Segmental SpeechCLIP: Utilizing Pretrained Image-text Models for Audio-Visual Learning

Saurabhchand Bhati, Jesús Villalba[†,‡]*, Laureano Moro-Velazquez*[†]*, Najim Dehak*[†,‡]

[†]Center for Language and Speech Processing, Johns Hopkins University, USA
[‡]Human Language Technology Center of Excellence, Johns Hopkins University, USA
{sbhati1,jvillalba,laureano,ndehak3}@jhu.edu

Abstract

Visually grounded models learn from paired images and their spoken captions. Recently, there have been attempts to utilize the visually grounded models trained from images and their corresponding text captions, such as CLIP, to improve speech-based visually grounded models' performance. However, the majority of these models only utilize the pretrained image encoder. Cascaded SpeechCLIP attempted to generate localized word-level information and utilize both the pretrained image and text encoders. Despite using both, they noticed a substantial drop in retrieval performance. Here, we propose to use a hierarchical segmental audio encoder that can generate a sequence of word-like units from audio. We use the pretrained CLIP text encoder on top of these word-like units representations and show significant improvements over the cascaded variant of SpeechCLIP.

1. Introduction

Speech processing systems aided by large amounts of labeled data and computational resources achieve remarkable performance [1–3]. However, vast amounts of labeled data are not available for most languages, and transcribing a large amount of speech data is expensive. Therefore, there has been a lot of interest in developing methods to learn useful information from unlabeled data [4–10]. Recently, self-supervised learning (SSL) methods have emerged as a significant paradigm for learning representations from unlabeled audio data [1, 11, 12]. In SSL methods, the model is trained to solve a pretext task for which labels can be generated from the raw audio. Some common pretext tasks include masked language modeling [1,13], next frame prediction [11], next segment prediction [14, 15], and masked reconstruction [16, 17]. Speech systems built on top of these SSL representations require much less labeled data to match the performance of systems built without them [1]. Another direction is to use multimodal data and extract useful information to improve performance in a given modality.

Parallel text and image data have been leveraged for learning representations that help downstream performance in both modalities [18, 19]. Contrastive language image pretraining (CLIP) learns to align the parallel image and text data crawled from the internet [19]. CLIP shows remarkable performance in zero-shot setting for image classification and image/text retrieval from text/images [19]. Parallel images and spoken captions have also been leveraged to improve speech processing systems [20–25]. These systems are commonly referred to as visually grounded speech (VGS) systems. VGS systems have been shown to improve speech systems performance for speech recognition [22], word discovery [23], and speech synthesis [24]. VGS models trained with just retrieval loss can learn semantic [25] and word-level information, such as boundaries [23] from speech.

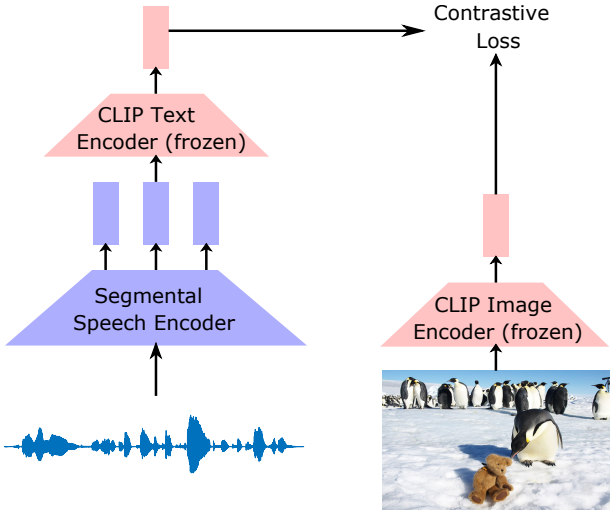

Figure 1: *Overview of the proposed Segmental SpeechCLIP.*

Recently, there have been efforts to utilize CLIP for improving the performance of VGS systems. However, most of these systems only utilize the CLIP model's image encoder. WAV2CLIP [26] and the parallel variant of speechCLIP [27] generate a single representation per utterance summarizing the information. This global representation is then used for classification and retrieval tasks. There are no constraints to localize word-level information. Guidance from models trained on text data, such as CLIP text encoder, could help extract semantic information from speech. E.g., unsupervised ASR systems use nonparallel text data and a pronunciation lexicon for speech recognition in the absence of transcribed speech data. These were some of the motivations behind the cascaded variant of speechCLIP.

The cascaded SpeechCLIP [27] model appends K learnable CLS tokens to the utterance to extract the most important keywords. Vector quantization is then used to map these keywords to CLIP's subword embeddings. The frozen text encoder is used on top to generate sentence embedding. A frozen CLIP image encoder is used to extract image representations. However, the cascaded variant has significantly lower retrieval recall scores than the parallel variant.

We propose Segmental SpeechCLIP to improve the keywords extraction from speech utterances and better utilization of the text CLIP encoder as shown in Fig. 1. We show that our Segmental SpeechCLIP significantly outperforms the cas-

caded variant of SpeechCLIP. The cascaded speechCLIP model discovers a fixed number of keywords, i.e., eight from the utterances. Our Segmental SpeechCLIP automatically deduces the number of word-like units in an utterance. There is no implicit constraint to enforce the temporal structure on the discovered keywords in the cascaded speechCLIP, whereas Segmental SpeechCLIP, by design, discovers word-like units in temporal order. Instead of quantization and mapping the keywords to subword embedding, we directly learn the subword embeddings.

Our method uses a segmental speech encoder based on Segmental Contrastive Predictive Coding(SCPC) [14, 15]. SCPC introduced the two-level architecture, which looked at both frame and phone-level information for phone and word segmentation. Experimental evidence shows that multi-level information was useful for phone and word segmentation. [28] modified the boundary detector and used reinforcement to learn the segment boundaries. However, these models are unimodal and only utilize speech data. We are utilizing multimodal data. These models are typically trained from speech via the next-term prediction task. Here we combine next-term prediction with the contrastive image-speech retrieval task.

The segmental encoder generates a sequence of word-like units from an utterance. We stack a pretrained text encoder on top of the generating sub-words to extract a sentence embedding. A pretrained image encoder is used to extract image representations. The model tries to align the semantically related images and their captions. By using the text encoder, we want to infuse semantic information in the speech encoder. We use the SpokenCOCO dataset [24] for training and evaluating the proposed method. On the image-speech and speech-image retrieval task, our model significantly outperforms the cascaded variant of SpeechCLIP. In the end, we show competitive performance on the Zerospeech 2021 semantic similarity task [29].

2. Segmental SpeechCLIP

The CLIP model is trained with a large amount of paired image-text data. CLIP uses two encoders for processing images and text separately and learns to align semantically similar images and text captions. The features extracted from CLIP transfer well to other computer vision tasks. We aim to utilize both the text and image encoder to learn speech representations. By cascading the output of the segmental speech encoder with the CLIP text encoder, we aim to induce semantic information in the speech encoder.

The main difference between our proposed method and previously proposed approaches is the word extraction process from the utterances. The segmental speech encoder used for word extraction is summarized in Fig. 2. For the audio encoder, we first use frozen Wav2vec2 to extract audio frame-level features. A trainable segmental audio encoder then extracts subword from the frame-level features. The frozen CLIP text encoder generates sentence embeddings from the sub-words. A frozen CLIP image encoder is used for extracting image embeddings. We describe the various components of the segmental speech encoder in detail below.

2.1. Next Frame Classifier

Let the sequence $\mathbf{X} = (x_1, x_2, ..., x_T)$ represent a waveform. We use a frozen Wav2vec2 followed by a feed-forward network to extract frame level features $\mathbf{Z}(\in \mathbb{R}^{p\times L}) = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_L)$ at low frequency. Each p-dimensional vector $\mathbf{z}_i$ corresponds to a

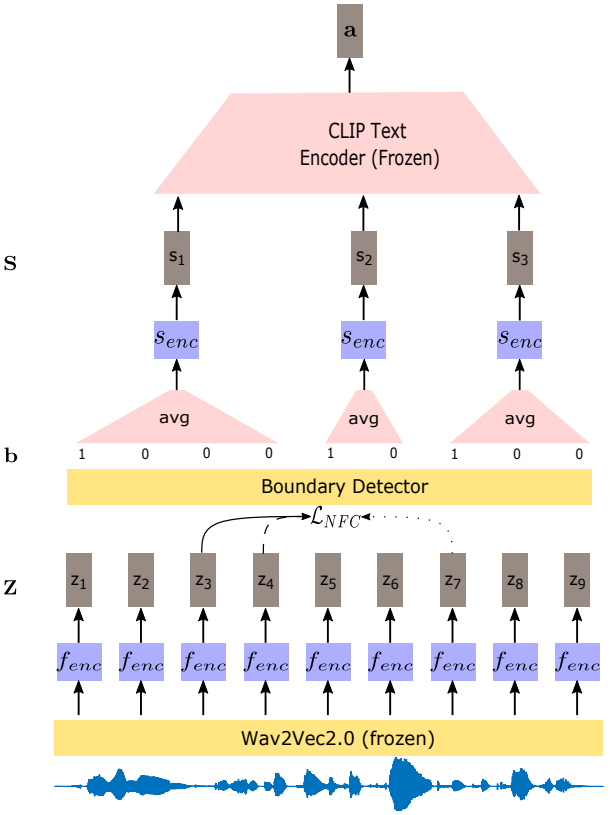

Figure 2: *Overview of the Segmental speech encoder.*

25ms audio frame extracted with a 20 ms shift. Given frame $\mathbf{z}_t$, the encoder tries to classify the next frame $\mathbf{z}_{t+1}$ correctly within a set of $K+1$ representations $\tilde{\mathbf{z}} \in \mathcal{Z}_t$, which include $\mathbf{z}_{t+1}$ and K negative examples, randomly sampled from the same utterance, as

$$\mathcal{L}_{\text{NFC}} = -\log \frac{\exp(\text{sim}(\mathbf{z}_t, \mathbf{z}_{t+1}))}{\sum_{\tilde{\mathbf{z}}\in\mathcal{Z}_t} \exp(\text{sim}(\mathbf{z}_t, \tilde{\mathbf{z}}))} \quad (1)$$

where $\text{sim}(\mathbf{x}, \mathbf{y}) = \frac{\mathbf{x}\mathbf{y}^T}{\|\mathbf{x}\|\|\mathbf{y}\|}$ denotes the cosine similarity.

2.2. Boundary detection

We use the boundary detector from [14], which compares the adjacent frames and output a boundary if the similarity between the adjacent frame falls below a threshold. The boundary detector outputs a sequence of ones and zeros, each one indicating there is a boundary change at that timestep. We generate the segment representations by feeding the average of constituting frames in the segment through a segment encoder, s_{enc}. We use the vectorized computation method from [14] for a faster segment representation calculation. After the boundary detection stage the feature sequence $\mathbf{Z} = (\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_L)$ is segmented into disjoint contiguous segments $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_M)$

2.3. Directly learning the CLIP sub-word representations

In the cascaded SpeechCLIP approach, the audio encoder generates keyword embeddings. Then, argmax is used to find the index of subword embedding in the CLIP vocabulary closest to the keyword. The corresponding subword embedding is used as the keyword embedding. This process uses the argmax operator,

Table 1: *Recall scores for image-speech retrieval on SpokenCOCO 5k test set*

	Image			Speech			Mean		
	R@1	R@5	R@10	R@1	R@5	R@10	R@1	R@5	R@10
Parallel SpeechCLIP [27]	35.8	66.5	78.0	50.6	80.9	89.1	43.2	73.7	83.6
Cascaded SpeechCLIP [27]	6.4	20.7	31.0	9.6	27.7	39.7	8	24.2	35.6
Segmental SpeechCLIP	28.2	55.3	67.5	28.5	56.1	68.9	28.4	55.7	68.2

which is non-differentiable. Straight-through gradient estimator is used for training the model.

Here, we directly generate subword embeddings via the segmental audio encoder. These embeddings are fed into the text encoder bypassing the pretrained vocabulary in the CLIP model. This way, our model can be trained without straight-through estimators. Although, this may not generate exact embeddings from the vocabulary. We pass the segment embeddings $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, ..., \mathbf{s}_M)$ through the CLIP text encoder to generate the audio embedding, $\mathbf{a}$.

2.4. Retrieval loss

Typically, both audio and image encoders are trained in visually grounded models. The image embedding should be closest to the corresponding audio embedding, and the audio should be closest to the corresponding image embedding from a pool of negative examples. The loss is the sum of the two losses. Since we are not training the image encoder, we train the audio encoder to pick the image embedding of the paired image from a set that contains negative examples. It can be considered a classification problem where we classify the paired image embedding from a set with negative examples given the audio embedding. More information on the negative sampling process can be found in section 2.5. The retrieval loss is given as:

$$\mathcal{L}_{\text{RET}} = -log \frac{\exp(\mathbf{a_k} \mathbf{i_k}^T / \tau)}{\sum_{i' \in \mathcal{I}_k} \exp(\mathbf{a_k} \mathbf{i'}^\mathbf{T} / \tau)} \tag{2}$$

where $\mathbf{a_k}$ is the output of the CLIP text encoder, $\mathbf{i_k}$ is the output of the CLIP image encoder and τ is the temperature.

Our model has multiple components, and we train our model progressively. We begin by training the frame-level encoder for a few steps and then add the retrieval loss. The two losses are trained together for the first epoch. For the rest of the training, only the retrieval loss is optimized.

2.5. Negative sampling

Negative sampling is an important part of the contrastive loss. It helps us prevent model collapse. We make the following two changes to the negative sampling process.

2.5.1. Disentangling batch size and negative examples

Typically, the negative samples are sampled from the batch. This, unfortunately, ties the number of negative examples with the batch size. To increase the number of negative examples, we must increase the batch size, which is not always possible on smaller GPUs.

Since we are using frozen pretrained CLIP, i.e., image representation stays the same throughout training. We load all the image representations in advance and sample negative examples from them. This way, we can increase the number of negative examples without increasing the batch size. This is impossible when training the image encoder, as the image representation changes after every update.

2.5.2. Hard negative mining through clustering

The quality of the negative samples impacts the contrastive loss, and sampling better negative examples have been an active research direction. We want to find the closest examples for each entry in the batch and then contrast them. Since the image embeddings are fixed, we can find the closest examples before training the audio encoder. Here we use a simple clustering-based technique to sample harder negative examples.

We first cluster the image embeddings for the dataset into K subsets and store the cluster index for each image embedding. During training, we find the clustered index for each embedding and sample up to 512 examples from that cluster. These form the hard negative examples. The rest of the examples are randomly sampled. Again this is possible because the image embeddings do not change during training.

3. Experiments

3.1. Experimental setup

Dataset: We train the Segmental SpeechCLIP model on SpokenCOCO dataset [24]. Each image in the dataset is paired with five spoken captions. SpokenCOCO contains 123k images and 742 hours of speech. We follow the SpeechCLIP [27] for train/test splits. We use a much smaller split of validation to save time during training. We use image-to-speech and speech-to-image retrieval performance as the evaluation metric.
Model: In our experiments, the wav2vec2 [1] model and the CLIP model are frozen. We use them as feature extractors. The next frame classifier is a two-layer feed-froward network with 1024 hidden units. The segment encoder contains one convolution layer with 1024 filters followed by a single-layer feed-forward network with either 512/768 hidden units for small/large CLIP models. The segmental speech encoder contains approximately 10 million parameters. We use Adam optimizer with 2e-5 learning rate and a batch size of 21. We decay the learning rate by a factor of 0.95 every three epochs. All the experiments are conducted on a single 12GB GPU.

3.2. Retrieval performance

In this section, we evaluate the segmental SpeechCLIP on the image-speech retrieval task to measure how well we can align speech and image embeddings. As shown in Table 1 our proposed model significantly outperforms the cascaded SpeechCLIP model. We almost doubled the performance of cascaded SpeechCLIP. Segmental SpeechCLIP has a slightly lower number of trainable parameters.

We use Wav2vec2.0 as the feature extractor for speech, whereas cascaded SpeechCLIP uses Hubert [30] as the feature extractor. Another big difference is the feature extraction pro-

Table 2: *Semantic similarity scores on the Zerospeech 2021 sSIMI task*

	budget	dev		test	
		syn.	lib.	syn.	lib.
VG baseline	72	9.65	12.61	9.71	0.16
VG baseline	160	9.60	15.09	9.99	-0.10
FaST-VGS+ [25]	468	23.07	**23.10**	15.10	14.32
Seg. SpeechCLIP	72	**28.79**	16.80	**19.60**	**15.69**
Phone topline	1536	9.86	16.11	12.23	20.16

cess; we use the features from a single layer, i.e., layer 11 in the Wav2vec2 transformer encoder, where cascaded SpeechCLIP learns weights to combine the transformer encoder's hidden representations. A weighted combination of layer-wise features from wav2vec2/Hubert tends to work better than single-layer features on downstream tasks [31, 32]. We only use features from a single layer due to GPU memory constraints.

However, the performance is still lower than the parallel variant of SpeechCLIP. Our hypothesis is that the segmentation process and the passing of the segmented speech through the CLIP text encoder loses information.

3.3. Impact of hard mining through clustering

We proposed to use clustering for mining hard negative examples. We explore if this change helps the learning process. We train a system where all the negative examples are sampled randomly and the other where clustering is used for selecting the hard negative examples. Both cases use the same number of negative examples. As seen from 3, hard mining via clustering helps the learning process.

Table 3: *Average retrieval on SpokenCOCO test set. "with" and "without" indicate use/lack of clustering for hard mining negative examples.*

	R@1	R@5	R@10
with	26.1	52.2	64.8
without	22.4	48.6	61.1

3.4. Impact of initial word boundaries

Unsupervised word segmentation has been a growing research area. Recent state-of-the-art solutions utilize multi-modal (paired speech, image) data [23]. Our segmental encoder segments the audio data in sub-word like segments. We experiment with whether using the word boundaries from an existing word segmentation system can be useful. We use the VG-Hubert model for extracting the initial word boundaries [23]. VG-Hubert achieves the best word segmentation performance on TIMIT and Buckeye datasets. We insert the VG-Hubert boundaries in between the boundaries generated by the segmental speech encoder.

As seen from Table 4, using word boundaries have no or a little negative impact on the retrieval performance. This might be either the word boundaries do not help the retrieval task or our insertion process is not optimal. In the future, we plan to explore more ways of utilizing the VG-Hubert boundaries in the segmental SpeechCLIP model.

Table 4: *Average retrieval on SpokenCOCO test set. "with" and "without" indicate use/lack of initial word boundaries.*

	R@1	R@5	R@10
with	22.4	48.6	61.1
without	23.1	49.2	61.5

3.5. Impact of model size

The CLIP model is used as a feature extractor for images and for extracting the final audio representations. We analyze the impact of CLIP model size on retrieval performance. We use CLIP small model (ViT-B/32) with 250 million parameters and the large CLIP model (ViT-L/14) with 422 million parameters. The segmental speech encoders used in the two cases are very similar. For the large CLIP model, the SSE generates 768-dimensional representations, and for the small CLIP model, the output dimension is 512.

As evident from Table 5, the large model helps the retrieval performance. The observation is similar to [27], where the system with large CLIP models outperformed the one with smaller CLIP models.

Table 5: *Average retrieval on SpokenCOCO test set. Small/Large denotes the CLIP model size.*

	R@1	R@5	R@10
Small	26.1	52.2	64.8
Large	28.3	55.7	68.2

4. Semantic representation learning

One of the motivations for utilizing the pretrained CLIP was to learn semantic representations. We use Zerospeech 2021 challenge [29] semantic similarity task, sSIMI, to evaluate the representations' quality. This task aims to compute the similarity between representations of pairs of words and compare it with similarity scores assigned by human annotators.

As seen in Table 2, we perform competitively with a state-of-the-art method. However, there are a few key differences in the methodologies. Our model has fewer parameters and less training time than FaST-VGS+. FaST-VGS+ relies on pretrained R-CNN to generate bounding boxes for the objects in the image; our model does not. FaST-VGS+ can leverage speech-only data via a Masked language modeling task which needs to be improved in our approach. Overall, FaST-VGS+ is trained on SpokenCoCo and Librispeech, whereas we only use SpokenCOCO. This might explain the lower performance on Librispeech test data.

5. Conclusions and future work

Here, we propose to use a segmental speech encoder to distill information from the pretrained CLIP model. We modify the contrastive loss and the negative sampling to lower computational requirements for training such systems. We outperform the cascaded speechCLIP model on the retrieval task. On the semantic similarity task, we achieve comparable performance to SOTA systems.

In the future, we want to explore a weighted combination of layer-wise features from HuBERT as input to our system. We also want to modify our model to utilize speech-only data via masked language modeling.

6. References

[1] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv preprint arXiv:2006.11477*, 2020.

[2] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end asr: from supervised to semi-supervised learning with modern architectures," *arXiv preprint arXiv:1911.08460*, 2019.

[3] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, *et al.*, "Transformer-based acoustic modeling for hybrid speech recognition," *arXiv preprint arXiv:1910.09799*, 2019.

[4] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 7634–7638, IEEE, 2014.

[5] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pp. 40–49, Association for Computational Linguistics, 2012.

[6] M.-h. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an hmm-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[7] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

[8] S. Bhati, S. Nayak, and K. S. R. Murty, "Unsupervised speech signal to symbol transformation for zero resource speech applications," *Proc. Interspeech 2017*, pp. 2133–2137, 2017.

[9] H. Kamper, K. Livescu, and S. Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 719–726, IEEE, 2017.

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

[11] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[12] S. Schneider, A. Baevski, R. Collobert, and M. Auli, "wav2vec: Unsupervised pre-training for speech recognition," *arXiv preprint arXiv:1904.05862*, 2019.

[13] Y.-A. Chung, Y. Zhang, W. Han, C.-C. Chiu, J. Qin, R. Pang, and Y. Wu, "W2v-bert: Combining contrastive learning and masked language modeling for self-supervised speech pre-training," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 244–250, IEEE, 2021.

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

[15] S. Bhati, J. Villalba, P. Żelasko, L. Moro-Velazquez, and N. Dehak, "Unsupervised speech segmentation and variable rate representation learning using segmental contrastive predictive coding," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 30, pp. 2002–2014, 2022.

[16] A. T. Liu, S.-w. Yang, P.-H. Chi, P.-c. Hsu, and H.-y. Lee, "Mockingjay: Unsupervised speech representation learning with deep bidirectional transformer encoders," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6419–6423, IEEE, 2020.

[17] A. T. Liu, S.-W. Li, and H.-y. Lee, "Tera: Self-supervised learning of transformer encoder representation for speech," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 2351–2366, 2021.

[18] Y.-C. Chen, L. Li, L. Yu, A. El Kholy, F. Ahmed, Z. Gan, Y. Cheng, and J. Liu, "Uniter: Universal image-text representation learning," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXX*, pp. 104–120, Springer, 2020.

[19] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*, pp. 8748–8763, PMLR, 2021.

[20] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass, "Jointly discovering visual objects and spoken words from raw sensory input," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 649–665, 2018.

[21] P. Peng and D. Harwath, "Fast-slow transformer for visually grounding speech," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7727–7731, IEEE, 2022.

[22] W.-N. Hsu, D. Harwath, and J. Glass, "Transfer learning from audio-visual grounding to speech recognition," *arXiv preprint arXiv:1907.04355*, 2019.

[23] P. Peng and D. Harwath, "Word discovery in visually grounded, self-supervised speech models," *arXiv preprint arXiv:2203.15081*, 2022.

[24] W.-N. Hsu, D. Harwath, C. Song, and J. Glass, "Text-free image-to-speech synthesis using learned segmental units," *arXiv preprint arXiv:2012.15454*, 2020.

[25] P. Peng and D. Harwath, "Self-supervised representation learning for speech using visual grounding and masked language modeling," *arXiv preprint arXiv:2202.03543*, 2022.

[26] H.-H. Wu, P. Seetharaman, K. Kumar, and J. P. Bello, "Wav2clip: Learning robust audio representations from clip," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4563–4567, IEEE, 2022.

[27] Y.-J. Shih, H.-F. Wang, H.-J. Chang, L. Berry, H.-y. Lee, and D. Harwath, "Speechclip: Integrating speech with pre-trained vision and language model," *arXiv preprint arXiv:2210.00705*, 2022.

[28] S. Cuervo, A. Łańcucki, R. Marxer, P. Rychlikowski, and J. Chorowski, "Variable-rate hierarchical cpc leads to acoustic unit discovery in speech," *arXiv preprint arXiv:2206.02211*, 2022.

[29] E. Dunbar, M. Bernard, N. Hamilakis, T. A. Nguyen, M. De Seyssel, P. Rozé, M. Rivière, E. Kharitonov, and E. Dupoux, "The zero resource speech challenge 2021: Spoken language modelling," *arXiv preprint arXiv:2104.14700*, 2021.

[30] 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*, vol. 29, pp. 3451–3460, 2021.

[31] S.-w. Yang, P.-H. Chi, Y.-S. Chuang, C.-I. J. Lai, K. Lakhotia, Y. Y. Lin, A. T. Liu, J. Shi, X. Chang, G.-T. Lin, *et al.*, "Superb: Speech processing universal performance benchmark," *arXiv preprint arXiv:2105.01051*, 2021.

[32] H.-S. Tsai, H.-J. Chang, W.-C. Huang, Z. Huang, K. Lakhotia, S.-w. Yang, S. Dong, A. T. Liu, C.-I. J. Lai, J. Shi, *et al.*, "Superb-sg: Enhanced speech processing universal performance benchmark for semantic and generative capabilities," *arXiv preprint arXiv:2203.06849*, 2022.