ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-6874-1/25/$31.00 ©2025 IEEE | DOI: 10.1109/ICASSP49660.2025.10890719

Unsupervised Word Discovery: Boundary Detection with Clustering vs. Dynamic Programming

Simon Malan, Benjamin van Niekerk, Herman Kamper
Electrical and Electronic Engineering, Stellenbosch University, South Africa
24227013@sun.ac.za, benjamin.l.van.niekerk@gmail.com, kamperh@sun.ac.za

***Abstract*—We look at the long-standing problem of segmenting unlabeled speech into word-like segments and clustering these into a lexicon. Several previous methods use a scoring model coupled with dynamic programming to find an optimal segmentation. Here we propose a much simpler strategy: we predict word boundaries using the dissimilarity between adjacent self-supervised features, then we cluster the predicted segments to construct a lexicon. For a fair comparison, we update the older ES-KMeans dynamic programming method with better features and boundary constraints. On the five-language ZeroSpeech benchmarks, our simple approach gives similar state-of-the-art results compared to the new ES-KMeans+ method, while being almost five times faster. Project webpage: https://s-malan.github.io/prom-seg-clus.**

***Index Terms*—word segmentation, lexicon learning, zero resource speech processing, unsupervised learning**

I. Introduction

Unsupervised word segmentation aims to identify word-like segments in raw speech audio. This is challenging since speech is a continuous stream without obvious silences between words [1]. Constructing a lexicon poses another challenge, as no two speakers are identical and even individual speakers show a lot of variation in their speech. Remarkably, human infants navigate these challenges, demonstrating word discrimination and recognition capabilities within their first year [2], [3]. Solving the problem of segmenting and clustering speech could provide a way to improve our understanding of human language acquisition [4]. It could also advance the development of low-resource speech technologies [5].

Early word discovery methods employed direct pattern matching, usually using dynamic time-warping [6], to find matching segments in pairs of utterances. Although these methods have shown progress [7], they fail to discover patterns that cover all the speech audio. In this paper we are particularly interested in full-coverage systems that provide a full tokenization of the input speech into word-like units.

Several full-coverage methodologies have been considered [8]–[10]. One strand of methods discovers phone-like units and does word segmentation and lexicon learning on top of (or in conjunction with) the subword units [11]–[13]. The recent duration-penalized dynamic programming (DPDP) method is one example [14], [15]. It uses quantized self-supervised speech representations for subword learning and an autoencoding recurrent model for subsequent word learning. Another strand of methods models higher-level units like words directly without explicit subword modeling [16]–[19]. The older embedded segmental K-means (ES-KMeans) method is an example [20]. It uses an iterative segmenting and clustering scheme, each time choosing the current best boundary hypothesis using dynamic programming (similar to DPDP). Surprisingly, ES-KMeans is still competitive [15], despite using older speech features and boundary constraints.

In this paper we propose a much simpler full-coverage speech segmentation system that does not require dynamic programming. Inspired by Pasad et al. [21], we argue that (1) well-defined word boundaries can be found through a lightweight method measuring dissimilarity between adjacent self-supervised features, and (2) an explicit lexicon can be constructed by just doing K-means on the discovered word segments (if appropriate segmental features are used). We compare this method to several other approaches – including our own updated version of ES-KMeans – on English, French, Mandarin, German, and Wolof evaluations from Track 2 of the ZeroSpeech Challenge [22].

We make the following contributions. (1) We show that the combination of good word boundaries with high-quality self-supervised segmental features can compete with dynamic programming methods while being much faster. (2) We introduce ES-KMeans+, an updated version of ES-KMeans that is efficient and achieves some of the best results on the ZeroSpeech benchmarks. (3) We investigate how the pre-training languages in the self-supervised speech models affects our simple method. We find that language-specific models trained on the target language outperform multilingual models.

II. Prominence-Based Boundaries with Clustering

Our simple full-coverage unsupervised word segmentation system consists of two components. First, we determine word boundaries using a simple prominence-based approach. Second, we cluster these predicted word-like units to build a lexicon.

For word boundary detection, we follow the method of Pasad et al. [21]. This lightweight method extracts high-quality word boundaries without the need for training an explicit boundary detection model. Speech utterances are first encoded by extracting features from an intermediate HuBERT [23] layer, which we denote as $\mathbf{y}_{1:T} = \mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_T$. To predict word boundaries, the dissimilarity between neighboring frames is calculated using the cosine distance $f_t = d(\mathbf{y}_{t+1}, \mathbf{y}_t)$ between adjacent frames. Hereafter, a smoothing function, using a moving average, is applied to the dissimilarity curve. (For this

This work was supported through a grant from Fab Inc.

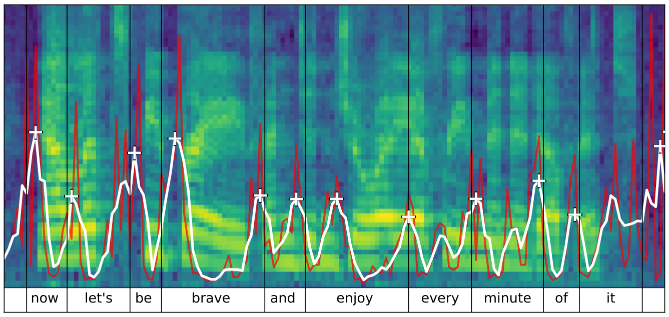

Fig. 1: An example of word boundaries from the prominence-based approach of [21]. The red (dark) line is the dissimilarity curve between adjacent frames, which is smoothed to produce the white line. The crosses are the predicted boundaries. The black vertical lines are the ground truth boundaries.

step, the HuBERT features are mean and variance normalized beforehand.)

Peaks on the smoothed dissimilarity curve, as seen in Fig. 1, are marked as word boundaries when the dissimilarity at frame t is greater than some prominence threshold. Using prominence instead of a hard dissimilarity threshold provides more nuanced boundaries since the prominence of the curve indicates how dissimilar the current frames are relative to the dissimilarity of the surrounding frames. The method works on the principle that frames within the same word are close to each other, while frames at word boundaries are further away from each other. The features are crucial as HuBERT focuses on encoding phonetic information while throwing away speaker-specific information [23], [24].

The lexicon building step, illustrated in Fig. 2, takes the word boundaries (the dashed lines on the left side of the figure) and their corresponding speech utterances as input. Again, utterances are encoded with an intermediate HuBERT layer, resulting in speech features in a high dimensional space $\mathbf{y}_t \in \mathbb{R}^D$ (a in the figure). For clustering, working with these high-dimensional features can become computationally expensive. Therefore, we apply PCA dimensionality reduction (b) to the features, reducing them to a lower dimensional space $\mathbf{x}_t \in \mathbb{R}^M$, with $M < D$, without loosing much phonetic information [25]. In this case, the HuBERT features are not normalized and can come from a different layer than the one used for boundary detection, i.e. we overload the symbol $\mathbf{y}$.

To cluster the variable duration word segments, we turn to acoustic word embeddings, which map variable-length speech segments to fixed dimensional vectors [26]–[28]. We specifically follow the simple approach of [25], where an embedding is obtained by averaging the features in the predicted word segment. Concretely, a word segment $\mathbf{x}_{t_1:t_2}$ is transformed into a fixed dimensional embedding vector $\mathbf{z}_i = g(\mathbf{x}_{t_1:t_2})$, where g represents averaging followed by normalization to the unit sphere. The result is a set of embeddings $\mathbf{z}_i \in \mathbb{R}^M$ (c in Fig. 2) that are clustered using K-means clustering (d). Our implementation uses the efficient FAISS[1] library for clustering. As illustrated on the right side of the figure, each word segment is assigned to a class k whose centroid, $\boldsymbol{\mu}_k$, is closest to the segment embedding.

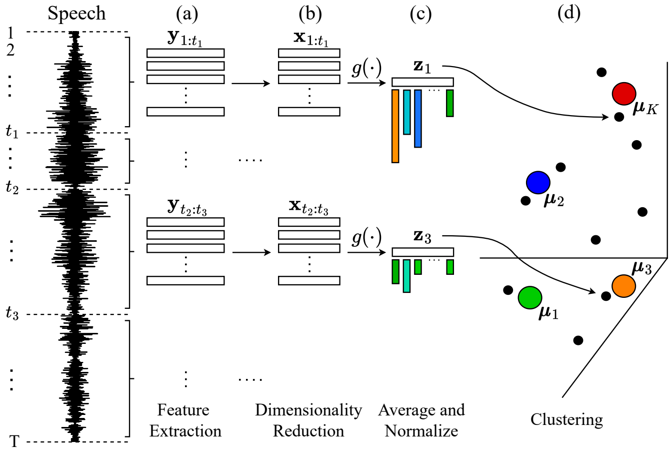

Fig. 2: Our lexicon building step. After extracting frame-level features (a), PCA dimensionality reduction is applied (b). For each segment from the prominence-based approach (Fig. 1), an averaged embedding is obtained (c). These are K-means clustered (d) to get a lexicon.

III. Dynamic Programming Methods

We will compare our approach to state-of-the-art dynamic progamming based methods. Duration-penalized dynamic programming (DPDP) is a two-stage method that first tokenizes input speech into phone-like units and then segments the phone tokens into word-like units [14]. The original method did word segmentation without lexicon learning, but [15] extended DPDP using K-means on averaged HuBERT features to get a lexicon.

In [14] and [15], DPDP is compared to the much older embedded segmental K-means (ES-KMeans) method [20]. In ES-KMeans, potential word boundaries are iteratively clustered and re-segmented using dynamic programming until the best boundaries are selected to form the final word-like segments. To constrain the huge number of possible boundary positions, potential word boundary positions are determined by SylSeg [16], an unsupervised syllabification method using signal processing techniques. The original ES-KMeans uses subsampled MFCC features to represent the word-like segments. Although DPDP outperforms ES-KMeans, this comparison is unfair due to the outdated nature of ES-KMeans. To level the playing field, we improve the components of ES-KMeans and call our updated approach ES-KMeans+.

Concretely, we replace the SylSeg boundary constraints with the prominence-based approach described in Section II. We replace MFCCs with HuBERT features and follow the same clustering approach as in Fig. 2 (PCA projected segment features are averaged, normalized, and then iteratively clustered). We change the original ES-KMeans algorithm which iterated

[1] https://github.com/facebookresearch/faiss

TABLE I: Performance (%) of prominence segmentation with clustering on English HuBERT, and other state-of-the-art methods for word segmentation and lexicon building on Track 2 of the ZeroSpeech Challenge.

	English			French			Mandarin			German			Wolof		
Model	NED	R-val.	Token F_1	NED	R-val.	Token F_1	NED	R-val.	Token F_1	NED	R-val.	Token F_1	NED	R-val.	Token F_1
VG-HuBERT [29]	41.0	59.8	**24.0**	62.0	44.0	15.0	73.0	32.5	19.0	56.0	21.9	**15.0**	92.0	59.7	9.0
DPDP [15]	41.7	**63.2**	19.6	66.0	60.3	11.6	86.0	**67.9**	24.5	56.8	**49.2**	12.5	72.2	66.9	13.1
ES-KMeans [20]	73.2	51.5	19.2	68.7	37.2	6.3	88.1	23.3	8.1	66.2	17.1	11.5	72.4	58.3	10.9
ES-KMeans+ [ours]	33.5	50.0	14.7	**43.2**	56.3	**20.0**	**65.5**	56.9	**25.1**	**42.8**	25.5	9.7	**56.2**	**69.4**	**24.5**
Prom. Seg. Clus. [ours]	**32.9**	60.6	**24.0**	47.9	**61.0**	17.2	71.4	58.2	22.7	44.3	41.8	10.9	59.3	67.9	19.5

over individual utterances to instead operate over batches. This allows us to again use the efficient FAISS library for clustering. The benefits of each of these changes were verified in developmental experiments, leading to a much improved and scalable implementation.

After these updates, ES-KMeans+ and the prominence-based approach of Section II are very similar. The latter corresponds to the first iteration of ES-KMeans+, just before word segmentation is performed. The prominence-based approach is therefore much faster, but ES-KMeans+ has the benefit that it can decide to remove prominence-detected boundaries if these are poorly matched to the clustering model.

IV. Experimental Setup

We perform our main experiments on Track 2 of the ZeroSpeech Challenge [22]. This covers five languages: English, French, Mandarin, German, and Wolof, respectively consisting of 45, 24, 2.5, 25, and 10 hours of speech. The challenge encourages participants to develop language-invariant methods and is one of the most comprehensive benchmarks for unsupervised speech systems. For development, we use the dev-clean subset of LibriSpeech [30] with 5.4 hours of English speech from 40 speakers.

Word segmentation is evaluated using R-value and token F_1, where higher is better for both. R-value measures how close hypothesized word boundaries are to an ideal operating point with a 100% hit-rate and 0% over-segmentation [31]. Token F_1 evaluates how well the hypothesized word tokens match ground truth tokens, requiring both predicted boundaries to be correct in order to receive credit.

The quality of a lexicon is evaluated using normalized edit distance (NED) [32], which relies on phonemic transcriptions found by forced alignments. Discovered word tokens are mapped to their overlapping phoneme sequence, and the NED is calculated between all phoneme sequences of the segments within each cluster. Lower NED is better.

For both the prominence-based word segmentation step of Section II and for the potential boundary set of ES-KMeans+ in Section III, we extract features from the 9th HuBERT layer, based on experiments in [21]. The smoothing window controls how emphasized the dissimilarity curve is, while the prominence threshold determines which peaks are chosen as word boundaries. We set these parameters by finding a balance between NED and R-value on the development data. For our prominence-based approach, we select a four-frame window with a 0.75 prominence threshold, while for ES-KMeans+ we opt for a five-frame window with a threshold of 0.3. The high-recall hyperparameter setting for ES-KMeans+ creates more word boundaries than are needed, enabling the method to choose the best subset of these boundaries.

When clustering, both in our prominence-based method and ES-KMeans+, we extract features from the 12th HuBERT layer and reduce the feature dimensionality to 250 dimensions using PCA. These settings are based on development experiments. To enable a fair comparison to previous work, we use the same number of K-means clusters as in the original ES-KMeans and DPDP papers: we use 43k, 29k, 3k, 29k, and 3.5k clusters for English, French, Mandarin, German, and Wolof.

We compare our system to the dynamic programming methods of Section III: DPDP, ES-KMeans, and ES-KMeans+. We also compare to the visually-grounded HuBERT (VG-HuBERT) method [29] that pairs unlabeled speech with images.

V. Experimental Results

We start our experiments with a comparison to previous full-coverage unsupervised segmentation and clustering systems, with a specific focus on comparing our new simple approach to the previous dynamic programming methods that inspired it. We then look at the impact of different design choices within our simple method, and finally consider the impact of using an English self-supervised speech model on non-English data.

A. Comparison to Other Systems

The performance of all systems can be seen in Table I. When comparing the prominence-based method to ES-KMeans+, we see a tradeoff between NED and R-value: the simple method achieves better R-value on all languages except Wolof, while ES-KMeans+ gives better NED in all cases but English. Compared to the other approaches, both our systems achieve several state-of-the-art results and improve upon ES-KMeans. Prominence segmentation achieves a good tradeoff between metrics, finding a middle ground between ES-KMeans+ and DPDP (which consistently achieves a high R-value).

One limitation of the prominence-based method is its dependence on the quality of the word segmentation step since it cannot remove bad boundaries like DPDP or ES-KMeans

TABLE II: Runtime (min) of prominence segmentation with clustering and ES-KMeans+.

Model	English	French	Mandarin	German	Wolof
ES-KMeans+ [ours]	765	330	5	316	8
Prom. Seg. Clus. [ours]	**146**	**68**	**1**	**50**	**3**

TABLE III: Ablation scores (%) for the main components of prominence segmentation with clustering on LibriSpeech dev-clean. A boundary tolerance of 20 ms is allowed.

Components	NED	R-value	Token F_1
MFCC + Prom. Seg.	75.3	**50.7**	**15.6**
HuBERT + SylSeg	41.8	39.1	7.5
HuBERT + Prom. Seg.	**40.4**	**50.7**	**15.6**

can. But, while ES-KMeans+ is marginally better than the prominence-based approach on several metrics, Table II shows that ES-KMeans+ is four to five times slower. This is because ES-KMeans+ applies dynamic programming iteratively, while the simple approach only segments and clusters once.

B. Effect of our Design Choices

Our simple approach gives competitive performance by combining good word boundaries with good self-supervised features for clustering (last line, Table I). To support this argument, we look at how performance is impacted when an older boundary detection method is used or when conventional speech features are employed. The last line of Table III gives the performance of our final system. The first line is a system using subsampled MFCCs (as in the original ES-KMeans) instead of self-supervised features. This comparison shows that modern features are much better for lexicon building: NED worsens by 34.9% absolute when using MFCCs. A comparison between lines two and three also shows that the prominence-based approach of [21] performs much better than when the SylSeg boundaries are used. The combination of improved boundaries and improved features are therefore what leads to the competitive performance of our simple approach. But both of these improvements can actually be attributed to the representation capabilities of self-supervised speech models: SylSeg boundaries are computed directly on the raw speech while the prominence-based approach takes advantage of the dissimilarity of encoded features.

C. Impact of the Languages in HuBERT Pre-Training

Up to now, all experiments were conducted using the English HuBERT model, even when applying the approach to non-English data. To investigate the effect of the feature encoder's pre-training language, we use the recent multilingual HuBERT (mHuBERT) model [33]. Specifically, we inspect the lexicon-building ability of this model compared to the English-specific setup. We find that the 8th mHuBERT layer performs the best overall and, using the same boundaries as in Table I, we cluster the word segments.[2] Figure 3 shows the resulting NED scores.

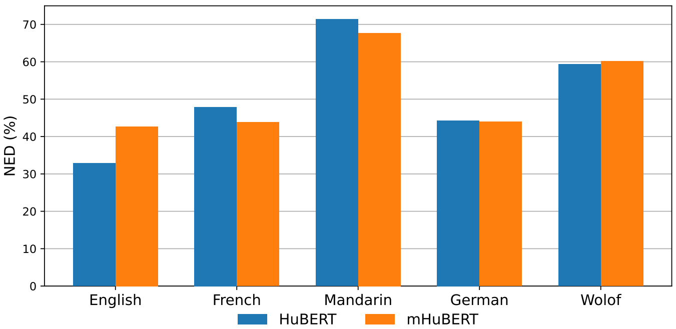

Fig. 3: Normalized edit distance (%) of our prominence-based approach when swapping English HuBERT features for multilingual HuBERT (mHuBERT) features.

For all non-English languages, mHuBERT gives similar or better performance compared to the English model. For English, the language-specific HuBERT performs 9.8% absolute better than mHuBERT. (English is one of the 147 training languages used in mHuBERT.) It therefore seems beneficial to train a self-supervised speech model specifically for the language on which it will be applied. To further investigate this, we perform a test using a Mandarin-specific HuBERT,[3] also extracting features from the 8th transformer layer while keeping the boundaries consistent. Here, the NED improves from 67.7% to 61.9%.[4]

Overall, our results show that there may be small benefits from using multilingual training rather than a monolingual model trained on a language different from the target, but language-specific self-supervised training remains the best. Similar findings have been made in other recent work [7].

VI. Conclusion

This paper showed that unsupervised word segmentation and lexicon learning can be performed competitively by combining a simple boundary detection method with clustering on modern self-supervised features. Concretely, boundary detection is performed using dissimilarities between adjacent self-supervised features and K-means clustering is then performed on averaged features to obtain a lexicon. This simple method was compared to state-of-the-art dynamic programming methods, including our own updated version of the embedded segmental K-means (ES-KMeans) approach. While our ES-KMeans+ method gave new state-of-the-art results on several metrics in the ZeroSpeech benchmarks, our simple prominence-based segmentation and clustering method performed competitively while being much faster. We showed that both the initial boundaries and clustering features are important to achieve good performance. We also showed that even better performance is possible if the self-supervised speech model is trained specifically on data from the particular testing language.

[2] Another option is to use mHuBERT boundaries, but by keeping them constant, we specifically assess the impact of the features on the lexicon.

[3] https://huggingface.co/TencentGameMate/chinese-hubert-base

[4] Open-source HuBERT models are not available for the other languages.

References

[1] O. J. Räsänen, “Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions,” *Speech Communication*, vol. 54, pp. 975–997, 2012.

[2] E. Bergelson and D. Swingley, “At 6–9 months, human infants know the meanings of many common nouns,” *National Academy of Sciences of the United Sates of America*, vol. 190, pp. 3253–3258, 2012.

[3] P. W. Jusczyk, “Some critical developments in acquiring native language sound organization during the first year,” *Annals of Otology, Rhinology & Laryngology*, vol. 189, pp. 11–15, 2002.

[4] E. Dupoux, “Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner,” *Cognition*, vol. 173, pp. 43–59, 2016.

[5] L. Besacier, E. Barnard, A. Karpov, and T. Schultz, “Automatic speech recognition for under-resourced languages: A survey,” *Speech Communication*, vol. 56, pp. 85–100, 2014.

[6] A. S. Park and J. R. Glass, “Unsupervised pattern discovery in speech,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 16, pp. 186–197, 2008.

[7] B. van Niekerk, J. Zaïdi, M.-A. Carbonneau, and H. Kamper, “Spoken-term discovery using discrete speech units,” in *Interspeech*, 2024.

[8] C.-y. Lee, T. O’Donnell, and J. R. Glass, “Unsupervised lexicon discovery from acoustic input,” *Transactions of the Association for Computational Linguistics*, vol. 3, pp. 389–403, 2015.

[9] R. Algayres, T. Ricoul, J. Karadayi, H. Laurençon, S. Zaiem, A. Mohame, B. Sagot, and E. Dupoux, “DP-Parse: Finding word boundaries from raw speech with an instance lexicon,” *Transactions of the Association for Computational Linguistics*, vol. 10, pp. 1051–1065, 2022.

[10] Y. Okuda, R. Ozaki, S. Komura, and T. Taniguchi, “Double articulation analyzer with prosody for unsupervised word and phone discovery,” *IEEE Transactions on Cognitive and Developmental Systems*, vol. 15, pp. 1335–1347, 2022.

[11] A. Jansen *et al.*, “A summary of the 2012 JHU CLSP workshop on zero resource speech technologies and models of early language acquisition,” in *ICASSP*, 2013.

[12] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, “Segmental contrastive predictive coding for unsupervised word segmentation,” in *Interspeech*, 2021.

[13] S. Cuervo, M. Grabias, J. Chorowski, G. Ciesielski, A. Łańcucki, P. Rychlikowski, and R. Marxer, “Contrastive prediction strategies for unsupervised segmentation and categorization of phonemes and words,” in *ICASSP*, 2022.

[14] H. Kamper, “Word segmentation on discovered phone units with dynamic programming and self-supervised scoring,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 31, pp. 684–694, 2023.

[15] H. Kamper and B. van Niekerk, “Revisiting speech segmentation and lexicon learning with better features,” *arXiv preprint arXiv:2401.17902*, 2024.

[16] O. J. Räsänen, G. Doyle, and M. Frank, “Unsupervised word discovery from speech using automatic segmentation into syllable-like units,” in *Interspeech*, 2015.

[17] H. Kamper, A. Jansen, and S. Goldwater, “A segmental framework for fully-unsupervised large-vocabulary speech recognition,” *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

[18] Y.-H. Wang, H.-y. Lee, and L.-s. Lee, “Segmental audio word2vec: Representing utterances as sequences of vectors with applications in spoken term detection,” in *ICASSP*, 2018.

[19] T. S. Fuchs and Y. Hoshen, “Unsupervised word segmentation using temporal gradient pseudo-labels,” in *ICASSP*, 2023.

[20] H. Kamper, K. Livescu, and S. Goldwater, “An embedded segmental K-means model for unsupervised segmentation and clustering of speech,” in *ASRU*, 2017.

[21] A. Pasad, C.-M. Chien, S. Settle, and K. Livescu, “What do self-supervised speech models know about words?” *Transactions of the Association for Computational Linguistics*, vol. 12, pp. 372–391, 2024.

[22] 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 *Interspeech*, 2020.

[23] W. Hsu, B. Bolte, Y. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, “HuBERT: Self-supervised speech representation learning by masked prediction of hidden units,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[24] B. van Niekerk, M.-A. Carbonneau, J. Zaidi, M. Baas, H. Seute, and H. Kamper, “A comparison of discrete and soft speech units for improved voice conversion,” in *ICASSP*, 2022.

[25] R. Sanabria, H. Tang, and S. Goldwater, “Analyzing acoustic word embeddings from pre-trained self-supervised speech models,” in *ICASSP*, 2023.

[26] A. L. Maas, S. D. Miller, T. M. O’neil, A. Y. Ng, and P. Nguyen, “Word-level acoustic modeling with convolutional vector regression,” in *ICML*, 2012.

[27] K. Levin, K. Henry, A. Jansen, and K. Livescu, “Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings,” in *ASRU*, 2013.

[28] H. Kamper, A. Jansen, S. King, and S. Goldwater, “Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings,” in *SLT*, 2014.

[29] P. Peng and D. Harwath, “Word discovery in visually grounded, self-supervised speech models,” in *Interspeech*, 2022.

[30] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “LibriSpeech: An ASR corpus based on public domain audio books,” in *ICASSP*, 2015.

[31] O. J. Räsänen, U. K. Laine, and T. Altosaar, “An improved speech segmentation quality measure: the R-value,” in *Interspeech*, 2009.

[32] B. Ludusan, M. Versteegh, A. Jansen, G. Gravier, X.-N. Cao, M. Johnson, and E. Dupoux, “Bridging the gap between speech technology and natural language processing: an evaluation toolbox for term discovery systems,” in *LREC*, 2014.

[33] M. Z. Boito, V. Iyer, N. Lagos, L. Besacier, and I. Calapodescu, “mHuBERT-147: A compact multilingual HuBERT model,” in *Interspeech*, 2024.