Interspeech 2024
1-5 September 2024, Kos, Greece

Spoken-Term Discovery using Discrete Speech Units

Benjamin van Niekerk[1], *Julian Zaïdi*[2], *Marc-André Carbonneau*[2], *Herman Kamper*[1]

[1]E&E Engineering, Stellenbosch University, South Africa
[2]Ubisoft La Forge, Montreal, Canada

benjamin.l.van.niekerk@gmail.com, julian.zaidi@ubisoft.com, marc-andre.carbonneau@ubisoft.com, kamperh@gmail.com

Abstract

Discovering a lexicon from unlabeled audio is a longstanding challenge for zero-resource speech processing. One approach is to search for frequently occurring patterns in speech. We revisit this idea by proposing DUSTED: Discrete Unit Spoken-TErm Discovery[1]. Leveraging self-supervised models, we encode input audio into sequences of discrete units. Inspired by alignment algorithms from bioinformatics, we find repeated speech patterns by searching for similar sub-sequences of units. Since discretization discards speaker information, DUSTED finds better matches across speakers, improving the coverage and consistency of the discovered patterns. We demonstrate these improvements on the ZeroSpeech Challenge, achieving state-of-the-art results on the spoken-term discovery track. Finally, we analyze the duration distribution of the patterns, showing that our method finds longer word- or phrase-like terms.

Index Terms: spoken-term discovery, pattern matching, zero resource speech processing

1. Introduction

Spoken-term discovery relies on finding matching speech segments representing words or short phrases. The main difficulty is the enormous variability of spoken language. Words are seldom said the same way due to differences in speaking rate, intonation, pronunciation, context, and speaker identity. Another challenge is segmentation – delineating continuous speech into separate words [1]. Unlike the spaces between written words, speech rarely has easily identifiable boundaries. Despite this complexity, children learn to recognize a few words even before their first birthday [2]. Their vocabulary expands rapidly over the next years, growing to about a thousand words by age three [3, p.282].

Recently, the ZeroSpeech Challenge [4] has driven progress on this problem. The goal is to build systems that generalize across languages without requiring textual annotations or labels. Such systems could facilitate low-resource speech technology [5] or serve as cognitive models of language acquisition [6].

Although various methods have been developed to tackle spoken-term discovery [7–9], many submissions to the ZeroSpeech Challenge rely on dynamic time-warping (DTW) [10–13]. These methods trace back to the Segmental-DTW algorithm [10]. The basic idea is to search for similar speech patterns by aligning pairs of utterances using DTW. Intuitively, shared words between the utterances will sound similar, leading to low-distortion regions in the alignment.

However, DTW methods have several drawbacks. With increasing dataset sizes, aligning every pair of utterances has become infeasible. Instead, these methods use heuristics such as pre-filtering and windowing to manage computational costs [11]. Additionally, since alignments depend on spectral features that contain speaker-specific information, matching words across speakers is less likely than within the same speaker. As a result, words infrequently repeated by the same speaker may not be discovered. Finally, setting hyperparameters that perform consistently across different datasets and languages is challenging [13].

To tackle these limitations, we revisit the idea of pattern matching using discrete speech representations. Leveraging recent self-supervised speech models, we encode input audio into sequences of discrete units that capture phonetic content while discarding speaker-specific details [14, 15]. Next, we find matching segments across pairs of utterances by searching for common sub-sequences of units.

We evaluate our method on the spoken-term discovery track of the ZeroSpeech Challenge. We investigate language pretraining and clustering strategies and analyze the speaker invariance and duration distribution of the discovered patterns.

Our main contributions are:

1. We propose DUSTED: **D**iscrete **U**nit **S**poken-**TE**rm **D**iscovery. Our approach significantly increases the number of discovered pairs, particularly cross-speaker matches (Section 4.3).
2. We investigate the trade-off between the quality and quantity of discovered pairs (Section 4.1). By adjusting a similarity threshold, we can prioritize coverage or phonemic similarity. We show similar threshold settings perform consistently across languages and give state-of-the-art results.
3. We investigate native language effects due to the discrete units by comparing pattern matching on one language using discrete units learned on another (Section 4.2). In contrast to previous work [16], we find that the units are not language-independent and that language-specific models improve performance.

2. Method

Our method consists of two parts. First, the content encoder extracts discrete representations of speech. Next, the pattern matcher builds a set of candidate words by searching for similar speech segments across pairs of utterances.

2.1. Content Encoder

The content encoder extracts discrete speech representations that capture phonetic content while discarding speaker-specific details [17]. This is crucial for matching patterns across different speakers. For the same reason, discrete units have also been successful in tasks like voice conversion [15, 18] and speech-to-speech translation [19]. Here, we discretize input speech by clustering features from an intermediate layer of HuBERT [14]. Formally, given a sequence of features $\langle \mathbf{z}_1, \dots, \mathbf{z}_T \rangle$, we replace

[1]Code available at `https://github.com/bshall/dusted`

 10.21437/Interspeech.2024-1051

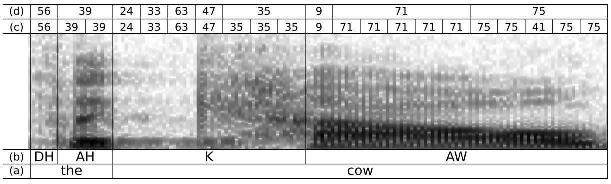

Figure 1: ***Content Encoder****. An example segmentation of the phrase 'the cow'. a) Ground truth word boundaries. b) Aligned phonemic transcription. c) Discrete speech units extracted by clustering features from an intermediate layer of HuBERT. d) A grouping of the units into longer segments using the method described in Section 2.1.*

each frame with the index of the nearest cluster centroid. Figure 1(c) illustrates this step.

Often, neighboring frames belong to the same cluster. Nevertheless, some acoustically similar frames are mapped to different units. For instance, the end of the vowel /AW/ in Figure 1(b) is split between clusters 75 and 41. So, to group the frames into longer segments we apply the dynamic programming method from [20]. Specifically, we partition the frames into a sequence of contiguous segments $\langle g_1, \ldots, g_N \rangle$, where each segment $g_n = (a_n, b_n, i_n)$ is defined by a start step a_n, an end step b_n, and a representative cluster index i_n. We determine the segmentation by minimizing the total distance between the features and their assigned cluster centroids:

$$\mathcal{E}(\mathbf{z}_{1:T}, g_{1:N}) = \sum_{g_n \in g_{1:N}} \sum_{t=a_n}^{b_n} \|\mathbf{z}_t - \mathbf{e}_{i_n}\| - \gamma(b_n - a_n),$$

where $\mathbf{e}_i$ is the ith centroid. The last term in the summation encourages longer segments, with γ controlling its weight. Without the regularizer, the optimal segmentation places each frame in its own segment. Figure 1(d) shows an example segmentation where the units in row (c) are combined into longer groups. Ultimately, the content encoder represents an utterance as the sequence of cluster indexes given by the segmentation.

2.2. Pattern Matcher

After translating input speech into sequences of discrete units, the pattern matcher searches for similar fragments across pairs of utterances. The intuition is that matching fragments should represent common words or phrases. Specifically, we find the most similar sub-sequence given discrete representations for two utterances $\langle x_1, \ldots, x_N \rangle$ and $\langle y_1, \ldots, y_M \rangle$. We identify similar sub-sequences using the Smith-Waterman algorithm [21], originally designed for nucleic acid or protein sequence alignment. The algorithm accounts for variability in the sequences by allowing insertions, deletions, and substitutions. Figure 2 shows an example alignment using the algorithm. The orange path represents the most similar sub-sequence between the two utterances, which includes a gap and a substitution (in bold):

```
 Top: 42 80 70 49 78 81 56 95 23 93 1
Left: 42 80 70 49 78 -- 56 95 40 93 1
```

We score the sub-sequences based on similarity (how many units they have in common). We apply the pattern matcher to each pair of utterances in a dataset and record matches scoring above a similarity threshold τ. The threshold controls the trade-off between the quantity and quality of the discovered patterns (see the experiments in Section 4.1).

Next, we describe the four steps of the algorithm:

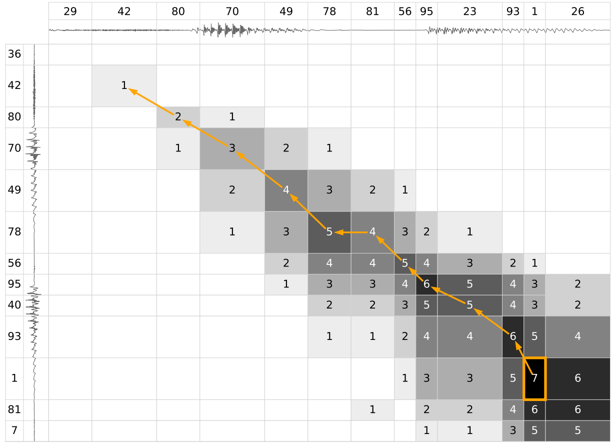

Figure 2: ***Pattern Matcher****. The scoring matrix and alignment path for two instances of the word 'something'. The first row and column show discrete representations of the words (obtained from the content encoder, Figure 1). The highest score (highlighted in orange) represents the similarity of the aligned sub-sequences. The orange arrows visualize the traceback path.*

1. **Determine a scoring scheme.** First, we define a substitution function $\text{sim}(x, y)$ that returns a score for matching cluster centroids x and y. This score is positive if x and y are similar and negative if dissimilar. In this paper, we only consider exact matches:

$$\text{sim}(x, y) = \begin{cases} +1, & \text{if } x = y, \\ -1, & \text{if } x \neq y. \end{cases}$$

However, this formulation allows more flexible measures of similarity. For example, we could specify different scores for matching units representing sonorants, obstruents, or silences [22]. We also define a gap penalty W for including an insertion or deletion in the alignment. We set $W = 1$ for all experiments.

2. **Fill the scoring matrix.** Next, we set up a scoring matrix H of size $(N + 1) \times (M + 1)$. A cell $H_{i,j}$ represents the maximum similarity between two sub-sequences ending in x_i and y_j. We initialize the first row and column of H to zeros and iteratively fill H from left to right and top to bottom using the recurrence:

$$H_{i,j} = \max \begin{cases} H_{i-1,j-1} + \text{sim}(x_i, y_j), \\ H_{i-1,j} - W, \\ H_{i,j-1} - W, \\ 0 \end{cases}$$

The first line is the score for aligning x_i with y_j. The second and third lines are the scores for an insertion or deletion. Finally, the zero represents no similarity between the sub-sequences. Figure 2 shows the scoring matrix for the sequences along the top and left.

3. **Traceback to find the most similar sub-sequence.** The traceback starts at the highest-scoring element in H above the similarity threshold τ (highlighted in orange in Figure 2). If two or more elements are tied for the maximum, we select the one with the lowest index sum $i + j$ (towards the top-left corner in Figure 2). From this starting point, we recursively trace back by visiting the neighboring element that leads to the

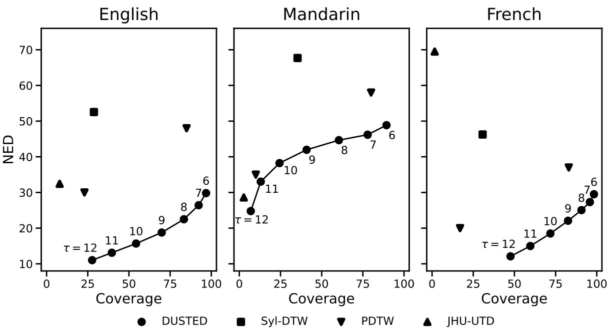

Figure 3: ***Comparison with the Baselines**. Coverage versus NED for DUSTED (at different similarity thresholds τ) and three state-of-the-art systems built on dynamic time-warping.*

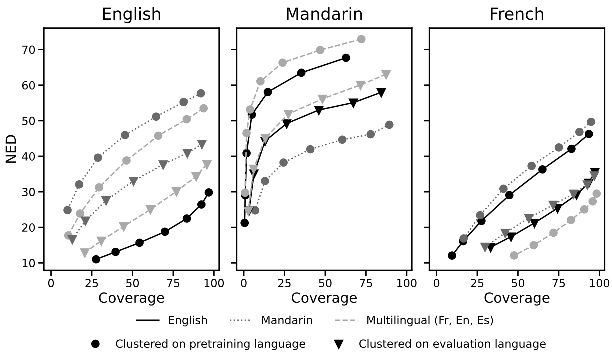

Figure 4: ***Effect of Language Pre-training and Clustering**. We compare pattern matching on one language using discrete units learned on another. We report coverage versus NED at different similarity thresholds τ.*

maximum score. We stop the procedure when we encounter a zero. The orange arrows illustrate the traceback path in Figure 2.

4. **Iteratively identify all matching sub-sequences.** The scoring matrix may include multiple matches above the similarity threshold τ. We use the rescoring method from [23] to find the next highest-scoring alignment. To avoid overlapping matches, we set all cells along the previous traceback path to zeros and recompute the scoring matrix. Only part of H needs to be updated since only elements below and to the right of the path are affected. We repeat the traceback and rescoring steps (3 and 4) until no matches above the threshold remain.

3. Experimental Setup

We conduct four experiments evaluating DUSTED. First we compare DUSTED to three state-of-the-art systems built on dynamic time-warping: PDTW [13], Syl-DTW [24], and JHU-UTD [11]. Next, we examine the effect of pre-training language. Specifically, we investigate pattern matching on one language using discrete units learned on another. Then, we analyze the impact of discrete units on cross-speaker matches, confirming DUSTED's improvements stem from speaker invariance. Finally, we examine the duration distribution of the discovered patterns, showing that DUSTED finds longer word- or phrase-like terms.

We evaluate DUSTED on the spoken-term discovery track of the ZeroSpeech Challenge [4]. The challenge covers five languages: English, Mandarin, French, German, and Wolof. We limit our experiments to languages with publicly available HuBERT models (English[2], Mandarin[3], and French[4]). We were unable to find a language-specific model for French. So we use a multilingual model trained on French, English, and Spanish [19].

3.1. Implementation Details

We split the evaluation datasets into short audio clips using the voice activity detection markers provided by the challenge. Then, we extract features for each language using the corresponding HuBERT model. Following previous work [15], we take activations from the 7th transformer layer because they perform well for phone discrimination [14, 25]. We cluster the features using k-means with 100 clusters. Next, we apply the method described in Section 2.1 to segment the features, setting the duration weight $\gamma = 0.2$, following [22]. Finally, we find matching patterns between each pair of utterances in the language dataset using the method in Section 2.2. We filter out short matches that are unlikely to contain complete words. Specifically, we ignore matches below $200\,\text{ms}$ given that the average duration of a consonant-vowel syllable is $156\,\text{ms}$ at a normal speaking rate [26]. We report results at thresholds τ from 6 to 12.

3.2. Evaluation Metrics

We evaluate spoken-term discovery using the matching metrics provided by the ZeroSpeech Challenge. The first metric is coverage: the proportion of the corpus covered by the patterns (higher is better). The second is normalized edit distance (NED), which measures the phonemic similarity between discovered pairs. Computing NED requires phonemic transcriptions for each discovered pattern, which are extracted from forced alignments. A phoneme is included in a transcription if it overlaps with the pattern by more than $30\,\text{ms}$ or 50% of its duration. Then, the normalized Levenshtein distance between the transcriptions of each discovered pair is computed. Finally, NED reports the average distance over all pairs (lower is better).

4. Results

4.1. Comparison to State-of-the-Art Systems

This section compares DUSTED to existing methods based on dynamic time-warping. Typically, spoken-term discovery systems balance NED against coverage. DUSTED controls this trade-off through the similarity threshold τ. Increasing the threshold encourages more similar but longer matches. However, being more restrictive leads to fewer pairs and lower coverage. We further investigate the effect of the threshold on the duration of the discovered patterns in Section 4.4.

Figure 3 reports the performance of DUSTED at different thresholds alongside three state-of-the-art methods. The ideal system would be in the bottom-right corner of the figure (low NED and high coverage). Regardless of the threshold, DUSTED outperforms other methods operating at similar trade-off points. At comparable coverage, we improve NED over PDTW by 13.5 points on average. Additionally, the threshold's effect is relatively consistent across languages, allowing us to reliably prioritize NED or coverage.

A drawback of DUSTED is the amount of data required to

[2] https://huggingface.co/facebook/hubert-base-ls960
[3] https://huggingface.co/TencentGameMate/chinese-hubert-base
[4] https://huggingface.co/voidful/mhubert-base

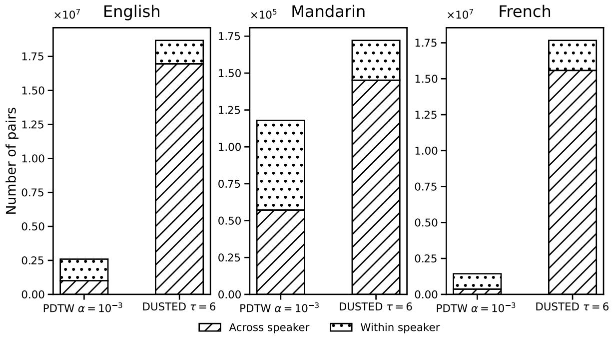

Figure 5: *A comparison of the number of within- and across-speaker pairs discovered by PDTW [13] and DUSTED.*

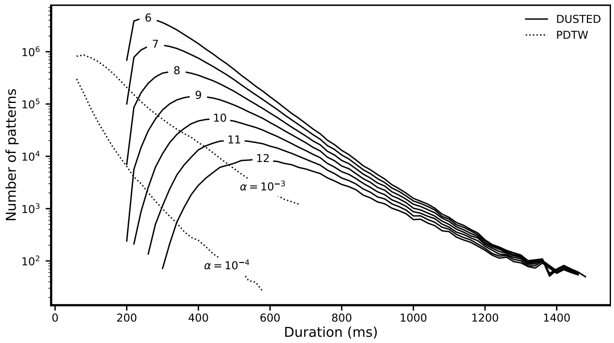

Figure 6: *The duration distribution of discovered patterns on English for DUSTED (at different similarity thresholds τ) and PDTW [13] (at different significance thresholds α).*

train the content encoder on new languages [14]. While DTW-based methods rely on spectral features, we use self-supervised models trained on large datasets to learn meaningful representations of speech. One method to address this limitation is transfer learning from a model trained on well-resourced languages. We analyze the effect of language transfer in the next section.

4.2. Effect of Language Pre-training and Clustering

We investigate the native language effect of the content encoder in two scenarios:

1. The training language of the content encoder and k-means clustering differs from the evaluation language. For example, we could use an English HuBERT clustered on English data to encode French speech.
2. We cluster on the evaluation language, but the content encoder is trained on a different language. Here, we could use an English HuBERT but cluster on French data.

Scenario 1 represents the largest mismatch between the content encoder and evaluation language. We test all combinations of training and evaluation languages using the hyperparameters described in Section 3.1.

Figure 4 presents our findings. Overall, matching the training and evaluation languages leads to the best performance. Compared to the mismatched content encoders (other lines with circle markers), these results suggest that HuBERT learns language-specific representations, contradicting previous work [16]. However, clustering on the evaluation language (triangle markers) improves performance despite a mismatched content encoder, showing we can mitigate some language mismatch.

The results for the multilingual content encoder are particularly interesting. Although the pre-training languages include English, the multilingual encoder performs worse than the English-specific model. Additionally, when evaluating on Mandarin, multilingual training gives no advantage over training solely on English. Previous studies argue that multilingual training results in transferable representations [27]; however, we do not see this advantage in our experiments. To summarize, matching the content encoder's pre-training language to the evaluation language is best for spoken-term discovery.

4.3. Analysis of Speaker Invariance

This section analyzes the speaker composition of the discovered patterns. Figure 5 compares the number of pairs found by DUSTED and PDTW, divided into across-speaker and within-speaker matches. Our method discovers more patterns in each language, in line with the coverage results from Section 4.1. Importantly, DUSTED predominantly finds pairs from different speakers: over 80% of the matches found by DUSTED are cross-speaker, compared to less than 50% for PDTW. These findings demonstrate that the discrete speech units effectively discard speaker information. As a result, the pattern matcher can discover terms based on content rather than speaker-specific details. This is essential for spoken-term discovery since many words and phrases will not be repeated by the same speaker. In contrast, PDTW relies on spectral features that contain speaker information, limiting the number of cross-speaker matches.

4.4. Duration of Discovered Fragments

Finally, we examine the durations of the discovered patterns. Ideally, the patterns should capture words or short phrases spanning hundreds of milliseconds to over a second. Figure 6 shows duration distributions for DUSTED and PDTW at different thresholds. As discussed in section 4.1, raising the threshold τ encourages longer matches with higher similarity, reflected in a larger average duration of the patterns. However, more restrictive thresholds reduce the number of matches, lowering overall coverage.

Figure 6 shows that DUSTED discovers longer fragments compared to PDTW. To reduce computational costs, PDTW imposes a maximum window size on alignments, limiting the length of the discovered patterns to $700\,\mathrm{ms}$. Consequently, PDTW discovers shorter fragments concentrated around 100 ms—roughly the duration of a syllable [26]. On the other hand, DUSTED does not set an upper limit on the matches. As a result, we discover patterns ranging from 200 to $1400\,\mathrm{ms}$, which represent longer word- or phrase-like units.

5. Conclusion

This paper introduced DUSTED, a new spoken-term discovery method combining pattern matching with discrete speech units. Since discrete units discard speaker information, DUSTED finds matches based on phonetic content rather than speaker details. This results in significantly more discovered patterns, particularly across speakers. Our experiments showed that DUSTED outperforms existing systems on the ZeroSpeech Challenge, improving the quality and quantity of the discovered terms. We also evaluated the impact of pre-training language on the discrete speech units. Our findings indicate that self-supervised representations are not language-independent, and that language-specific models can improve spoken-term discovery.

6. Acknowledgements

We thank Okko Räsänen for kindly providing us with the official submission files for PDTW.

7. References

[1] J. Saffran, R. Aslin, and E. Newport, "Statistical Learning by 8-Month-Old Infants," *Science*, 1996.

[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 States of America*, 2012.

[3] K. G. Shipley and J. G. McAfee, *Assessment in Speech-Language Pathology: A Resource Manual*, 6th ed. Plural Publishing, 2021.

[4] 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.

[5] H. Kamper, A. Jansen, and S. Goldwater, "A Segmental Framework for Fully-Unsupervised Large-Vocabulary Speech Recognition," *Computer Speech and Language*, 2017.

[6] E. Dupoux, "Cognitive Science in the Era of Artificial Intelligence: A Roadmap for Reverse-Engineering the Infant Language-Learner," *Cognition*, 2018.

[7] O. Räsänen, G. Doyle, and M. C. Frank, "Unsupervised Word Discovery from Speech Using Automatic Segmentation into Syllable-like Units," in *Interspeech*, 2015.

[8] H. Kamper, K. Livescu, and S. Goldwater, "An embedded Segmental k-means Model for Unsupervised Segmentation and Clustering of Speech," in *IEEE Automatic Speech Recognition and Understanding Workshop*, 2017.

[9] S. Bhati, J. Villalba, P. Żelasko, and N. Dehak, "Self-Expressing Autoencoders for Unsupervised Spoken Term Discovery," in *Interspeech*, 2020.

[10] A. S. Park and J. R. Glass, "Unsupervised Pattern Discovery in Speech," *IEEE Transactions on Audio, Speech, and Language Processing*, 2008.

[11] A. Jansen and B. Van Durme, "Efficient Spoken Term Discovery Using Randomized Algorithms," in *IEEE Workshop on Automatic Speech Recognition and Understanding*, 2011.

[12] V. Lyzinski, G. Sell, and A. Jansen, "An Evaluation of Graph Clustering Methods for Unsupervised Term Discovery," in *Interspeech*, 2015.

[13] O. Räsänen and M. A. C. Blandón, "Unsupervised Discovery of Recurring Speech Patterns Using Probabilistic Adaptive Metrics," in *Interspeech*, 2020.

[14] W.-N. Hsu, B. Bolte, Y.-H. 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*, 2021.

[15] B. van Niekerk, M.-A. Carbonneau, J. Zaïdi, M. Baas, H. Seuté, and H. Kamper, "A Comparison of Discrete and Soft Speech Units for Improved Voice Conversion," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2022.

[16] J. Millet and E. Dunbar, "Do Self-Supervised Speech Models Develop Human-like Perception Biases?" in *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, 2022.

[17] B. van Niekerk, L. Nortje, M. Baas, and H. Kamper, "Analyzing Speaker Information in Self-Supervised Models to Improve Zero-Resource Speech Processing," in *Interspeech*, 2021.

[18] A. Polyak, Y. Adi, J. Copet, E. Kharitonov, K. Lakhotia, W.-N. Hsu, A. Mohamed, and E. Dupoux, "Speech Resynthesis from Discrete Disentangled Self-Supervised Representations," in *Interspeech*, 2021.

[19] A. Lee, H. Gong, P.-A. Duquenne, H. Schwenk, P.-J. Chen, C. Wang, S. Popuri, Y. Adi, J. Pino, J. Gu, and W.-N. Hsu, "Textless Speech-to-Speech Translation on Real Data," in *North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2022.

[20] H. Kamper and B. van Niekerk, "Towards Unsupervised Phone and Word Segmentation Using Self-Supervised Vector-Quantized Neural Networks," in *Interspeech*, 2021.

[21] T. Smith and M. Waterman, "Identification of Common Molecular Subsequences," *Journal of Molecular Biology*, 1981.

[22] B. van Niekerk, M.-A. Carbonneau, and H. Kamper, "Rhythm Modeling for Voice Conversion," *IEEE Signal Processing Letters*, 2023.

[23] M. Waterman and M. Eggert, "A New Algorithm for Best Subsequence Alignments with Application to tRNA-rRNA Comparisons," *Journal of Molecular Biology*, 1987.

[24] O. Räsänen and S. Seshadri, "ZS2017 AaltoLAG," 2017. [Online]. Available: https://doi.org/10.5281/zenodo.810808

[25] N. Tu Anh, d. S. Maureen, R. Patricia, R. Morgane, K. Evgeny, B. Alexei, D. Ewan, and D. Emmanuel, "The Zero Resource Speech Benchmark 2021: Metrics and Baselines for Unsupervised Spoken Language Modeling," in *NeurIPS Self-Supervised Learning for Speech and Audio Processing Workshop*, 2020.

[26] H. Kuwabara, "Acoustic Properties of Phonemes in Continuous Speech for Different Speaking Rate," in *International Conference on Spoken Language Processing*, 1996.

[27] A. Conneau, A. Baevski, R. Collobert, A. Mohamed, and M. Auli, "Unsupervised Cross-Lingual Representation Learning for Speech Recognition," in *Interspeech*, 2021.