ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3315-6701-9/26/$31.00 ©2026 IEEE | DOI: 10.1109/ICASSP55912.2026.11464450

UNSUPERVISED LEXICON LEARNING FROM SPEECH IS LIMITED BY REPRESENTATIONS RATHER THAN CLUSTERING

Danel Slabbert, Simon Malan, Herman Kamper

Electrical and Electronic Engineering, Stellenbosch University, South Africa

ABSTRACT

Zero-resource word segmentation and clustering systems aim to tokenise speech into word-like units without access to text labels. Despite progress, the induced lexicons are still far from perfect. In an idealised setting with gold word boundaries, we ask whether performance is limited by the representation of word segments, or by the clustering methods that group them into word-like types. We combine a range of self-supervised speech features (continuous/discrete, frame/word-level) with different clustering methods (k-means, hierarchical, graph-based) on English and Mandarin data. The best system uses graph clustering with dynamic time warping on continuous features. Faster alternatives use graph clustering with cosine distance on averaged continuous features or edit distance on discrete unit sequences. Through controlled experiments that isolate either the representations or the clustering method, we demonstrate that representation variability across segments of the same word type—rather than clustering—is the primary factor limiting performance.

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

1. INTRODUCTION

Unsupervised word discovery aims to discover word-like units from unlabelled speech by locating word boundaries and clustering the resulting segments into hypothesised lexical categories [1]. This is difficult because, unlike text, speech is continuous and lacks explicit word delimiters [2]. Moreover, different instances of the same word exhibit substantial acoustic variability, even when produced by the same speaker. Despite these challenges, human infants can discriminate between words in their native language within their first year [3–5]. Developing computational models that mimic this ability could therefore both inform theories of language acquisition [6] and support the development of low-resource speech technology [7].

Early approaches to unsupervised word discovery sought to identify recurring patterns across speech utterances, often relying on dynamic time warping (DTW) [8]. Despite advances [9, 10], these techniques still struggle to segment the entire input into meaningful units. To address this, full-coverage methods have been proposed which aim to tokenise all of the input speech into word-like units [11–20].

Full-coverage systems can typically be broken into three components: an unsupervised word boundary detection component, a feature extraction method for representing word-like segments, and a clustering method used to group the hypothesised segments into a lexicon of word-like types. To assess the influence of boundary detection, Malan et al. [21] performed word discovery using ground-truth word boundaries. Even in this idealised setting, the resulting lexicon is still far from perfect. Since true boundaries were used, the errors stem either from the representation or clustering method.

This paper investigates these limitations by revisiting the task of unsupervised lexicon learning, where unlabelled word segments with true boundaries need to be clustered into hypothesised word types. We combine a range of representations from self-supervised learned (SSL) speech models with different clustering methods. We consider both discrete and continuous frame-level features. These are either kept in sequence form to represent word segments, with distances between sequences used for clustering, or averaged to obtain fixed-dimensional acoustic word embeddings [22, 23]. For clustering, we consider k-means, agglomerative, BIRCH, and graph clustering. In experiments on English and Mandarin, our best system uses graph clustering with DTW over continuous sequences. While producing a better lexicon than previous work [21], the result remains imperfect, with word-level purity below 90%. This system is also much slower than those using averaged embeddings or discrete sequences.

To definitively answer whether it is the representations or clustering that limits performance, we conduct two controlled experiments. First, we reduce variability in clustering by forcing all instances of a word into the same cluster at initialisation. Second, we eliminate representational inconsistency by artificially fixing all instances of the same word type to near-identical representations. This allows the influence of speech representations and clustering methods to be assessed individually. The results show that, despite substantial improvements, current speech representations, not clustering methods, constrain lexicon quality. Thus, further research on SSL features is needed to make progress on unsupervised word discovery.

2. METHODS

We investigate several combinations of speech representation and clustering methods to understand the current landscape of unsupervised lexicon learning. The systems are shown in Figure 1. Apart from [21], who implicitly looked at lexicon learning, work on this task is limited. One exception is [24], which applied different clustering approaches on acoustic word embeddings. But this was done using conventional features in a time before neural representations.

2.1. Representations

Given ground truth word segments (Figure 1-a), we start by extracting continuous features from intermediate layers of SSL speech models (Figure 1-b-i). We consider a range of SSL models, all trained using a masked prediction pretext task [25–28]. The different SSL models are described in detail in Section 3.2. As an alternative to continuous features, discretised representations are obtained by quantising continuous features, producing a lower-bitrate encoding which reduces downstream computational load (Figure 1-b-ii). Previous work [10] showed that quantisation can remove speaker-specific information from continuous features, potentially improving consistency in representations used for lexicon learning.

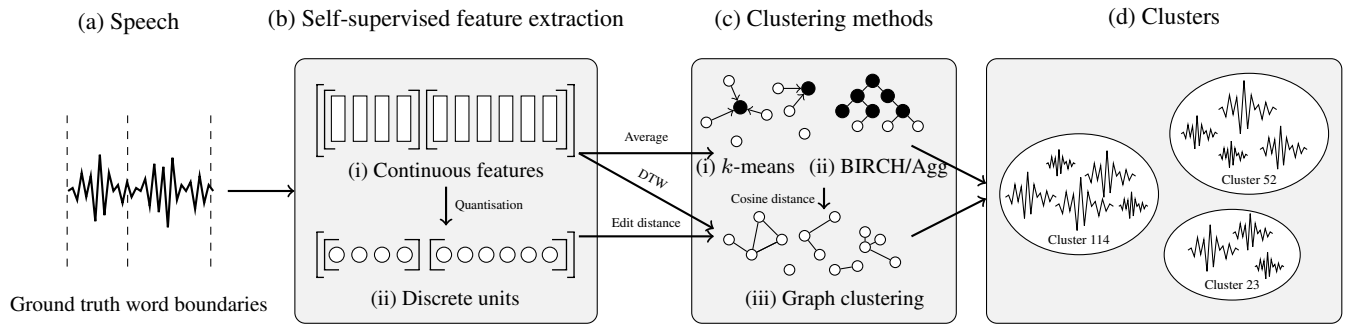

Fig. 1. We consider different systems of (b) speech representations and (c) clustering approaches in order to learn (d) a lexicon from (a) unlabelled speech. For this study, we assume that true word boundaries are known.

While some clustering approaches can be applied directly to feature sequences, others require word segments to be represented as single fixed-dimensional vectors. A common approach is to use an acoustic word embedding obtained by averaging the features of a word segment across the temporal dimension [22]. While simple, several studies have shown that averaging appropriate SSL features results in robust acoustic word embeddings [23, 29, 30].

2.2. Clustering Methods

We couple the different representations with a range of clustering methods (Figure 1-c), resulting in six lexicon learning systems. The first four systems cluster averaged acoustic word embeddings using one of the following methods (Figure 1-c): k-means clustering; BIRCH clustering [31], an efficient tree-based algorithm; agglomerative hierarchical clustering [32]; or graph clustering, where word segments are represented as nodes and edges are weighted by the cosine similarity between their embeddings.

The two remaining systems also use graph clustering, but operate on feature sequences, retaining temporal information. The distance metric used for graph construction corresponds to the type of representations. For continuous features, pairwise distances are computed using DTW, which aligns sequences of varying lengths and calculates the cost of the optimal alignment (Figure 1-b-i to Figure 1-c-iii). For discrete unit sequences, edit distance is used, which measures the minimum number of operations required to convert one sequence into another (Figure 1-b-ii to Figure 1-c-iii). For all graph clustering systems, a graph is incrementally constructed with edges inserted between nodes when the similarity exceeds a predefined threshold. The graph is then partitioned using the efficient Leiden algorithm [33] with the constant Potts model [34] as clustering objective.

3. EXPERIMENTAL SETUP

3.1. Data and Evaluation

We use LibriSpeech [35] dev-clean for development and test-clean for final evaluations. Each set contains 5.4 hours of English speech from 40 speakers. Word boundaries are obtained from forced alignments [36]. For the Mandarin evaluation, we use data from Track 2 of the ZeroSpeech challenge [37], consisting of 2.5 hours of speech from 12 speakers with word alignments. For the Mandarin experiments, the optimal hyperparameters from the English development experiments are retained as is.

We evaluate lexicon quality using a range of standard metrics. Normalised edit distance (NED) [1] measures the phonetic purity of clusters by calculating the length-normalised edit distance between phonetic transcriptions (from forced alignments) of all word segment pairs within a cluster. A single average over all clusters gives the final NED; lower is better. Purity measures how homogeneous each cluster is with respect to the true word labels. For every cluster, the most frequent word type is identified, and the number of segments belonging to that type is counted. These are summed over all clusters and divided by the total number of segments in the dataset. V-measure [38] is the harmonic mean of homogeneity (a purity-like metric) and completeness (measuring the degree to which segments of the same type are assigned to the same cluster). Higher purity and V-measure are better. Bitrate is the average data rate of the encoded output, measured in bits per second (bits/s); lower is better.

Purity and NED can be artificially improved by increasing the number of clusters. To enable a fair comparison between methods, and since we are operating in an idealised setting, we fix the number of clusters to the true number of word types in each dataset. The number of clusters for LibriSpeech dev-clean and test-clean is 8,216 and 8,006, respectively, while for Mandarin it is 8,871. Graph clustering does not take the number of clusters as an explicit input but instead infers it from the data; we tune the hyperparameters to get approximately the desired number of clusters.

3.2. Implementation: Representations and Clustering Methods

We compare speech representations from a range of SSL models, all producing 20 ms frames. To test state-of-the-art features [39, 40], we use HuBERT Large [25] and WavLM Large [26]. A HuBERT Base model fine-tuned for voice-conversion, HuBERT Soft [27], is also considered for its ability to preserve phonetic content while suppressing speaker-specific information: its continuous features are extracted from a linear projection layer that predicts a distribution over discrete units, thereby providing a middle-ground between continuous and discrete features. To assess the effects of pre-training language, we use the multilingual HuBERT (mHuBERT) [28] pre-trained on 90k hours across 147 languages (including English and Mandarin). For the Mandarin tests, we include a HuBERT Large model pre-trained on 10k hours of Mandarin.[1] For all large SSL models, 1,024-dimensional features are extracted from the 21st layer based on its ability to capture word information [39, 40]. Following [20], 768-dimensional mHuBERT features are extracted from the eighth layer.

All these features are subsequently mean and variance normalised, and then projected to 350 dimensions using principal component analysis (PCA). This gave the best performance on dev-clean, and

[1] https://huggingface.co/TencentGameMate/chinese-hubert-large

improved on the results from [20, 21], where 250 dimensions were used. For HuBERT Soft, features from the 12th layer are pushed through the projection layer, resulting in 256-dimensional soft features, which are mean and variance normalised. For each SSL model, acoustic word embeddings are obtained by averaging the different low-dimensional features and then normalising to the unit sphere.

Discrete features are obtained by applying k-means clustering with 500 clusters on the different SSL representations (without normalisation or PCA). These unit extraction models are trained on 50 hours of audio from LibriSpeech train-clean for English data, or on the complete dataset for the Mandarin data. Following [19, 41], we use duration-penalised dynamic programming to smooth the resulting sequences by discouraging rapid unit changes. The unit sequences are not deduplicated, which worked better in development experiments.

We first describe the clustering methods that learn a lexicon on averaged acoustic word embeddings. For k-means, we use the efficient FAISS library.[2] But unlike [21], we switch the default random initialisation to k-means++ initialisation [42]. This selects the first centroid at random, with each subsequent centroid chosen from the remaining data points with probability proportional to the squared distance from the closest existing centroid. This alternative initialisation gave large improvements over [20, 21] on development data. BIRCH and agglomerative clustering are implemented using scikit-learn.[3] We apply BIRCH with a distance threshold of 0.25. Agglomerative clustering uses Ward linkage [43], which merges clusters by attempting to minimise the total within-cluster variance.

Graph clustering can be used with either averaged acoustic embeddings or feature sequences. We use the efficient igraph library [44]. Distance-based thresholds are applied globally: nodes with no edges remain isolated. Thresholds are tuned based on memory constraints and development performance, with values of 0.65 for edit distance graphs, 0.4 for cosine distance graphs, and 0.35 for DTW graphs.

4. EXPERIMENTAL RESULTS

4.1. A First Evaluation: Representations

Before comparing the various combinations of representations and clustering methods, we start by selecting the best-performing SSL features. Table 1 reports English development scores using two representative systems: continuous averaged features which are k-means clustered (top), and discrete unit sequences which are graph clustered using edit distance (bottom).

Systems using WavLM Large features perform well, especially in its continuous form (Table 1-top) where it outperforms all other models. In the discrete graph clustering system, HuBERT Large also performs well. In contrast, mHuBERT generally results in worse lexicons, indicating that diluting English pre-training data with other languages worsens performance compared to using English-only models. Although HuBERT Soft is competitive in the continuous setting, the discretised version (bottom) performs much worse. Based on these results, we use WavLM Large features for the subsequent lexicon learning experiments on English data.

4.2. Evaluation on English: Representations with Clustering

Table 2 shows the lexicon learning performance on English test data of the different representation–clustering systems using WavLM Large features (continuous or discrete). Continuous averaged features with graph clustering achieves the best purity (89.6%), V-measure (90.3%), and bitrate (35.6). The continuous features with DTW and graph clustering yield the lowest NED (5.2%), but at a substantial computational cost, being around 250 times slower than alternative systems. These results all improve on the continuous averaged k-means approach from [20, 21], which achieves a NED of 17.3%, purity of 78.2%, V-measure of 80.0%, and a bitrate of 41.4 (after altering the experimental setup to exactly match the protocol here).

In general, continuous representations outperform discrete units on purity and NED, with the discrete graph system only being competitive in V-measure (88.4%). Of the different clustering methods, graph-based clustering consistently yields higher V-measure scores than simpler methods like k-means or agglomerative clustering. Bitrate indicates that graph clustering produces a more compact encoding output, especially when used with the averaged embeddings.

Although all systems deliver competitive lexicons and improve on the results from [20, 21], none of them achieve a perfect lexicon, even when provided with a ground-truth word segmentation: NED shows that the clustered sequences still have phonetic differences, and none of the systems achieve perfect purity or completeness. Before we investigate what limits performance, we see if this is also true on another language, Mandarin. Apart from being in a different language family, the SSL representations available in Mandarin are also different from those available in English.

4.3. Evaluation on Mandarin

Table 3 shows how well our systems developed on English data generalise to Mandarin data. We also evaluate the effect of an SSL

Table 1. Lexicon quality (%) on LibriSpeech dev-clean when different SSL features are used in two representative systems, one using continuous and the other discrete features.

Features	NED	Purity	V-measure
Continuous + average + k-means			
WavLM Large	**7.4**	**89.3**	**83.7**
HuBERT Large	9.3	89.0	83.6
HuBERT Soft	10.0	85.0	83.1
mHuBERT	10.8	83.4	82.2
Discrete + edit distance + graph clustering			
WavLM Large	**7.3**	83.3	88.6
HuBERT Large	7.8	**85.0**	**89.8**
HuBERT Soft	23.5	59.6	78.9
mHuBERT	29.7	61.0	79.1

Table 2. Lexicon quality on LibriSpeech test-clean in terms of NED (%), purity (%), V-measure (%), bitrate (bits/second), and runtime (seconds) for six different representation–clustering systems.

System	NED	Purity	V-m	Bitrate	Runtime
continu + avg + k-means	8.6	88.4	83.6	40.9	281.0
continu + avg + BIRCH	6.8	89.5	84.1	41.0	**415.0**
continu + avg + agglom	6.8	89.5	84.1	40.9	433.0
continu + avg + graph	6.7	**89.6**	**90.3**	**35.6**	484.0
continu + DTW + graph	**5.2**	89.3	89.1	36.6	123,630.9
discrete + edit + graph	7.9	83.0	88.4	36.9	1,526.6

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

[3]https://scikit-learn.org

Table 3. Lexicon quality (%) on Mandarin data using different SSL features in a subset of relevant systems.

Features	NED	Purity	V-measure
Continuous + average + k-means			
WavLM Large	52.7	63.3	87.9
mHuBERT	46.9	66.6	88.6
Mandarin HuBERT Large	**16.8**	**80.5**	**92.1**
Continuous + average + cosine distance + graph clustering			
WavLM Large	43.4	64.3	88.5
mHuBERT	39.1	67.5	89.3
Mandarin HuBERT Large	**4.9**	**82.8**	**94.2**
Continuous + DTW distance + graph clustering			
WavLM Large	33.2	68.3	89.6
mHuBERT	30.3	71.5	90.5
Mandarin HuBERT Large	**5.8**	**82.0**	**93.6**

Table 4. Lexicon quality (%) on LibriSpeech test-clean when the effect of representations or clustering methods is isolated.

System	NED	Purity	V-measure
Continuous + average + k-means			
Baseline	8.6	88.4	83.6
Perfect cluster initialisation	17.0	81.5	81.3
Perfect word embeddings	12.1	**100.0**	**100.0**
Discrete + edit + graph clustering			
Baseline	7.9	83.0	88.4
Perfect cluster initialisation	7.4	83.6	88.7
Perfect word representations	12.1	**100.0**	**100.0**

model's pre-training language by testing with models trained on different amounts of Mandarin data. Concretely, we compare the English-only WavLM Large [26], the multilingual mHuBERT [28] including limited Mandarin data, and the Mandarin HuBERT Large trained entirely on Mandarin (see Section 3.2 for details). We show results for the k-means system representative of previous work [21], together with the two systems that worked best on English (Table 2).

Across the different systems, the English-only WavLM performs worst, the mHuBERT being exposed to some Mandarin gives intermediate results, and the Mandarin HuBERT performs the best by a substantial margin. In particular, Mandarin HuBERT features achieve a NED of 4.9% when averaged and graph clustered. This results in a lexicon of comparable quality to the best English lexicon (Table 2) and shows that including more of the target language in pre-training improves lexicon quality. This underscores the importance of language-specific SSL representations [10, 20].

Although we show that quality lexicons can be obtained on a different language, the results here corroborate those on English where even the strongest systems produce lexicons with a NED of around 5% and word-level purity of between 80% and 90%. Below we examine the source of this limitation.

5. REPRESENTATIONS VS CLUSTERING

Given that we are using true word boundaries, lexicon imperfections stem from clustering methods that fail to group instances of the same word together and/or representations that fail to capture similarity across different instances of the same word. In Table 4 we present controlled experiments on LibriSpeech test-clean where we idealise either the clustering method or representations in two representative systems. The baseline results are from Table 2.

In the second row of each section in the table, the two clustering algorithms are initialised so that all instances of the same word type are assigned to the same cluster. For k-means (top section), this means that centroids are initialised at the mean embeddings of segments corresponding to the same word type. For graph clustering (bottom), segments are initially assigned to clusters based on their true word types. At initialisation, both systems have a purity and V-measure of 100%. However, after the methods are run from their perfect starting points, the resulting lexicons are far from ideal: purity and V-measure drop to baseline levels. From the k-means system we conclude that embeddings of the same word type are not close to all the other instances of that word. Similarly, the discrete system shows that unit sequences vary across instances of the same word type.

Next we consider the case where representations are ideal and clustering is performed as usual. For continuous features (top section), each word type is represented by its mean acoustic word embedding with a small amount of random noise added. For the discrete unit sequences (bottom), a single sequence is randomly selected to represent all instances of that word type. Results are given in the third row of each section in Table 4. We see that, in both cases, a perfect lexicon is produced with word-level purities and V-measures of 100%. In both cases, NED is worse than the baselines, dropping from roughly 8% to 12%. This is because phonetic variation, even in clusters consisting of only one word type, limits NED as a word-level metric. Conversely, word-level metrics, such as purity, ignore phonetic detail. This is why we report a range of complementary metrics.

Pronunciation and co-articulation not only affects NED but also lead to differences in the representations, which in part explains the poor performance when initialising the clustering methods perfectly (the second rows in the two sections). But in the k-means systems we see that perfect initialisation actually leads to worse performance than the non-ideal baseline, e.g. purity dropping from 88.4% to 81.5%. Together these findings suggest that clustering methods, particularly graph clustering, perform robustly, but inconsistency in the representation of spoken words hinder effective lexicon learning. If consistent representations aligned with word types were available, perfect word-level purity and V-measure would be achievable.

6. CONCLUSION

This paper compared six systems for lexicon learning that combine different feature representations and clustering approaches, achieving clear improvements over previous methods. These results summarise the current landscape of lexicon learning in terms of accuracy and computational cost. In controlled experiments, we showed that if representations are idealised, existing clustering methods can learn perfect lexicons. The current bottleneck is therefore not in clustering, but in the consistency of feature representations. Future work will explore how these insights can be applied to the training of self-supervised speech models to improve the resulting representations. We will also consider how our best systems can be extended for true word discovery, where boundaries are unknown.

7. REFERENCES

[1] B. Ludusan et al., "Bridging the gap between speech technology and natural language processing: An evaluation toolbox for term discovery systems," in *LREC*, 2014.

[2] O. Räsänen, "Computational modeling of phonetic and lexical learning in early language acquisition: Existing models and future directions," *Speech Communication*, 2012.

[3] J. R. Saffran et al., "Statistical learning by 8-month-old infants," *Science*, 1996.

[4] P. W. Jusczyk, "Some critical developments in acquiring native language sound organization during the first year," in *The Annals of Otology, Rhinology and Laryngology*, 2002.

[5] E. Bergelson and D. Swingley, "At 6-9 months, human infants know the meanings of many common nouns," *PNAS*, 2012.

[6] E. Dupoux, "Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner," *Cognition*, 2018.

[7] L. Besacier et al., "Automatic speech recognition for under-resourced languages: A survey," *Speech Communication*, 2014.

[8] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech and Language Processing*, 2008.

[9] O. Räsänen and M. A. C. Blandón, "Unsupervised discovery of recurring speech patterns using probabilistic adaptive metrics," in *Interspeech*, 2020.

[10] B. van Niekerk et al., "Spoken-term discovery using discrete speech units," in *Interspeech*, 2024.

[11] C. Lee et al., "Unsupervised lexicon discovery from acoustic input," *Transactions of the Association for Computational Linguistics*, 2015.

[12] O. Räsänen et al., "Unsupervised word discovery from speech using automatic segmentation into syllable-like units," in *Interspeech*, 2015.

[13] H. Kamper et al., "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *ASRU*, 2017.

[14] S. Bhati et al., "Self-expressing autoencoders for unsupervised spoken term discovery," in *Interspeech*, 2020.

[15] S. Bhati et al., "Segmental contrastive predictive coding for unsupervised word segmentation," in *Interspeech*, 2021.

[16] R. Algayres et al., "DP-Parse: Finding word boundaries from raw speech with an instance lexicon," *Transactions of the Association for Computational Linguistics*, 2022.

[17] S. Cuervo et al., "Contrastive prediction strategies for unsupervised segmentation and categorization of phonemes and word," in *ICASSP*, 2022.

[18] Y. Okuda et al., "Double articulation analyzer with prosody for unsupervised word and phone discovery," *IEEE Transactions on Cognitive and Developmental Systems*, 2023.

[19] H. Kamper, "Word segmentation on discovered phone units with dynamic programming and self-supervised scoring," *IEEE/ACM Transactions on Audio Speech and Language Processing*, 2023.

[20] S. Malan et al., "Unsupervised word discovery: Boundary detection with clustering vs. dynamic programming," in *ICASSP*, 2025.

[21] S. Malan et al., "Should top-down clustering affect boundaries in unsupervised word discovery?," *arXiv preprint: arXiv:2507.19204*, 2025.

[22] K. Levin et al., "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *ASRU*, 2013.

[23] R. Sanabria et al., "Analyzing acoustic word embeddings from pre-trained self-supervised speech models," in *ICASSP*, 2023.

[24] H. Kamper et al., "Unsupervised lexical clustering of speech segments using fixed-dimensional acoustic embeddings," in *SLT*, 2014.

[25] W. N. Hsu et al., "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio Speech and Language Processing*, 2021.

[26] S. Chen et al., "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal on Selected Topics in Signal Processing*, 2022.

[27] B. van Niekerk et al., "A comparison of discrete and soft speech units for improved voice conversion," in *ICASSP*, 2022.

[28] M. Z. Boito et al., "mHuBERT-147: A compact multilingual HuBERT model," in *Interspeech*, 2024.

[29] C. Jacobs and H. Kamper, "Leveraging multilingual transfer for unsupervised semantic acoustic word embeddings," *IEEE Signal Processing Letters*, 2024.

[30] J. Herreilers et al., "Low-resource keyword spotting using contrastively trained transformer acoustic word embeddings," *arXiv preprint arXiv:2506.17690*, 2025.

[31] T. Zhang et al., "BIRCH: An efficient data clustering method for very large databases," *ACM SIGMOD Record*, 1996.

[32] F. Nielsen, *Introduction to HPC with MPI for Data Science*, chapter Hierarchical Clustering, pp. 195–211, Springer, 2016.

[33] V. Traag et al., "From Louvain to Leiden: Guaranteeing well-connected communities," *Scientific Reports*, 2019.

[34] J. Reichardt and S. Bornholdt, "Detecting fuzzy community structures in complex networks with a potts model," *Physical Review Letters*, 2004.

[35] V. Panayotov et al., "LibriSpeech: An ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[36] M. McAuliffe et al., "Montreal forced aligner: Trainable text-speech alignment using kaldi," in *Interspeech*, 2017.

[37] E. Dunbar et al., "The Zero Resource Speech Challenge 2017," in *ASRU*, 2017.

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

[39] A. Pasad, B. Shi, and K. Livescu, "Comparative layer-wise analysis of self-supervised speech models," in *ICASSP*, 2023.

[40] A. Pasad et al., "What do self-supervised speech models know about words?," *Transactions of the Association for Computational Linguistics*, 2024.

[41] N. Visser and H. Kamper, "Spoken language modeling with duration-penalized self-supervised units," in *Interspeech*, 2025.

[42] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA*, 2007.

[43] F. Murtagh and P. Legendre, "Ward's hierarchical agglomerative clustering method: Which algorithms implement Ward's criterion?," *Journal of Classification*, 2014.

[44] G. Csardi and T. Nepusz, "The igraph software package for complex network research," *InterJournal Complex Systems*, 2006.