ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 979-8-3503-4485-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICASSP48485.2024.10446327

UNSUPERVISED SPEECH RECOGNITION WITH N-SKIPGRAM AND POSITIONAL UNIGRAM MATCHING

Liming Wang[⋆]*, Mark Hasegawa-Johnson*[⋆] *and Chang D. Yoo*[†]

[⋆]Department of Electrical and Computer Engineering, University of Illinois Urbana-Champaign
[†]Artificial Intelligence and Machine Learning Lab, KAIST

ABSTRACT

Training unsupervised speech recognition systems presents challenges due to GAN-associated instability, misalignment between speech and text, and significant memory demands. To tackle these challenges, we introduce a novel ASR system, ESPUM. This system harnesses the power of lower-order N-skipgrams (up to $N = 3$) combined with positional unigram statistics gathered from a small batch of samples. Evaluated on the TIMIT benchmark, our model showcases competitive performance in ASR and phoneme segmentation tasks. Access our publicly available code at https://github.com/lwang114/GraphUnsupASR.

Index Terms— speech recognition, self-supervised speech processing, acoustic unit discovery, unsupervised phoneme segmentation

1. INTRODUCTION

Learning a speech recognizer with only unpaired speech and text corpora, or unsupervised speech recognition (ASR-U)[1], is a self-supervised learning task crucial for developing speech technology for low-resource languages. Beyond converting speech to text without reliance on transcribed speech, an ASR-U system can serve as the linchpin for low-resource text-to-speech synthesis [2, 3], speech translation [4, 5] and other spoken language understanding tasks. Despite significant strides made in the domain [6, 7, 8, 9, 10], the *stability* of ASR-U systems remains a conspicuous bottleneck [11, 2]. Many leading ASR-U models, including the current state-of-the-art models, wav2vec-U and its 2.0 iteration [9, 10], rely heavily on generative adversarial networks (GAN) [12]. These GANs are notoriously difficult to train, demanding rigorous regularization and hyperparameter tuning, often displaying sensitivity to the relative weightings of regularization losses [2].

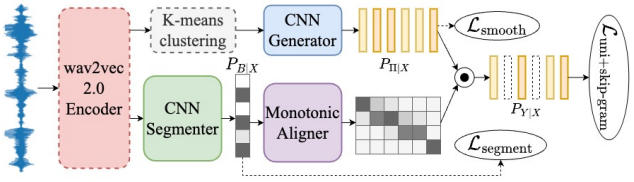
The only existing method that bypasses the need for GANs is the empirical output distribution matching (EODM) approach [8]. This approach trains an ASR system to directly match the empirical N-gram distribution of authentic phoneme sequences, eliminating the need for a discriminator. However, straightforward N-gram matching encounters difficulties: the quantity of unique N-grams quickly grows to intractable number and the accuracy of approximating the N-gram distribution diminishes as N increases. Notably, [8] revealed the necessity of 5-gram and large batch size (50,000 tokens per batch) for EODM to yield optimal ASR-U outcomes. Due to memory restrictions, they consider only the top 10,000 5-gram distributions.

An additional pivotal challenge for ASR-U lies in phoneme segmentation. While [7] utilized a fixed phoneme segmentation derived from an unsupervised model [13] during GAN training, [8] opted for a block-wise alternative minimization technique to refine segmentations. By contrast, wav2vec-U (2.0) [9, 10] performs segmentation by directly merging successive frames assigned to identical phonemes. Unfortunately, current methodologies have yet to incorporate the innovative strides made in unsupervised phoneme segmentation [14, 15, 16]. These recent techniques improve the quality of the detected phoneme boundaries by employing differentiable self-supervised learning objectives, such as (segmental) contrastive predictive coding (CPC) [14, 15] and teacher-student learning [16].

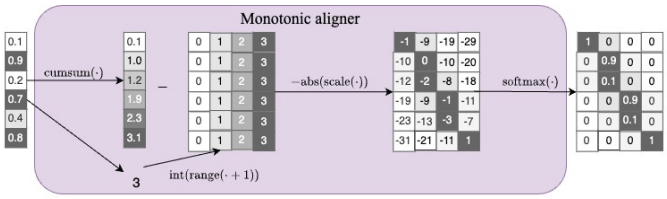
In this paper, we proposed Empirical Skipgrams and Positional Unigram Matching (ESPUM), a novel GAN-free ASR-U model based on N-skipgrams and positional unigram matching. Our model achieves competitive ASR-U performance on the standard TIMIT [17] benchmark while being more consistent in different hyperparameter settings than the GAN-based approach. Further, our model is more memory-efficient than the previous GAN-free ASR-U models [8] by requiring only lower-order N-skipgrams (up to $N = 3$) and positional unigram information. Last but not least, we design a novel differentiable phoneme segmenter end-to-end trainable with the rest of the ASR-U systems and outperform all previous methods in the unsupervised phoneme segmentation task.

2. PROBLEM FORMULATION

Suppose we have an unlabeled speech corpus consisting of speech feature sequences $X^{(i)} \in \mathcal{X}^T \sim P_X, i = 1, \cdots, n_X$ and another unpaired text-only corpus containing phoneme label sequences $Y^{(j)} \in \mathcal{Y}^L \sim P_Y, j = 1, \cdots, n_Y$. Suppose



(a) Overall architecture. Dashed components are fixed during training.



(b) Differentiable monotonic aligner

Fig. 1: Overview of ESPUM

that the speech and text corpora are *matched* in the sense that there exists a speech recognizer (ASR) $G : \mathbb{X}^T \mapsto \mathbb{Y}^L$ such that for any phoneme sequence $\forall \mathbf{y} \in \mathbb{Y}^L$,

$$P_X \circ G(\mathbf{y}) := \sum_{\mathbf{x} \in \mathbb{X}^T} P_X(\mathbf{x}) G_{\mathbf{y}}(\mathbf{x}) = P_Y(\mathbf{y}). \tag{1}$$

The goal of ASR-U is to recover such an ASR.

3. METHOD

While most existing methods employ a GAN [7, 9] for learning such an ASR, we propose a GAN-free method consisting of five main modules as shown in Fig. 1a.

3.1. Self-supervised speech generator

Our model first takes raw speech waveform as input to a pretrained, *self-supervised speech encoder* such as wav2vec 2.0 [18]. Empirically we found that the raw features lead to unstable training, and instead we use a *K-means clustering* module to discretize the speech features into one-hot vectors. A convolutional neural network (CNN) *generator* (the ASR) then converts the one-hot vectors into phoneme probabilities.

3.2. CNN segmenter

While solving Eq. (1) over all possible ASR is both statistically and computationally infeasible, in practice we can constrain the structure of the ASR significantly if we have access to *noisy, unsupervised* phoneme-level segmentations $\tilde{B}^{(i)}_{1:T}, i = 1, \cdots, n_X$, where $\tilde{B}_t = 1$ if frame t is a boundary between two phonemes and otherwise 0 if it is within a phoneme. Using these labels, we train a CNN-based *segmenter* to predict the phoneme boundaries from speech features:

$$P_\theta[B_t = 1 | X_{1:T} = \mathbf{x}] := \sigma(\texttt{CNN}_\theta(\mathbf{x})), \tag{2}$$

where $\sigma(\cdot)$ is the sigmoid function. The segmenter is then trained using a weighted binary cross-entropy (BCE) loss $\mathcal{L}_{\text{segment}}(\theta)$ on the pseudo-labels (more in Sec.4.1).

3.3. Monotonic alignment

Due to the noise in the segment labels, the raw labels from the segmenter often lead to severe misalignments between the speech features and phoneme labels. To address this issue, we propose a "soft" *monotonic alignment* $\mathbf{A}^\theta \in [0, 1]^{L \times T}$ from the segmenter by a sequence of differentiable operation shown in Fig. 1b, inspired by a similar mechanism from the Segmental CPC model [15]. Using this alignment, we then compute a sequence of segment-level features from the frame-level features as

$$\bar{X}^\theta_l = \sum_{t=1}^{T} A^\theta_{lt} X_t. \tag{3}$$

Note that this soft monotonic alignment can be trained end-to-end with the speech generator with the ASR-U criteria, allowing the model to refine the segment boundaries using information from the unpaired text data.

3.4. N-skipgram and positional unigram matching

After the monotonic alignment, the ASR now takes a much simpler form as it can predict each phoneme label independently given each segment-level speech feature. We can then learn the ASR by performing distribution matching between the segmented speech distribution $P^\theta_{\bar{X}_{1:L}}$ and the text distribution $P_{\bar{Y}_{1:L}}$. Instead of matching the full distributions or N-grams, we find it much more memory-efficient and reliable to use (bi-)*skipgrams* and more generally, N-skipgrams defined for skip sizes $\mathbf{k} := (k_1, \cdots, k_{N-1})$ as

$$P_{Y,\mathbf{k}}(y_{1:N}) := P_{Y_1, Y_{1+k_1}, \cdots, Y_{1+\sum_{l=1}^{N-1} k_l}}(y_{1:N}). \tag{4}$$

Using Eq. (1), we then learn the ASR G by minimizing the following matching loss:

$$\mathcal{L}_{\text{skipgram}}(G, \theta) := \sum_{\mathbf{k} \in \mathcal{K}} \left\| \hat{P}_{Y,\mathbf{k}} - \hat{P}^\theta_{\bar{X},\mathbf{k}} \circ G \right\|_1, \tag{5}$$

where $\mathcal{K}$ is the set of skip sizes used and $\hat{P}_{Y,\mathbf{k}}$ and $\hat{P}^\theta_{\bar{X},\mathbf{k}}$ are empirical distributions estimated from sample batches. Further, inspired by the ASR-U theory [19], we also use the *positional unigram* $P^\theta_{\bar{X}_l}$ and P_{Y_l} by another L_1 loss:

$$\mathcal{L}_{\text{unigram}}(G, \theta) := \sum_{l=1}^{L} \| \hat{P}_{Y_l} - \hat{P}_{X_l} G \|_1. \tag{6}$$

	GAN?	LM	matched		unmatched	
			val ↓	test ↓	val ↓	test ↓
EODM [8]	No	5-grams	-	42.6	-	49.1
+ HMM ST			-	36.5	-	41.6
wav2vec-U [9]	Yes	4-grams	17.0	17.8	21.3	22.3
+ CTC ST			11.3	12.0	13.8	15.0
Proposed models						
uni+bi+tri	No	4-grams	42.9	43.3	51.3	47.3
uni+bi+tri, iter 1			39.4	39.1	49.8	45.1
+ HMM ST			33.1	33.7	47.0	42.9

Table 1: PER of various ASR-U models on TIMIT.

	P ↑		R ↑		F1 ↑		R-val ↑	
	H	L	H	L	H	L	H	L
Speech-only								
Kreuk et al [14]	81.4	85.3	76.5	83.5	78.9	84.4	81.7	86.6
Kreuk et al [14]*	78.3	85.8	75.8	82.7	77.1	84.2	80.4	86.3
Bhati et al [15]	-	84.6	-	86.0	-	85.3	-	87.4
Strgar et al [16]	82.4	91.0	81.2	88.5	81.8	89.7	84.5	91.0
Strgar et al [16]*	82.6	89.6	74.8	81.6	78.5	85.4	81.0	86.4
Speech+unpaired text								
EODM [8]	-	80.9	-	84.3	-	82.6	-	84.8
wav2vec-U+HMM ST [9]	67.8	74.3	74.4	80.0	71.0	77.1	73.8	79.5
Ours (matched)	88.9	93.3	77.3	83.9	82.7	88.4	83.5	88.4
Ours (matched, iter 1)	**87.2**	**93.4**	**85.3**	**89.3**	**86.2**	**91.3**	**88.1**	**91.9**
Ours (unmatched, iter 1)	88.2	90.8	76.4	84.0	81.9	87.3	82.8	88.3

Table 2: Unsupervised phoneme segmentation results on English (TIMIT). "L" stands for the lenient metric commonly reported in the literature and "H" is the harsh metric proposed in [16]. * stands for results we obtained by running the code provided by the authors. "Ours" is a uni+bi+tri-grams ESPUM trained on TIMIT.

3.5. Smoothness and segment relabeling

Similar to previous works [8, 9]. we also apply the smoothness loss to encourage similar phoneme labels for nearby speech feature frames:

$$\mathcal{L}_{\text{smooth}}(G) := \sum_{i=1}^{n_X} \sum_{t=1}^{T} \left\| G(X_{t+1}^{(i)}) - G(X_t^{(i)}) \right\|_2^2. \quad (7)$$

The overall training objective is then

$$\mathcal{L}_{\text{unigram}} + \mathcal{L}_{\text{skipgram}} + \mathcal{L}_{\text{segment}} + \lambda_{\text{smooth}} \mathcal{L}_{\text{smooth}}. \quad (8)$$

Moreover, to further improve the segmentation quality, we replace the older, noisier labels $\tilde{B}_{1:T}$ with the predicted labels from the segmenter $\tilde{B}'_{1:T}$ after training converges using the older labels, a process called *segment relabeling*.

4. EXPERIMENTS

4.1. Experimental setup

We use the TIMIT dataset [17] with the same split as in [8, 9] for the ASR-U experiments. For the phoneme segmentation

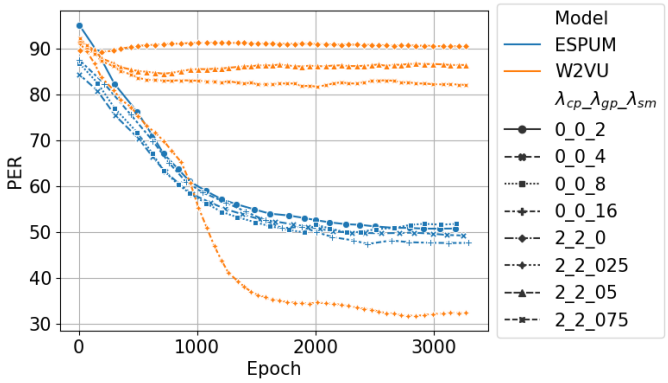


Fig. 2: Convergence behavior of ESPUM (no segment relabeling) vs wav2vec-U over a range of hyperparameters defined in [9], where λ_{cp} is the code penalty and λ_{gp} is the gradient penalty (neither used in ESPUM). $\lambda_{\text{sm}} = \lambda_{\text{smooth}}$ is defined in Eq. (8).

Model	PER ↓ (val)	Boundary F1 ↑ (val, harsh)
bigrams only	71.6	87.2
uni+bi-grams	39.2	87.4
uni+bi+tri-grams	**38.4**	87.1
uni+bi+4-grams	40.0	86.5
uni+4-grams	45.0	86.6
uni+5-grams	77.9	**87.6**
uni+bi+tri+5-grams	41.8	86.0

Table 3: Effect of different positional unigram and N-skipgram combinations. All models use the segmentation from a uni+bi+tri-grams ESPUM after one segment relabeling iteration.

experiments, the full TIMIT test set is used instead to align with prior work [14]. We map the original 60 phonemes to 39 as in [9]. Phoneme error rate (PER) is used to evaluate the ASR performance, while precision, recall, F1 and R-value metrics are used for the phoneme segmentation performance. For the latter task, we use both the *lenient* scores in [8, 14, 15] as well as the *harsh* scores defined in [16] designed to avoid double counting of detected boundaries.

We use the 14-th layer of the wav2vec 2.0 [18] model pretrained on 10,000-hour LibriLight [20] as the speech input features and a K-means module with 128 clusters for quantizing the speech features. The CNN generator is a one-layer CNN with a kernel size of 4 and a stride size of 1. To obtain noisy phoneme boundary labels, we use the wav2vec 2.0 readout model [16] with the LibriLight wav2vec 2.0 encoder as the backbone. We then use the same 7-layer CNN in [16] as the CNN segmenter. During testing, we replace the soft alignment with a mean pooling within the predicted boundaries for

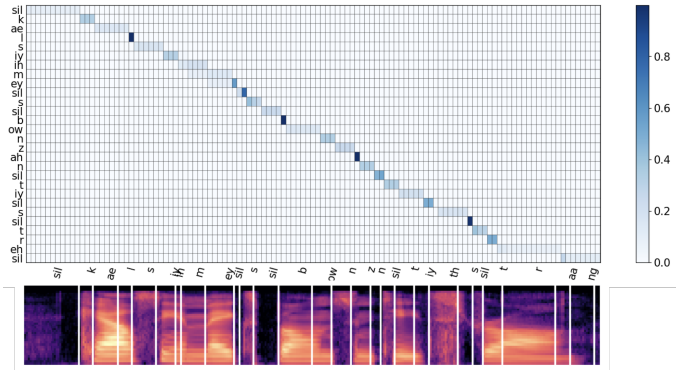


Fig. 3: An example pooling matrix generated by a uni+bi+tri-grams ESPUM after one segment relabeling iteration.

decoding. We also found that training the segmenter using BCE loss with a positive weight of 1.1 and only labels with a confidence score above 0.6 from the wav2vec 2.0 readout model achieves the best performance. We use $\lambda_{\text{smooth}} = 16$ unless specified otherwise.

For all models, we only use bi-skipgrams with skip sizes up to 6, tri-skipgrams with skip sizes up to 2 and the top 5000, or 68% of the 4-grams and 50% the 5-grams. We conduct all our experiments on a 12GB GTX 1080Ti GPU. We implement our models using fairseq and follow training settings in [9] if not specified. We train them end-to-end using Adam optimizer [21] with an initial learning rate of 0.004 and betas equal to $[0.5, 0.98]$ and a batch size of 640. We train the model further for one segment relabeling iterations for 20000 updates and observe further relabeling iteration leads to no significant improvement. We also experiment with HMM self-training (ST) techniques found previously to be effective [7, 8, 9].

4.2. Results

The overall ASR-U result is shown in Table 1. Compared with EODM [8] before self-training, our model achieves an 8% relative (42.6 → 39.1) improvement in phone error rate (PER) in both the matched and unmatched setting. Further, segment relabeling of one iteration helps to reduce PER by 8.9% relative (43.3 → 39.1). After self-training, while our model continues to outperform [8] by 7.7% relative PER (36.5 → 33.7) in the matched setting, while does not perform as well in the unmatched setting. This may be primarily due to the use of different language models used as well as discrepancy in hyperparameter settings of the self-training algorithms. Further, our model is still lagging behind the GAN-based wav2vec-U [9].

The overall unsupervised phoneme segmentation result by our model is shown in Table 2. With the help of unpaired text, our model trained in the matched setting outperforms the best previous speech-only model [16] by 5.4% relative F1 (81.8 → 86.2) and 4.2% relative R-value (84.5 → 88.1), despite starting with segmentations with lower F1 (78.5%) due to discrepancy in training setting. It is also superior to EODM, the best speech+unpaired text models by 10.5% relative F1 (82.6 → 91.3) and 8.4% relative R-value (84.8 → 91.9), though with the help of self-supervised representation pretrained on large speech corpora. We observe that segment relabeling helps to refine the segmentation by 2.3% relative F1 (82.7 → 86.2) and 2.9% relative R-value (83.5 → 88.1). In addition, ESPUM consistently achieves 4.3% relative F1 improvement (78.5 → 81.9) over the speech-only segmenter [16], the teacher of our model in the unmatched setup, demonstrating the ability of our model to leverage unpaired textual information.

4.3. Analysis

By comparing Table 1 and 2, we observe that a better recognizer (wav2vec-U) is not always a better boundary detector (ESPUM). One explanation is that better recognizers learn blurry boundaries in order to fully exploit between-phoneme context, while better boundary detector loses some recognition capacity by focusing mainly on within-phoneme context. We partially confirm our hypothesis by visualizing the pooling matrix from the ESPUM segmenter in Fig. 3, and finding that most of its weights align with the true phoneme boundaries.

Next, we compare the training stability of ESPUM and wav2vec-U by analyzing their training convergence curves. As shown in Fig. 2, wav2vec-U fails to converge when λ_{smooth} is too small or too large, while ESPUM remains consistent with a small improvement in PER as λ_{smooth} increases from 0 to 16. Such improvement suggests that local homogeneity of phoneme distributions provide additional constraints to the generator besides those provided by the N-skipgrams matching constraints.

Finally, we study the effect of different combinations of positional unigrams and N-skipgrams in Table 3. We found that lower-order skipgrams play bigger roles than higher-order skipgrams for ASR-U, evident by the fact that the uni+bi-grams model outperforms the uni+4/5-grams models. Also the information in the positional unigrams is crucial for ASR-U since the bigrams only model performs much worse than the uni+bi-grams model. While adding tri-skipgrams help improve the PER of the uni+bi-grams model, adding 4-grams or 5-grams degrades its performance.

5. CONCLUSION

In this work, we propose ESPUM, a novel GAN-free model for ASR-U that is better at phoneme segmentation and more stable to train. Future directions include a better understanding on the relation between recognition and segmentation in ASR-U as well as building end-to-end systems excelling at both tasks.

6. REFERENCES

[1] James Glass, "Towards unsupervised speech processing," in *International Conference on Information Sciences, Signal Processing and their Applications*, 2012.

[2] Junrui Ni, Liming Wang, Heting Gao, Kaizhi Qian, Yang Zhang, Shiyu Chang, and Mark Hasegawa-Johnson, "Unsupervised text-to-speech synthesis by unsupervised automatic speech recognition," in *Interspeech*, 2022.

[3] Alexander H. Liu, Cheng-I Jeff Lai, Wei-Ning Hsu, Michael Auli, Alexei Baevski, and James Glass, "Simple and effective unsupervised speech synthesis," in *Interspeech*, 2022.

[4] Changhan Wang, Hirofumi Inaguma, Peng-Jen Chen, Ilia Kulikov, Yun Tang, Wei-Ning Hsu, Michael Auli, and Juan Pino, "Simple and effective unsupervised speech translation," in *Annual Meeting of the Association for Computational Linguistics*, 2023.

[5] Yu-Kuan Fu, Liang-Hsuan Tseng, Jiatong Shi, Chen-An Li, Tsu-Yuan Hsu, Shinji Watanabe, and Hung yi Lee, "Improving cascaded unsupervised speech translation with denoising back-translation," in *Annual Meeting of the Association for Computational Linguistics*, 2023.

[6] Da-Rong Liu, Kuan-Yu Chen, Hung-Yi Lee, and Lin shan Lee, "Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings," in *Interspeech*, 2018.

[7] Kuan-Yu Chen, Che-Ping Tsai, Da-Rong Liu, Hung-Yi Lee, and Lin shan Lee, "Completely unsupervised speech recognition by a generative adversarial network harmonized with iteratively refined hidden Markov models," in *Interspeech*, 2019.

[8] Chih-Kuan Yeh, Jianshu Chen, Chengzhu Yu, and Dong Yu, "Unsupervised speech recognition via segmental empirical output distribution matching," in *International Conference on Learning Representations*, 2019.

[9] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, "Unsupervised speech recognition," in *Neural Information Processing System*, 2021.

[10] Alexander H. Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski, "Towards end-to-end unsupervised speech recognition," in *IEEE Spoken Language Technology Workshop*, 2022.

[11] Guan-Ting Lin, Chan-Jan Hsu, Da-Rong Liu, Hung-Yi Lee, and Yu Tsao, "Analyzing the robustness of unsupervised speech recognition," in *ICASSP*, 2022.

[12] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Neural Information Processing System*, 2014.

[13] Yu-Hsuan Wang, Cheng-Tao Chung, and Hung yi Lee, "Gate activation signal analysis for gated recurrent neural networks and its correlation with phoneme boundaries," in *Interspeech*, 2017.

[14] Felix Kreuk, Joseph Keshet, and Yossi Adi, "Self-supervised contrastive learning for unsupervised phoneme segmentation," in *Interspeech*, 2020.

[15] Saurabhchand Bhati, Jesus Villalba, Piotr Zelasko, Laureano Moro-Velazquez, and Najim Dehak, "Segmental contrastive predictive coding for unsupervised word segmentation," in *Interspeech*, 2021.

[16] Luke Strgar and David Harwath, "Phoneme segmentation using self-supervised speech models," in *IEEE Spoken Language Technology Workshop*, 2022.

[17] John S. Garofolo, Lori F. Lamel, William M. Fisher, Jonathon G. Fiscus, David S. Pallett, and Nancy L. Dahlgren, *The DARPA TIMIT Acoustic-Phonetic Continuous Speech Corpus CDROM*, Linguistic Data Consortium, 1993.

[18] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Neural Information Processing System*, 2020.

[19] Liming Wang, Mark Hasegawa-Johnson, and Chang Yoo, "A theory of unsupervised speech recognition," in *Annual Meeting of the Association for Computational Linguistics*, 2023.

[20] Jacob Kahn, Morgane Rivière, Weiyi Zheng, Evgeny Kharitonov, Qiantong Xu, Pierre-Emmanuel Mazaré, Julien Karadayi, Vitaliy Liptchinsky, Ronan Collobert, Christian Fuegen, Tatiana Likhomanenko, Gabriel Synnaeve, Armand Joulin, Abdelrahman Mohamed, and Emmanuel Dupoux, "Libri-light: A benchmark for asr with limited or no supervision," in *ICASSP*, 2020.

[21] Diederik P. Kingma and Jimmy Lei Ba, "Adam: A method for stochastic optimization," in *International Conference on Learning Representations*, 2015.