimensions. HuBERT representations also perform better overall, suggesting that the nature of the input features has a strong influence on AWE quality, regardless of the pooling method.

When we applied HuBERT to other languages, we found that although the encoded context trained on English does not fully generalize to those languages, mean-pooling worked almost as well as subsampling. With unreduced dimensionality, these methods perform similarly or better than state-of-the-art unsupervised AWE models, though performance degrades when equating dimensionality. Unlike the comparison models, these results are obtained with no training on the target language, so a promising future direction would be to explore whether results on other languages can be improved by adapting the HB model to the target language by continuing self-supervised training on a small amount of unlabeled target language data.

5. REFERENCES

[1] Yaodong Zhang and James R Glass, "Unsupervised spoken keyword spotting via segmental dtw on gaussian posteriorgrams," in *ASRU*, 2009.

[2] Mohamed S Barakat, Christian H Ritz, and David A Stirling, "Keyword spotting based on the analysis of template matching distances," in *ICSPCS*, 2011.

[3] Mathias De Wachter, Mike Matton, Kris Demuynck, Patrick Wambacq, Ronald Cools, and Dirk Van Compernolle, "Template-based continuous speech recognition," *TASLP*, 2007.

[4] Andrew L Maas, Stephen D Miller, Tyler M O'neil, Andrew Y Ng, and Patrick Nguyen, "Word-level acoustic modeling with convolutional vector regression," in *ICML Workshop*, 2012.

[5] Keith Levin, Katharine Henry, Aren Jansen, and Karen Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *ASRU*, 2013.

[6] Herman Kamper, Weiran Wang, and Karen Livescu, "Deep convolutional acoustic word embeddings using word-pair side information," in *ICASSP*, 2016.

[7] Yushi Hu, Shane Settle, and Karen Livescu, "Multilingual jointly trained acoustic and written word embeddings," in *Interspeech*, 2020.

[8] Shane Settle and Karen Livescu, "Discriminative acoustic word embeddings: Tecurrent neural network-based approaches," in *SLT*, 2016.

[9] Shane Settle, Keith Levin, Herman Kamper, and Karen Livescu, "Query-by-example search with discriminative neural acoustic word embeddings," in *Interspeech*, 2017.

[10] Nils Holzenberger, Mingxing Du, Julien Karadayi, Rachid Riad, and Emmanuel Dupoux, "Learning word embeddings: Unsupervised methods for fixed-size representations of variable-length speech segments," in *Interspeech*, 2018.

[11] Herman Kamper, "Truly unsupervised acoustic word embeddings using weak top-down constraints in encoder-decoder models," in *ICASSP*, 2019.

[12] Lisa van Staden and Herman Kamper, "A comparison of self-supervised speech representations as input features for unsupervised acoustic word embeddings," in *SLT*, 2021.

[13] Herman Kamper, Aren Jansen, and Sharon Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, 2017.

[14] Herman Kamper, Karen Livescu, and Sharon Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *ASRU*, 2017.

[15] Aren Jansen and Benjamin Van Durme, "Efficient spoken term discovery using randomized algorithms," in *ASRU*, 2011.

[16] Christiaan Jacobs, Yevgen Matusevych, and Herman Kamper, "Acoustic word embeddings for zero-resource languages using self-supervised contrastive learning and multilingual adaptation," in *SLT*, 2021.

[17] Algayres Robin, Adel Nabli, Benoit Sagot, and Emmanuel Dupoux, "Speech sequence embeddings using nearest neighbors contrastive learning," in *ICASSP*, 2022.

[18] Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *TASLP*, 2021.

[19] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[20] Puyuan Peng, Herman Kamper, and Karen Livescu, "A correspondence variational autoencoder for unsupervised acoustic word embeddings," in *NeurIPS Workshop*, 2020.

[21] Michael A Carlin, Samuel Thomas, Aren Jansen, and Hynek Hermansky, "Rapid evaluation of speech representations for spoken term discovery," in *Interspeech*, 2011.

[22] Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, et al., "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP*, 2020.

[23] Mark A Pitt, Keith Johnson, Elizabeth Hume, Scott Kiesling, and William Raymond, "The buckeye corpus of conversational speech: Labeling conventions and a test of transcriber reliability," *Speech Communication*, 2005.

[24] Herman Kamper, "Unsupervised neural and bayesian models for zero-resource speech processing," in *Thesis*, 2017.

[25] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*, 2015.

[26] Nic J De Vries, Marelie H Davel, Jaco Badenhorst, Willem D Basson, Febe De Wet, Etienne Barnard, and Alta De Waal, "A smartphone-based asr data collection tool for under-resourced languages," *Speech communication*, 2014.

[27] Ewan Dunbar, Xuan Nga Cao, Juan Benjumea, Julien Karadayi, Mathieu Bernard, Laurent Besacier, Xavier Anguera, and Emmanuel Dupoux, "The zero resource speech challenge 2017," in *ASRU*, 2017.

[28] "Montreal Forced Aligner," `https://github.com/MontrealCorpusTools/Montreal-Forced-Aligner`.

[29] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[30] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, "Unsupervised speech recognition," in *NeurIPS*, 2021.

[31] Alexander H Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski, "Towards end-to-end unsupervised speech recognition," *arXiv preprint arXiv:2204.02492*, 2022.

[32] Ankita Pasad, Ju-Chieh Chou, and Karen Livescu, "Layer-wise analysis of a self-supervised speech representation model," in *ASRU*, 2021.