Unsupervised Spoken Term Discovery Using wav2vec 2.0

Yu Iwamoto and Takahiro Shinozaki
Tokyo Institute of Technology
https://www.ts.ip.titech.ac.jp

***Abstract*—Unsupervised spoken term discovery is the task of finding recurring word-like patterns from raw audio without any manual transcription. Several approaches have been investigated, but the matching between automatically found fragments and actual words is still shallow. Recently, a self-supervised learning method wav2vec 2.0 has been proposed, and it is demonstrating outstanding performance in pre-training acoustic models for speech recognition. During the training, wav2vec 2.0 applies quantization to a latent representation of the input acoustic features. As a by-product, a discrete code sequence is obtained. In this work, we propose to use the code sequence for unsupervised term discovery. The temporal resolution of the code sequence is fine-grained, and it is closer to a phone sequence rather than a word sequence. To obtain larger units, we apply the ES-KMeans method to the code and feature sequences obtained by wav2vec 2.0. In addition, we iteratively optimize wav2vec 2.0 and ES-KMeans for further improvement. Experimental results using the Zero Resource Speech Challenge 2020's data show the proposed method outperforms existing methods on average while the results vary on languages.**

I. Introduction

Recent speech recognition systems have made remarkable progress [1], but these powerful systems require huge amounts of speech data and transcribed text labels for training. However, most languages other than the world's major languages have few transcribed data. Moreover, some languages do not have a writing system. Therefore, these speakers can not use speech recognition systems. Approaches to solving this problem are implementing a speech recognition system using a limited amount of text labels if the text is available and from a longer-term perspective to realize a spoken language acquisition system that automatically leans and understands spoken languages [2], [3]. The Zero Resource Speech Challenge [4] facilitates fundamental researches that contributes to develop the foundation for the goal.

The Zero Resource Speech Challenge has set up several tasks. One of the essential tasks is unsupervised spoken term discovery. The purpose of the task is to discover repeated word-like patterns without using any text labels. The systems should take raw speech as input and output boundary and class labels of speech fragments.

One of the powerful existing methods is Embedded Segmental K-Means (ES-KMeans) [5]. ES-KMeans optimizes word segmentation and clusters jointly. As the initialization, the algorithm takes a set of candidate word boundaries. It maps the candidate word fragments to small fixed dimensional vectors by using a word embedding method. Then it alternatively repeats clustering the embedding vectors as word clusters and re-selecting the word boundaries. It efficiently performs the re-selection of the word boundaries by using Dynamic Programming with an objective function defined by the word clusters.

There are several existing segmentation methods that provide the initial word boundaries. Among them, syllable segmentation [6] was the first method that was used in the original ES-KMeans research. It segments sound waveform based on amplitude envelope. However, depending on language, word boundaries are not clear in the envelope and it overlooks many of them. Since ES-KMeans only considers candidate word boundaries given at the beginning, it does not work well for such languages. Other options for the initial segmentation includes phonetic segmentation [7] and self-expressing autoencoders [8]. However, their performance varies on languages and there is no single best method.

In this paper, we propose to apply wav2vec 2.0 [9] to provide the initial segmentation. The wav2vec 2.0 method has been proposed as a self-supervised learning framework to learn acoustic representations from raw audio data. It has been very powerful to develop speech recognition systems using a small amount of labeled data by using a large amount of raw speech data set or from the large amount of unpaired speech and text data [10]. The wav2vec 2.0 model is composed of a multi-layer convolutional feature encoder that takes a raw audio input and outputs its latent representations, a quantizer for the latent representation, and a Transformer [11]. For the self-supervised representation learning, it is trained with a contrastive task, where it is required to identify the true quantized latent speech representation for a masked time step among a set of distractors. As the by-product to perform the contrastive task, it generates a discrete code sequence using the quantizer. The quantizer works as a rich unsupervised segmentation method and produces a pseudo phone sequence.

We investigate two ways of the combination of wave2vec 2.0 and ES-KMeans. The first one is to simply connect the initial pseudo phone segmentation produced by wave2vec 2.0 as the input to the ES-KMeans method. The second one is to alternatively perform wave2vec 2.0 and ES-KMeans. We feedback the pseudo word segmentation result obtained by ES-KMeans to wave2vec 2.0 as an auxiliary input, and iteratively refine the segmentations.

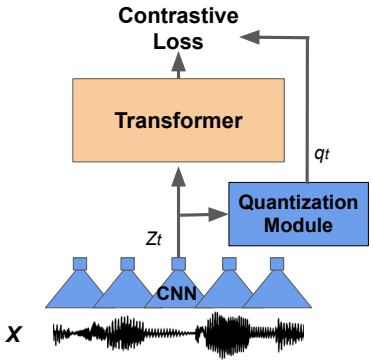

Fig. 1: Model structure of wav2vec 2.0.

II. Model Structure of wav2vec 2.0

Fig. 1 shows the model Structure of wav2vec 2.0. The feature encoder of wav2vec 2.0 is a multi-layer CNN that takes raw audio signal $\boldsymbol{X}$ as its input and outputs a sequence of latent speech representation vectors $\boldsymbol{z}_1, ..., \boldsymbol{z}_T$, where T is the frame length of the output. For the quantization, it uses product quantization with G codebooks. The output of feature encoder z is converted to logits $\boldsymbol{l} \in \mathbb{R}^G \times V$, where V is the number of quantization classes. To choose discrete codes in a differentiable manner, Gumbel softmax [12] is used as shown in Equation (1).

$$p_{g,v} = \frac{exp(l_{g,v} + n_v)/\tau}{\sum_{k=1}^{V} exp(l_{g,k} + n_k)/\tau)}, \tag{1}$$

where τ is a non-negative temperature, $n = -log(-log(u))$, u is a random sample uniformly drawn from $(0, 1)$, and $p_{g,v}$ is the probability that $l_{g,v}$ belong to v-th class. The quantized class i is found by $i = \arg\max_j p_{g,j}$. Let $I = i_1, ..., i_T$ be the code sequence and $\boldsymbol{Q} = \boldsymbol{q}_1, ..., \boldsymbol{q}_T$ be the code vector sequence corresponding to $\boldsymbol{z}_1, ..., \boldsymbol{z}_T$.

The code sequence I may be directly regarded as a sequence of pseudo words by marking the consecutive frames having the same code as a word. However, the time resolution of I is too fine-grained for it.

III. Embedded Segmental K-Means

ES-KMenas divides the sequences into word like segments and clusters these segments. Let $\boldsymbol{Y} = \boldsymbol{y}_1, ..., \boldsymbol{y}_T$ is the feature vector sequences. An acoustic word embedding method f_e maps variable length segments $\boldsymbol{y}_{t_1:t_2}$ that segmented by initial segmentation method to fixed dimensional vector $\boldsymbol{w}_i = f_e(\boldsymbol{y}_{t_1:t_2})$. Let $B = \{b_i\}_{i=1}^{M}$ is the boundaries, where M is the number of segments and b_i indicates the boundary for segment i. These boundaries are selected from word boundary candidates generated by the initial segmentation method. The embedded segments under the current segmentation are represented by $W(B)$.

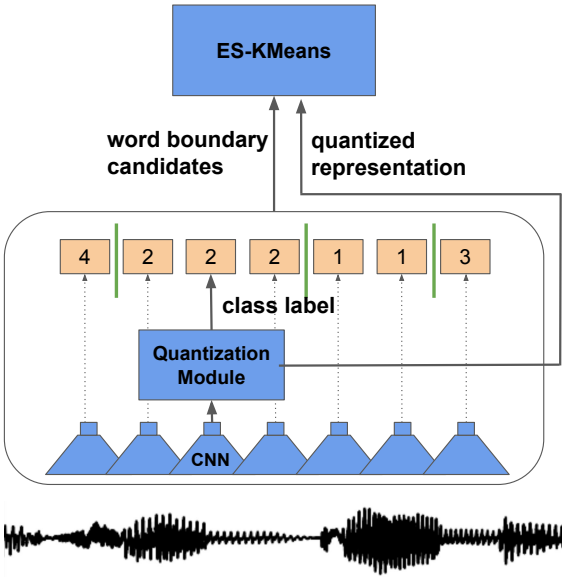

Fig. 2: Structure of the proposed sequential combination of wav2vec 2.0 and ES-KMeans.

ES-KMeans objective is an extension of standard K-Means objective, and it is defined as follows

$$\min_{B,a} \sum_{c=1}^{K} \sum_{\boldsymbol{w} \in W_c \cap W(B)} len(\boldsymbol{w}) ||\boldsymbol{w} - \boldsymbol{\mu}_c||^2 \tag{2}$$

where a indicates which cluster $\boldsymbol{w}$ is assigned to, K is the number of clusters, $W_c \cap W(B)$ are segments assigned to cluster c under segmentation B, $len(\boldsymbol{w})$ is the number of frames of segment w and μ_c is cluster mean.

ES-KMeans algorithm is a 2-step iteration. The algorithm classifies like Standard K-Means with fixed boundaries and determines the assignments a. Then dynamic programming algorithm updates boundaries B based on the assignments a and cluster centers $\boldsymbol{\mu}_c$.

IV. Proposed Method

A. Sequential Combination

Fig. 2 shows our strategy of combining wav2vec 2.0 and ES-KMeans sequentially. We use the code sequence generated by the quantization module of wav2vec 2.0 as the initial word boundary candidates for ES-KMeans. We use the code vector sequence $\boldsymbol{Q}$ as the feature vector sequence $\boldsymbol{Y}$. We regard the changes of the codes in the code sequence I as indicators of the possible word boundaries. ES-KMeans selects the boundaries B from them.

Compared to directly use wav2vec 2.0 as the word segmentation method, we can expect longer and better word segmentation. For simplicity, we set the number of the quantization Groups G to 1, where the product quantization reduces to vector quantization.

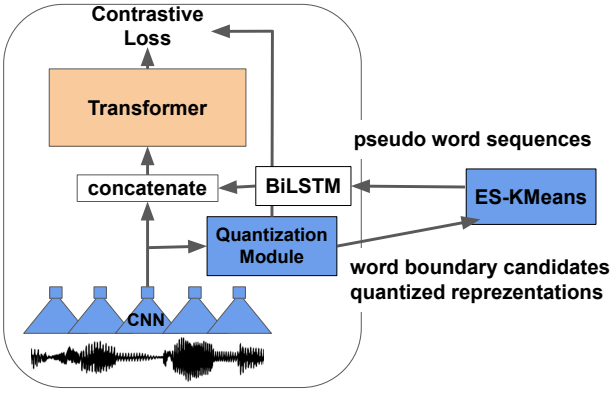

Fig. 3: Structure of the proposed iterative combination of wav2vec 2.0 and ES-KMeans.

B. Iterative Combination

There is no explicit constraint for wav2vec 2.0 that it generates the same pseudo phone sequence for multiple instances of the same word in the waveform input. In fact, it often produces different pseudo phone sequences for different appearances of the same word. We expect that wave2vec2.0's phone segmentation performance is improved by using the pseudo word segmentation information obtained by ES-Kmeans as an additional input.

Fig. 3 shows how to iteratively refining the ES-KMeans word segmentation and wav2vec 2.0 phone segmentation. In the first iteration, we sequentially run wav2vec 2.0 and ES-KMeans in order as in the sequential method without modifying the input to wav2vec 2.0. In the following iterations, we extend the input of wav2vec 2.0 by using the pseudo word segmentation information produced by ES-KMeans in the previous iteration.

As the word segment information, we use a sequence of centroid vectors of pseudo word clusters made by ES-KMeans regarding the centroid vectors as an embedding expression of the pseudo word segments. We up-sample the embedding vector sequence to the frequency of the latent speech representation vectors $\boldsymbol{z}_t$ by repeating the same vector in a pseudo word segment. After applying a Bi-LSTM network, we input it to the transformer module together with the latent speech representation vectors.

V. Experimental setup

We used the data set provided by the Zero Resource Speech Challenge 2020. The data set contains five languages, but an evaluation script is only publicly available for three languages among them. We used the three languages, which are English, French, and Mandarin. They have 45, 24 and 2.5 hours of data, respectively. The evaluation script by the Zero Resource Speech Challenge 2020 provides multiple performance measures. Boundary is a measure that evaluates word boundaries at acoustic frame level aligned to nearest reference phone boundaries. It is explained in the Zero Resource Speech Challenge 2020's web page that the Boundary measure is provided for completeness and for system diagnostic[1]. The primary measures are Token and Type. Token evaluates the quality of the found word segments in frame level, and Type evaluates the quality of the discovered vocabulary.

We trained wav2vec 2.0 using the fairseq toolkit [15]. Fairseq provides two configurations of models having different number of parameters, and we used the BASE model. The CNN output size is 768, and we set the quantization number V to 128. In the proposed method, the Bi-LSTM module to use the pseudo word information from ES-Kmeans is a one-layer network whose hidden size is 256. When combined with the CNN output, the input size of the transformer module is $768 + 256 = 1024$. We separately trained a model for each language.

VI. Results

Table I shows summary results by averaging the results over the three languages. We denote the sequential wav2vec 2.0 and ES-KMeans combination as "wav2vec 2.0+ES-KMeans" and the iterative combination as "wav2vec 2.0+ES-KMeans(iterative)", where the number of the iterations was 3.

For comparison, we include scores of existing methods reported at the Zero Resource Speech Challenge in the table. These include; "Self clustering Autoencoder for unsupervised features learning" (self clustering autoencoder) [8], "seq2seq RNN for features learning with UAD pairs" (seq2seq RNN), and "Probabilistic DTW" (PDTW) [14]. PDTW has two variations registered with the same name, and we describe them as PDTW(1) and PDTW(2). Further, SylSeg+ES-KMeans and phnSeg+ES-KMeans indicate the results of ES-KMeans using syllable segmentation and phonetic segmentation, respectively.

While wav2vec 2.0+ES-KMeans and wav2vec 2.0+ES-KMeans(iterative) are inferior to several conventional methods (i.e., self clustering autoencoder, seq2seq RNN, and PDTW(1)) in terms of the Boundary F-score, they outperform conventional methods in Token and Type F-measures. We conjecture that the reason is that wav2vec 2.0 has high sensitivity in finding phonetic changes in sound signals. It finds more possible boundaries than existing methods, resulting in higher recall and lower F-score in the Boundary measure. However, it is actually advantageous to achieve higher F-scores in more important Token and Type measures.

Compared to the conventional ES-KMeans based methods (i.e. SylSeg+ES-KMeans and PhnSeg+ES-KMeans), wav2vec 2.0+ES-KMeans is superior to these methods for all three evaluation measures. Comparing the results before and after applying ES-KMeans to wav2vec 2.0, Boundary was slightly reduced but Token and Type were improved. Especially, the improvement of Token was large. This comparison shows the usefulness of applying ES-KMeans and of using wav2vec 2.0 as an initial segmentation for running ES-KMeans.

[1] https://zerospeech.com/2020/index.html

TABLE I: Summary scores averaged over the three languages.

	mean of Boundary-F	mean of Token-F	mean of Type-F
baseline [13]	3.2	0.2	0.9
self clustering autoencoder [8]	48.9	9.7	6.5
seq2seq RNN	**51.3**	9.5	7.7
PDTW(1) [14]	46.4	4.5	7.5
PDTW(2)	25.3	2.6	2.1
SylSeg+ES-KMeans [5]	44.5	6.7	6.1
PhnSeg+ES-KMeans [7]	38.6	6.9	6.8
wav2vec 2.0	45.5	6.1	8.5
wav2vec 2.0+ES-KMeans	45.3	10.0	8.8
wav2vec 2.0+ES-KMeans(iterative)	45.2	**11.2**	**10.0**

TABLE II: Detailed results for the three languages. P = Precision, R = recall, F = fscore.

	Boundary			Token			Type				
English	P	R	F	P	R	F	P	R	F	NED	Cov
baseline	32.1	3.2	5.9	1.9	0.1	0.3	1.9	1.7	1.8	32.4	7.9
self clustering autoencoder	32.5	78.9	46.1	5.8	16.8	8.6	2.1	24.1	3.9	89.5	99.5
seq2seq RNN	37.7	63.9	47.4	6.1	11.1	7.9	2.5	**27.1**	4.5	94.0	99.2
PDTW(1)	29.4	85.2	43.7	2.2	**27.8**	4.1	3.5	14.2	5.6	48.2	85.4
PDTW(2)	27.4	28.5	28.0	2.2	3.1	2.6	5.6	1.7	2.6	**30.4**	23.1
SylSeg+ES-KMeans	**51.0**	55.4	**52.7**	**13.0**	14.1	**13.5**	**8.3**	16.7	**11.1**	72.6	100.0
PhnSeg+ES-KMeans	26.4	41.0	32.2	5.0	8.0	6.2	4.5	9.4	6.1	72.2	100.9
wav2vec 2.0	26.7	**86.7**	40.8	1.7	6.2	2.7	6.0	2.0	3.0	-	-
wav2vec 2.0+ES-KMeans	27.5	77.0	40.6	4.4	13.1	6.6	4.1	5.9	4.8	86.7	100.0
wav2vec 2.0+ES-KMeans(iterative)	29.4	67.6	41	6.4	13.3	8.6	4.2	11.7	6.1		
French	Boundary			Token			Type			NED	Cov
baseline	32.5	0.6	1.2	1.3	0.0	0.1	3.0	0.3	0.5	69.5	1.6
self clustering autoencoder	34.0	83.9	48.4	5.5	17.2	8.3	2.6	16.2	4.5	89.0	99.8
seq2seq RNN	**39.2**	72.4	**50.9**	**6.3**	12.6	**8.4**	3.1	**22.5**	5.5	93.1	99.7
PDTW(1)	31.6	**86.4**	46.3	2.8	**30.1**	5.1	4.6	9.1	**6.1**	36.7	83.5
PDTW(2)	30.3	23.4	26.4	3.9	3.6	3.8	**7.4**	0.9	1.6	**20.3**	17.5
SylSeg+ES-KMeans	37.8	41.6	39.6	3.5	3.9	3.7	3.1	6.3	4.2	72.6	100.0
PhnSeg+ES-KMeans	25.4	38.4	30.6	4.8	7.6	5.9	4.2	7.9	5.5	72.2	100.0
wav2vec 2.0	28.8	77.6	42.0	3.3	11.0	5.1	4.5	4.8	4.7	-	-
wav2vec 2.0+ES-KMeans	29.5	69.2	41.4	5.3	13.4	7.6	4.1	8.4	5.5	86.4	100.0
wav2vec 2.0+ES-KMeans(iterative)	30.1	61.2	40.4	6.1	11	7.8	3.7	9.6	5.3		
Mandarin	Boundary			Token			Type			NED	Cov
baseline	**54.3**	1.3	2.5	6.4	0.1	0.2	4.9	0.2	0.3	28.6	2.7
self clustering autoencoder	36.5	**91.9**	52.2	7.9	**25.4**	12.1	6.9	**29.1**	11.1	94.7	99.9
seq2seq RNN	42.5	80.7	**55.7**	9.3	18.1	12.3	8.4	28.9	13.0	97.3	99.8
PDTW(1)	34.2	87.4	49.2	2.4	23.9	4.4	10.3	11.2	10.7	57.6	79.7
PDTW(2)	34.7	15.5	21.4	2.5	1.1	1.5	14.5	1.2	2.2	**34.9**	10.4
SylSeg+ES-KMeans	36.5	47.1	41.1	2.5	3.4	2.9	2.5	4.1	3.1	88.1	100.0
PhnSeg+ES-KMeans	43.8	66.8	52.9	6.9	11.5	8.7	7.7	10.4	8.8	80.0	117.0
wav2vec 2.0	38.5	89.4	53.8	6.7	22.9	10.4	**22.2**	14.9	17.8	-	-
wav2vec 2.0+ES-KMeans	41.1	77.9	53.8	11.8	24.3	15.9	13.7	19.1	16.0	94.9	100.0
wav2vec 2.0+ES-KMeans(iterative)	43.8	71.4	54.3	**13.7**	22.7	**17.1**	15.3	23.3	**18.5**		

Looking at the effect of the iteration with the proposed method, the Token-F increased by 1.1 and the Type-F increased by 0.8. Fig. 4 shows the relationship between the number of iterations and the performance. The score of the first iteration is equivalent to the simple combination of wav2vec 2.0 and ES-KMeans (wav2vec 2.0+ES-KMeans in Table I). The first three iterations were the most effective.

Table II shows the detailed results for each language. In the table, P, R, and F stand for precision, recall, and F-score, respectively. We could not obtain NED and Cov for wav2vec 2.0 using the toolkit, and they are not shown in the table. The evaluation script did not terminate. As can be seen, the performance of every method broadly varies on language.

We see that wav2vec 2.0 achieves high Boundary recall for all the languages though not always the best. This confirms our assumption that wav2vec 2.0 learns fine changes in speech and can find more correct boundaries. While the F-scores by our proposed methods are inferior to that of SylSeg+ES-

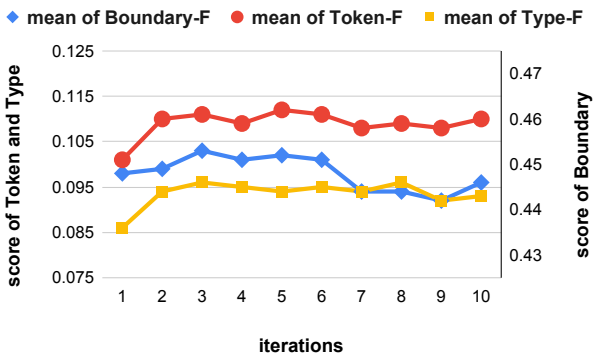

Fig. 4: Relationship between the number of iterations and the performance with wav2vec 2.0+ES-KMeans(iterative).

TABLE III: Comparison of summary scores of wav2vec 2.0+ES-KMeans(iterative) in Bi-LSTM hidden size.

hidden size	mean of Boundary-F	mean of Token-F	mean of Type-F
256	45.2	**11.2**	**10.0**
32	**45.3**	11.1	9.6

KMeans for English, they are working well for French and Mandarin. Comparing the two proposed methods, wav2vec 2.0+ES-KMeans(iterative) most always outperformed wav2vec 2.0+ES-KMeans.

Table III shows comparison of summary scores of wav2vec 2.0+ES-KMeans(iterative) in Bi-LSTM hidden size. We see that the hidden size 256 is comprehensively more effective than the hidden size 32.

VII. Conclusions

In this paper, we propose to use wav2vec 2.0 for unsupervised spoken term discovery. While the evaluation results largely vary on the languages including the existing methods, our proposed methods provided better Token and Type F-scores than other methods on average. Comparing the two proposed methods, iteratively refining the pseudo phone and word segmentation was useful to improve the performance.

VIII. Acknowledgements

This work was supported by Toray Science Foundation.

References

[1] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Trans. Audio Speech Lang. Process.*, vol. 25, no. 12, pp. 2410–2423, Dec. 2017.

[2] S. Gao, W. Hou, T. Tanaka, and T. Shinozaki, "Spoken language acquisition based on reinforcement learning and word unit segmentation," in *Proc. ICASSP*, May 2020, pp. 6149–6153.

[3] M. Zhang, T. Tanaka, W. Hou, S. Gao, and T. Shinozaki, "Sound-image grounding based focusing mechanism for efficient automatic spoken language acquisition," in *Proc. Interspeech*, Oct. 2020, pp. 4183–4187.

[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, K. Livescu, and S. Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," 2017.

[6] O. Rasanen, G. Doyle, and M. C. Frank, "Unsupervised word discovery from speech using automatic segmentation into syllable-like units," in *INTERSPEECH*, 2015, pp. 3204–3208.

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

[8] S. Bhati, J. Villalba, P. Żelasko, and N. Dehak, "Self-expressing autoencoders for unsupervised spoken term discovery," 2020.

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

[10] A. Baevski, W.-N. Hsu, A. Conneau, and M. Auli, "Unsupervised speech recognition," 2021.

[11] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in Neural Information Processing Systems*, vol. 30, pp. 5998–6008, 2017.

[12] S. G. E. Jang and B. Poole, "Categorical reparameterization with gumbel- softmax," 2017.

[13] A. Jansen and B. V. Durme, "Efficient spoken term discovery using randomized algorithms," in *in Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 401–406.

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

[15] M. Ott, S. Edunov, A. Baevski, A. Fan, S. Gross, N. Ng, D. Grangier, and M. Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *In Proc. of NAACL System Demonstrations*, 2019.