INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Memory Network-Based End-To-End Neural ES-KMeans for Improved Word Segmentation

Yu Iwamoto, Takahiro Shinozaki

Tokyo Institute of Technology
www.ts.ip.titech.ac.jp

Abstract

Unsupervised word learning from unlabeled speech is a fundamental problem in zero-resource speech processing, which enables dialogue agents to learn new words directly from spoken utterances. The embedded segmental K-means (ES-KMeans) is a representative unsupervised word segmentation method. However, it has a heterogeneous structure consisting of word boundary search based on Dynamic Programming, segment embedding, and K-Means clustering, which prevents unified optimization. This paper proposes an end-to-end neural network version of the ES-KMeans model. We apply the memory network to hold a dictionary of word embeddings and realize the word boundary search and the clustering respectively as forward and backward propagations. Moreover, we replace the fixed embedding function of the original method with a learnable neural network. Experimental results using the ZeroSpeech Challenge 2020 package show the proposed approach provides superior performance to the state-of-the-art methods.
Index Terms: unsupervised word segmentation, spoken term discovery, memory network

1. Introduction

In contrast to humans who learn spoken dialogue earlier than reading characters [1, 2], existing spoken dialogue agents (or systems) first need a large amount of labeled speech data to learn languages by supervised learning. While supervised learning is efficient given the labeled data, it requires an enormous development cost that is only affordable to limited applications in major languages. Moreover, the agents can not automatically adapt to language variations and changes through conversations lacking the ability to learn directly from spoken utterances, which is essential for flexible open dialogues.

Several techniques have been proposed that contribute to relaxing or removing the limitations. Unsupervised speech recognition trains speech recognition system using unpaired speech and text given a pronunciation dictionary or a text phonemization tool [3, 4]. Some end-to-end spoken language understanding approaches for classification-type tasks shrink the traditional pipeline of a speech recognition system and a natural language understanding module (NLU) by directly processing the audio input removing the requirement of the trained speech recognizer [5, 6].

An approach to creating a general spoken dialogue agent that can directly learn from raw audio is first to learn pseudo units of phones or words and then make a language model on them [7, 8]. Learning phone-like units has the advantage that learning from raw audio is relatively easy since we do not need to consider a longer temporal structure. On the other hand, learning word-like units is ideal for modeling sentences since words are meaning units corresponding to visual objects, etc. The larger granularity is also advantageous in generating sentences.

Unlike character sequence segmentation, matching acoustic segments is not trivial. Therefore, some unsupervised segmentation methods only find unit boundaries in the input continuous audio signals and do not make a dictionary [9]. Making a dictionary requires labeling of variable length segments in addition to finding acoustic boundaries [10, 11, 12].

The embedded segmental K-means (ES-KMeans) [10] is a representative unsupervised word segmentation method that can make the dictionary. It alternatively performs word boundary search based on Dynamic Programming given a dictionary of word embedding vectors and K-Means clustering to update the dictionary. It uses a heuristically designed embedding function to map variable length segments to fixed dimensional embedding vectors, where a segment is down-sampled and flattened to a vector. Instead of considering all possible word boundaries at the acoustic frame level, it takes candidates of word boundaries as the input to assist the boundary search. While the original method uses syllable segmentation based on amplitude envelope [13] to obtain boundary candidates, extension methods [14, 15] use other units such as phonemes based on the similarity of acoustic feature vector frames and the units found by wav2vec 2.0 [16]. While ES-KMeans is based on an approximation of the Bayesian approach and has a clear objective function, the algorithm is heterogeneous, consisting of the boundary search, segment embedding, and K-Means clustering, which prevents unified optimization. With the fixed embedding function, the learned dictionary would remain sub-optimal. It also makes the implementation complex when we want to use it as a sub-component of other systems, such as a dialogue agent.

The implementation problem is partially addressed by vector-quantized neural networks [17, 18] and neural network based Dynamic programming [19, 20] by realizing the dictionary and Dynamic programming as neural networks. By integrating all the components by a neural network, affinity for neural network based other systems is improved.

The vector-quantized neural network in [17] is an application of the memory networks [21] and holds the vector quantization codebook as a learnable memory module. It performs the vector quantization using the input continuous acoustic vector as the query and the memory elements as the keys and values, and obtains the quantized representation of the input. During the training using the reconstruction error or the prediction loss [22] as the learning objectives, the codebook is optimized as a part of the whole network. Using the trained codebook as a phone dictionary, assuming that the change of the acoustic vector is small inside a phone, [18] searches phone boundaries using Dynamic Programming as a generalization of the

10.21437/Interspeech.2023-1251

approach of [23]. The search process is the same as that of the ES-KMeans method. The limitations of these methods are that they are only useful for short units like phones, as reported in the paper. Because they do not model the internal temporal structure of the units, they can not handle longer units like words. Also, unlike the ES-KMeans, the phone dictionary is fixed after the initial learning and not updated using the segmentation results.

This paper proposes an end-to-end neural network version of the ES-KMeans model for unsupervised word learning extending the vector-quantized neural networks by introducing a neural network-based learnable segment embedding function that models the temporal structure of words. We integrate all the algorithm components in an end-to-end neural network, including the Dynamic Programming based boundary search [1]. The proposed method has a unified optimization structure useful for holistic optimization and a more straightforward implementation that is easy to extend. We evaluate our method using the ZeroSpeech Challenge 2020's evaluation package and show our proposed method outperforms the state-of-the-art methods.

2. Embedded segmental K-means

Given a feature vector sequence of a continuous utterance $X = \langle x_1, x_2, \cdots, x_T \rangle$, ES-KMeans algorithm aims to break it down into a sequence of meaningful segments or words $W = \langle w_1, w_2, \cdots, w_L \rangle$ and cluster them. It iteratively optimizes segmentation $\mathcal{Q}$ and clustering $\mathcal{Z}$, where T is the number of frames, and L is the number of words in the utterance. The overall optimization objective is:

$$H(\mathcal{Q}, \mathcal{Z}) = \sum_{c=1}^{K} \sum_{w \in W_c^{\mathcal{Q}}} len\,(w)\, \|f_e\,(w) - \mu_c^{\mathcal{Z}}\|^2, \quad (1)$$

where K is the number of clusters, $W_c^{\mathcal{Q}}$ is a set of segments by $\mathcal{Q}$ assigned to cluster c by $\mathcal{Z}$, $len\,(w)$ is segment length of w, and $\mu_c^{\mathcal{Z}}$ is c-th cluster centroid of $\mathcal{Z}$. An embedding function $f_e(\cdot)$ maps a variable-length segment to a fixed dimensional vector, which is implemented by down-sampling. ES-KMeans algorithm iteratively optimizes segmentation $\mathcal{Q}$ and clustering $\mathcal{Z}$ while fixing one of them.

Given a fixed clustering $\mathcal{Z}$, the objective (1) becomes:

$$H(\mathcal{Q}) = \sum_{w \in W^{\mathcal{Q}}} d(w) = \sum_{w \in W^{\mathcal{Q}}} len(w) \|f_e\,(w) - \mu_c^{\mathcal{Z}}\,(w)\|^2, \quad (2)$$

where $W^{\mathcal{Q}}$ is a set of all segments by $\mathcal{Q}$, $d(w)$ is a word segment score and $\mu_c^{\mathcal{Z}}\,(w)$ is the cluster centroid of $\mathcal{Z}$ closest to $f_e\,(w)$. Let $\gamma\,[t] = H\,(\mathcal{Q}_t^*)$ be the optimal segmentation score for a partial utterance up to t-th frame $X_t = \langle x_1, x_2, \cdots, x_t \rangle$. Using Dynamic Programming, the optimal segmentation score of the entire utterance $H\,(\mathcal{Q}_T^*) = \min_{\mathcal{Q}} H(\mathcal{Q}) = \gamma\,[T]$ is efficiently obtained by recursively applying Equation (3) from $t = 1$ to T.

$$\gamma\,[t] = \min_{j=1}^{t} \{d\,(w_{t-j+1:t}) + \gamma\,[t-j]\}\,, \quad (3)$$

where $w_{t-j+1:t}$ is a segment starting at frame $t-j+1$ and ending at t, and $\gamma\,[0] = 0$. By backtracking the recursion process, the optimal L and segmentation $\mathcal{Q}^* = \mathcal{Q}_T^*$ are obtained.

[1]The code is available at `https://github.com/tttslab/nn-eskmeans`.

Algorithm 1 NN-ES-KMeans algorithm

1: **Input:** feature vectors $X = \langle x_1, x_2, \cdots, x_T \rangle$, initial encoder weights θ_{enc}
2: Randomly initialize segmentation $\mathcal{Q}$
3: Initialize codebook V by k-means++
4: **while** not converge **do**
5: Compute $\min_{\mathcal{Q}} H(\mathcal{Q})$ (calculate $\mathcal{L}_s$ by forwardpropagation)
6: Update V and θ_{enc} by backpropagation
7: **end while**

Under the fixed segmentation $\mathcal{Q}$, the clustering $\mathcal{Z}$ is performed by the standard K-means method, which determines the segment assignments and updates the cluster mean. The process of the segmentation and the clustering is repeated until it converges.

3. Proposed method

3.1. Basic algorithm

We reformulate the ES-KMeans and implement it as an end-to-end neural network. In the original ES-Kmeans, the entity of the clustering $\mathcal{Z}$ is a set of cluster means $\{\mu_c^{\mathcal{Z}}\}_{c=1}^K$. In our proposed method, we replace it with an array of vectors $V = \{v_c\}_{c=1}^K$ used as keys and values of a memory network, where the vector v_c is an embedded representation of a word. We refer to V as a dictionary of word embeddings. The segmentation objective is the same as Equation (2) except for the term $\mu_c^{\mathcal{Z}}\,(w)$, which is replaced by $v_c(w)$ as shown in Equation (4).

$$H(\mathcal{Q}) = \sum_{w \in W^{\mathcal{Q}}} len\,(w)\, \|f_e\,(w) - v_c(w)\|^2, \quad (4)$$

where $v_c(w)$ is the dictionary entry closest to the embedding representation of the word w. We select the dictionary entry $v_c(w)$ by using $f_e\,(w)$ as a query in the memory network and making a hard decision in the query-key matching. We compute a one-hot weight vector for the hard decision by finding a minimum Euclidian distance between the keys and the query. We obtain the optimal segmentation score $H(\mathcal{Q}^*)$ by the recursive calculation of Equation (3) implemented as a forward propagation of a neural network.

While the original ES-KMeans uses heuristically designed fixed embedding function $f_e(\cdot)$, we use Long Short Term Memory (LSTM)[27] encoder as a learnable embedding function as shown in Equation (5) for improved word learning.

$$f_e(w_{t:t+I}) = y_{t+I}, \quad (5)$$

where $(w_{t:t+I})$ is an input segment starting at time t, I is the length of the segment, and y_{t+I} is the final output of the LSTM. We pretrain the LSTM encoder by forming a sequential autoencoder connecting a LSTM decoder to the output of the LSTM encoder and train it using a set of randomly split segments so as to minimize the reconstruction loss.

We form a single end-to-end neural network connecting the memory network, the Dynamic Programming network, and the LSTM-based embedding function, and train it using the optimal segmentation score $H(\mathcal{Q}^*)$ as the loss function $\mathcal{L}_s$. The word embedding vectors in V are updated by the backpropagation as a part of the end-to-end network, corresponding to the cluster centroid update in the original ES-Kmeans method. The forward and backward propagations are alternatively iterated for multiple epochs, just as in regular neural network learning.

Table 1: *Results for the three languages. P = Precision, R = recall, F = F-score.*

	Boundary			Token			Type		
English	P	R	F	P	R	F	P	R	F
SylSeg+ES-KMeans[10]	**51.0**	55.4	52.7	13.0	14.1	13.5	**8.3**	16.7	**11.1**
PhnSeg+ES-KMeans[14]	26.4	41.0	32.2	5.0	8.0	6.2	4.5	9.4	6.1
wav2vec2.0+ES-KMeans(iterative)[15]	29.4	67.6	41	6.4	13.3	8.6	4.2	11.7	6.1
Self-Expressing-Autoencoder [24]	32.5	78.9	46.1	5.8	16.8	8.6	2.1	24.1	3.9
seq2seq-RNN[25]	37.7	63.9	47.4	6.1	11.1	7.9	2.5	**27.1**	4.5
PDTW[26]	29.4	85.2	43.7	2.2	**27.8**	4.1	3.5	14.2	5.6
NN-ES-KMeans	46	**70.1**	**55.6**	**14.1**	18.2	**15.9**	4.5	16.7	7.1
NN-ES-KMeans+sampling	45.9	64.6	53.7	13.4	16	14.6	4.5	20.5	7.3
French	Boundary			Token			Type		
SylSeg+ES-KMeans[10]	37.8	41.6	39.6	3.5	3.9	3.7	3.1	6.3	4.2
PhnSeg+ES-KMeans[14]	25.4	38.4	30.6	4.8	7.6	5.9	4.2	7.9	5.5
wav2vec2.0+ES-KMeans(iterative)[15]	30.1	61.2	40.4	6.1	11	7.8	3.7	9.6	5.3
Self-Expressing-Autoencoder [24]	34.0	83.9	48.4	5.5	17.2	8.3	2.6	16.2	4.5
seq2seq-RNN[25]	39.2	72.4	50.9	6.3	12.6	8.4	3.1	**22.5**	5.5
PDTW[26]	31.6	**86.4**	46.3	2.8	**30.1**	5.1	4.6	9.1	6.1
NN-ES-KMeans	39.6	74.3	51.6	9.1	15.2	11.4	4.5	6.3	5.2
NN-ES-KMeans+sampling	**42.4**	72.5	**53.5**	**10.5**	15.8	**12.6**	**5.2**	8.8	**6.5**
Mandarin	Boundary			Token			Type		
SylSeg+ES-KMeans[10]	36.5	47.1	41.1	2.5	3.4	2.9	2.5	4.1	3.1
PhnSeg+ES-KMeans[14]	43.8	66.8	52.9	6.9	11.5	8.7	7.7	10.4	8.8
wav2vec2.0+ES-KMeans(iterative)[15]	43.8	71.4	54.3	13.7	22.7	17.1	15.3	23.3	18.5
Self-Expressing-Autoencoder [24]	36.5	91.9	52.2	7.9	25.4	12.1	6.9	**29.1**	11.1
seq2seq-RNN[25]	42.5	80.7	55.7	9.3	18.1	12.3	8.4	28.9	13.0
PDTW[26]	34.2	87.4	49.2	2.4	23.9	4.4	10.3	11.2	10.7
NN-ES-KMeans	53.2	**93.6**	**67.9**	**19.1**	**30.5**	**23.5**	13.9	14	13.9
NN-ES-KMeans+sampling	**55**	69.5	61.4	18.8	18.5	18.7	**16.3**	28.4	**20.7**

Table 2: *Ablarion study on Mandarin*

	min	features	$f_e(\cdot)$	pretrain	commit. loss	enc. undate	epoch	Boundary-F	Token-F	Type-F
(1)	det.	CPC	LSTM	✓	✓	✓	250	67.9	23.5	13.9
(2)	rand	CPC	LSTM	✓	✓	✓	300	61.4	18.7	20.7
(3)	det.	CPC	LSTM	✓	✓	✓	0	58.1	15.4	11.5
(4)	det.	CPC	LSTM	✓	✓	✓	100	63.9	20.1	14
(5)	det.	MFCC	LSTM	✓	✓	✓	40	59.3	17.5	9.9
(6)	det.	MFCC	downsample	-	-	-	50	52.9	7.9	11.6
(7)	det.	CPC	LSTM	-	✓	✓	120	56.8	13.8	13.8
(8)	det.	CPC	LSTM	✓	-	✓	100	64.2	19.1	13
(9)	det.	CPC	LSTM	✓	✓	-	400	61.4	17.8	11.7
(10)	rand	CPC	LSTM	✓	✓	-	150	61.4	17.9	13.5

Algorithm 1 summarizes the whole process of the proposed method. As the initialization of $\mathcal{Q}$, we make a random segmentation. Based on the random segmentation, we initialize the dictionary V using the initialization algorithm of K-means++[28]. Then we alternatively repeat the calculation of $H(\mathcal{Q}^*) = \min_{\mathcal{Q}} H(\mathcal{Q})$ by the forward propagation based Dynamic Programming and the update of V and θ_{enc} by the back-propagation until it converges.

3.2. Commitment loss

We train the network so that the embedded representation $f_e(w)$ and the dictionary entry $v_c(w)$ become closer. However, if the embedded representations approach to the dictionary entries faster than the dictionary entries appropriately distribute, the learning process might prematurely converge. To control the balance of the learning speed between the embedding function and the dictionary elements, we add a commitment loss to Equation (4) as in [29] as shown in Equation (6).

$$\mathcal{L}_s = H(\mathcal{Q}) = \sum_{w \in W^{\mathcal{Q}^*}} \{len(w) \, \| sg[f_e(w)] - v_c(w) \|^2 + \alpha \cdot len(w) \, \| f_e(w) - sg[v_c(w)] \|^2\}, \quad (6)$$

where $sg[\cdot]$ is an operator that indicates skipping the weight update. The first and the second terms contribute to the updates of the dictionary entries and the encoder, respectively, and α controls their balance.

3.3. Sampling

In the basic algorithm, we make deterministic decisions with the $min(\cdot)$ function for the dictionary element selection and the Dynamic Programming. However, we might be caught in a local minimum since we likely repeat the same selection. As an extension, we replace the deterministic $min(\cdot)$ operation with random samplings using the Gumbel-Max trick[30].

The new equation for the Dynamic Programming in Equation (3) becomes Equation (7).

$$\gamma[t] = d(w_{t-s+1:t}) + \gamma[t-s], \quad s = \arg\max_j [-\{d(w_{t-j+1:t}) + \gamma[t-j]\}/\tau_{\mathcal{Q}} + g_j], \tag{7}$$

where $\tau_{\mathcal{Q}}$ is temperature, $g_j = -log(-log(u))$, and u is a sample form a uniform distribution $Uniform(0, 1)$. Similarly, the new equation for v_c in (8) becomes as in Equation (8).

$$v_c(w) = \arg\max_{v_j} [-\{\|f_e(w) - v_j\|^2\}/\tau_{\mathcal{Z}} + g_j], \tag{8}$$

where $\tau_{\mathcal{Z}}$ is temperature.

4. Experimental Setup

We evaluated our proposed method with the Zero Resource Speech Challenge 2020's 2017 Track 2 task using the official package released in their website [31][2] . The package provides data sets of three languages of English, French, and Chinese. Their speech data amount to 45, 24, and 2.5 hours. In the evaluation, **Boundary**, **Token**, and **Type** accuracies are measured [32]. The boundary accuracy evaluates the start and end boundaries individually, the Token accuracy sees the pair of the start and end boundaries of each segment, and Type accuracy measures the accuracy of the obtained vocabulary.

As the acoustic features, we used CPC features [33] using the CPC_audio toolkit[3]. The pre-training of the LSTM encoder and the training of the neural ES-KMeans took around 100 and 13 hours using NVIDIA RTX3090 GPU. We chose the hyper-parameters based on a preliminary experiment using the Mandarin data set. With the minimum segment length, 150 ms gave the best results among 50, 100, 150, and 200 ms. Similarly, the dictionary size $K = 500$ gave the best among 250, 500, 1000, 2500, and 5000. The temperatures $\tau_{\mathcal{Q}} = 2$ and $\tau_{\mathcal{Z}} = 0.1$ were the best among 0.1, 0.5, 1.0, 2.0, and 3.0. We used the same hyper-parameter settings for all three languages.

5. Results

Table 1 show the results for the three languages. Among the baselines, SylSeg+ES-KMeans, PhnSeg+ES-KMeans, and wav2vec2.0+ES-KMeans(iterative) are families of the ES-KMeans method that use syllable segmentation (the original method), phonetic segmentation, and wav2vec2.0 to obtain the word boundary candidates. Others (i.e., Self-Expressing-Autoencoder[24], seq2seqRNN[25], and PDTW[26]) are non ES-KMeans methods. NN-ES-KMeans is the proposed method using the deterministic $\min$ function and NN-ES-KMeans+sampling is the one using the sampling version.

While the scores of all the methods tend to vary depending on the languages, the proposed methods outperformed all the existing methods in Boundary and Token F-scores. For the Type F-score, the original ES-KMeans provided the best score for English but NN-ES-KMeans+sampling gave the best for French and Mandarin. When we compare NN-ES-KMeans+sampling with NN-ES-KMeans, The Boundary and Token F-scores decreased slightly, but Type-F improved for all the languages. The sampling contributed to finding more word types in the dictionary but also increased broken segment instances due to the introduced randomness. The averaged Boundary, Token, and Type F-scores over the three languages by the original ES-KMeans method were 44.5, 6.7, and 6.1, and those for the NN-ES-KMeans and NN-ES-KMeans+sampling were 58.5, 16.9, and 8.8, and 56.2, 15.3, and 11.5, respectively, which were significantly superior to the original version.

[2] https://zerospeech.com/challenge_archive/2020/tasks/

[3] https://github.com/tuanh208/CPC_audio

6. Ablation study

We performed an ablation study using the Chinese data set for analysis. Table 2 shows the results. The epoch column shows the best number of training epochs after the initialization except for lines (3) and (4), where we specified 0 and 100 epochs to see the performance change. The first line (1) is NN-ES-KMeans, and the second line (2) is NN-ES-KMeans+sampling, and these are the same as the ones shown in Table 1. By comparing (3), (4), and (1), we observe the F-scores gradually improve with the increase of the epochs. Comparing (1) and (5) shows that CPC features are more effective than MFCC. When using the MFCC features, the LSTM encoder (5) gave slightly worse Type F-score but largely better Boundary and Token F-scores than the downsampling (6). Removing pretraining (7) and commitment loss (8) reduced the Boundary and Token F-scores, but the influence on Type-F was minor. There were no significant differences between the deterministic decision (9) and the random sampling (10) in the Boundary and the Token-Fscore when the encoder was not updated, but the random sampling improved the Type F-score.

7. Limitations of the work

The proposed method has a learnable embedding function. While it is useful to improve performance, it increases the computation cost since we can not pre-compute the embedding representations of possible segments as the embedding function changes with the learning. The training objective score largely improves when we initialize the dictionary entries by oracle word segments even after significant training epochs, which implies the optimization process has room for improvement.

8. Conclusions

We have proposed an end-to-end neural network version of the ES-KMeans method. It integrates all the components of the original ES-KMeans in a single neural network and replaces the fixed segment embedding function with a learnable neural network module. It realizes the segmentation and the clustering of the original ES-KMeans by the forward and backward propagation, which makes it easy to add extensions or use the method with other neural network-based systems. Experimental results show that the proposed methods outperform existing state-of-the-art approaches with Boundary, Token, and Type F-scores in most conditions. When we introduced the sampling using the Gumbel-Max trick for the dictionary element selection and the Dynamic Programming calculation, Boundary and Token F-scores slightly decreased due to the introduced randomness, but an improved Type F-score was obtained. Future work includes extending word dictionary learning by combining it with language model learning.

9. Acknowledgements

This study was supported by JSPS KAKENHI Grand Number JP22K12069.

10. References

[1] *The Cambridge Handbook of Child Language*, 2nd ed., ser. Cambridge Handbooks in Language and Linguistics. Cambridge University Press, 2015.

[2] 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, 2018.

[3] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," in *Advances in Neural Information Processing Systems*, 2021.

[4] A. H. Liu, W.-N. Hsu, M. Auli, and A. Baevski, "Towards end-to-end unsupervised speech recognition," in *IEEE Spoken Language Technology Workshop (SLT), Doha, Qatar, 2023, pp. 221-228*, 2022.

[5] Y.-P. Chen, R. Price, and S. Bangalore, "Spoken language understanding without speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Calgary, AB, Canada*, 2018, pp. 6189–6193.

[6] S. Dmitriy, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5754–5758.

[7] K. Lakhotia, E. Kharitonov, Y. A. Wei-Ning Hsu, A. Polyak, B. Bolte, T.-A. Nguyen, J. Copet, A. Baevski, A. Mohamed, and et al., "On generative spoken language modeling from raw audio," in *Transactions of the Association for Computational Linguistics*, vol. 9, 2021, pp. 1336–1354.

[8] R. Komatsu, S. Gao, W. Hou, M. Zhang, T. Tanaka, K. Toyoda, Y. Kimura, K. Hino, Y. Iwamoto, K. Mori, T. Okamoto, and T. Shinozaki, "Automatic spoken language acquisition based on observation and dialogue," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, pp. 1480–1492, 2022.

[9] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velázquez, and N. Dehak, "Segmental Contrastive Predictive Coding for Unsupervised Word Segmentation," in *Proc. Interspeech 2021*, 2021, pp. 366–370.

[10] H. Kamper, K. Livescu, and S. Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *in Proc. IEEE Automat. Speech Recognit. Understanding Workshop*, 2017, p. 719–726.

[11] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, "Self-expressing autoencoders for unsupervised spoken term discovery," in *Proc. Interspeech 2021*, 2020, pp. 4876–4880.

[12] O. Räsänen and M. A. C. Blandón, "Unsupervised discovery of recurring speech patterns using probabilistic adaptive metrics," in *Proc. Interspeech 2021*, 2020.

[13] O. Räsänen, G. Doyle, and M. C. Frank, "Unsupervised word discovery from speech using automatic segmentation into syllable-like units," in *Proc. Interspeech 2015*, 2015.

[14] S. Bhati, H. Kamper, and K. S. R. Murty, "Phoneme based embedded segmental k-means for unsupervised term discovery," *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5169–5173, 2018.

[15] Y. Iwamoto and T. Shinozaki, "Unsupervised spoken term discovery using wav2vec 2.0," *2021 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, pp. 1082–1086, 2021.

[16] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *NeurIPS*, 2020.

[17] B. van Niekerk, L. Nortje, and H. Kamper, "Vector-quantized neural networks for acoustic unit discovery in the zerospeech 2020 challenge," in *Interspeech*, 2020.

[18] H. Kamper and B. van Niekerk, "Towards unsupervised phone and word segmentation using self-supervised vector-quantized neural networks," in *Proc. Interspeech 2021*, 2021, pp. 1539–1543.

[19] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," ser. ICML '06, 2006, p. 369–376. [Online]. Available: https://doi.org/10.1145/1143844.1143891

[20] T. Tanaka and T. Shinozaki, "Efficient free keyword detection based on cnn and end-to-end continuous dp-matching," in *Proc. ASRU, Paper ID SDR.6*, 2019.

[21] J. Weston, S. Chopra, and A. Bordes, "Memory networks," in *International Conference on Learning Representations*.

[22] A. van den Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," in *CoRR abs/1807.03748*, 2018.

[23] J. Chorowski, N. Chen, R. Marxer, H. J. G. A. Dolfing, A. Łańcucki, G. Sanchez, T. Alumäe, and A. Laurent, "Unsupervised Neural Segmentation and Clustering for Unit Discovery in Sequential Data," in *NeurIPS 2019 PGR workshop*, 2019.

[24] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, "Self-expressing autoencoders for unsupervised spoken term discovery," in *Proc. Interspeech 2020*, 2020.

[25] "Zerospeech 2020 leaderboards," https://zerospeech.com/challenge_archive/2020/results/.

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

[27] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," in *https://doi.org/10.48550/arXiv.1402.1128*, 2014.

[28] D. Arthur and S. Vassilvitskii, "k-means++: the advantages of careful seeding," in *Society for Industrial and Applied Mathematics Philadelphia, PA, USA*, 2007, pp. 1027–1035.

[29] A. van den Oord, O. Vinyals, and K. Kavukcuoglu, "Neural discrete representation learning," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, 2017, p. 6309–6318.

[30] E. J. Gumbel, "Statistical theory of extreme values and some practical applications: a series of lectures." in *Number 33. US Govt. Print. Office*, 1954.

[31] 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 *Proc. Interspeech 2020*, 2020, pp. 4831–4835.

[32] M. Versteegh, R. Thiolliere, T. Schatz, X. N. Cao, X. Anguera, A. Jansen, and E. Dupoux1, "The zero resource speech challenge 2015," in *Proc. Interspeech 2015*, 2015, pp. 3169–3173.

[33] M. Rivière, A. Joulin, P.-E. Mazar'e, and E. Dupoux, "Unsupervised pretraining transfers well across languages," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7414–7418, 2020.