MOCKINGJAY: UNSUPERVISED SPEECH REPRESENTATION LEARNING WITH DEEP BIDIRECTIONAL TRANSFORMER ENCODERS

Andy T. Liu *Shu-wen Yang* *Po-Han Chi* *Po-chun Hsu* *Hung-yi Lee*

National Taiwan University
College of Electrical Engineering and Computer Science
{r07942089, r08944041, r08942074, r07942095, hungyilee}@ntu.edu.tw

ABSTRACT

We present Mockingjay as a new speech representation learning approach, where bidirectional Transformer encoders are pre-trained on a large amount of unlabeled speech. Previous speech representation methods learn through conditioning on past frames and predicting information about future frames. Whereas Mockingjay is designed to predict the current frame through jointly conditioning on both past and future contexts. The Mockingjay representation improves performance for a wide range of downstream tasks, including phoneme classification, speaker recognition, and sentiment classification on spoken content, while outperforming other approaches. Mockingjay is empirically powerful and can be fine-tuned with downstream models, with only 2 epochs we further improve performance dramatically. In a low resource setting with only 0.1% of labeled data, we outperform the result of Mel-features that uses all 100% labeled data.

Index Terms— speech representation learning, unsupervised training, transformer encoders, low resource

1. INTRODUCTION

The goal of speech representation learning is to find a transform from speech that makes high-level information more accessible to SLP (Speech and Language Processing) downstream tasks, as speech signal possess a rich set of acoustic and linguistic content, including phonemes, words, semantic meanings, tone, speaker characteristics, and even sentiment information. In this paper, we propose Mockingjay to learn speech representations through unsupervised training without the use of any labels. We use multi-layer transformer encoders and multi-head self-attention [1] to achieve bidirectional encoding; this framework allows our model to consider past and future contexts at the same time. To achieve unsupervised pre-training for speech representations, Mockingjay learns under the proposed Masked Acoustic Model (MAM) task. During training, masked frames are given, and the model learns to reconstruct and predict the original frames. Hence we gave the name Mockingjay, a bird that mimics sound. The proposed framework is illustrated in Figure 1.

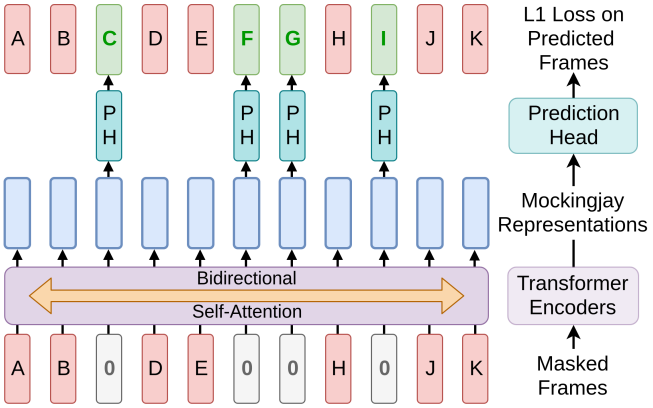

Fig. 1. The proposed Masked Acoustic Model pre-training task, 15% of input the frames are masked to zero at random.

1.1. Related work

Unsupervised speech representation learning [2, 3, 4, 5, 6, 7, 8, 9, 10] is effective in extracting high-level properties from speech. SLP downstream tasks can be improved through speech representations because surface features such as log Mel-spectrograms or waveform can poorly reveal the abundant information within speech.

Contrastive Predictive Coding (CPC) [5] and wav2vec [7] use a multi-layer CNN to encode past context, representations are learned by predicting the future in latent space under a contrastive binary classification task. Autoregressive Predictive Coding (APC) [6] uses autoregressive models to encode temporal information of past acoustic sequence; the model predicts future frames like an RNN-based language model [11], optimized with reconstruction loss. Unidirectional models are commonly used in the previous approaches [2, 3, 4, 5, 6, 7]. However, this constraint on model architectures limits the potential of speech representation learning.

The recently proposed vq-wav2vec [8] approach attempts to apply the well-performing Natural Language Processing (NLP) algorithm BERT [12] on continuous speech. Input speech is discretized to a K-way quantized embedding space, so continuous speech could act like discrete units similar to word tokens in NLP tasks. In vq-wav2vec [8], an exhaustive

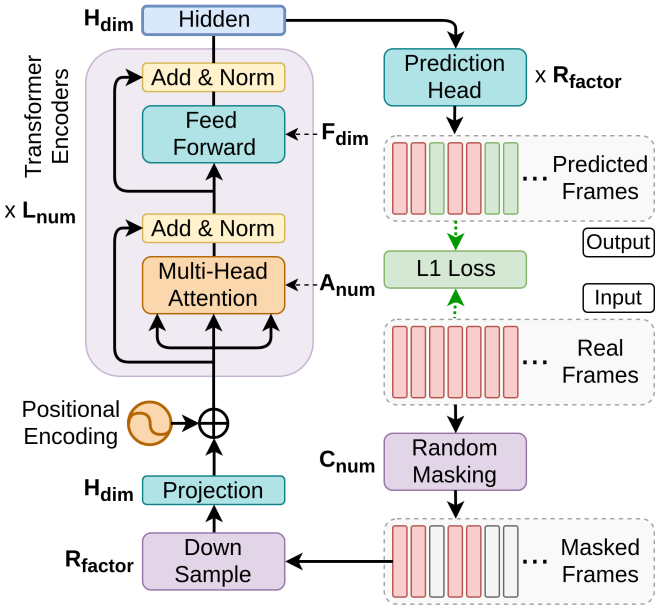

Fig. 2. The proposed Mockingjay training framework.

two-stage training pipeline with massive computing resources are required to adapt speech to NLP algorithm, as the quantization process is against the continuous nature of speech. Unlike [8] that adapts speech to BERT [12] through quantization, the proposed approach can be seen as a modified version of BERT [12] for direct application on continuous speech.

1.2. Proposed Method

Unlike previous left-to-right unidirectional approaches that only consider past sequences to predict information about future frames, the proposed method allows us to train a bidirectional speech representation model, alleviating the unidirectionality constraint of previous methods. As a result, the Mockingjay model obtains substantial improvements in several SLP tasks. Moreover, as previous approaches restrict the power of the pre-trained models to representation extraction only [5, 6, 7, 8], the proposed method is robust as it can be fine-tuned easily on downstream tasks. We show that fine-tuning for 2 epochs easily acquires significant improvement.

The proposed approach outperforms other representations and features. When compared to the commonly used log Mel-features, we outperformed it by 35.2% (absolute improvement) for phoneme classification accuracy, 28.0% (absolute improvement) for speaker recognition accuracy, and 6.4% (absolute improvement) for sentiment discrimination accuracy on a spoken content dataset unseen during pre-train. We also experiment in low resource settings to show that Mockingjay is capable of improving supervised training in real-life low-resource scenarios. With only 0.36 hours (0.1%) of transcribed speech, the proposed approach outperforms Mel-features with 360 hours (100%) of labels.

2. MOCKINGJAY

In this section, we first introduce model architecture and its designs, secondly we explain the proposed unsupervised context prediction task, and finally we explain how the proposed model is used with downstream task models.

2.1. Model Architecture

We use a multi-layer Transformer encoder with multi-head self-attention for left-and-right bidirectional encoding, this architecture is illustrated in Figure 2. Each encoder layer has two sub-layers, the first is a multi-head self-attention network, and the second is a feed-forward layer, each sub-layer has a residual connection followed by layer normalization [13], based on the design described in [1]. All encoder layers in the model, as well as the sub-layers, produce outputs of identical dimensions denoted as H_{dim}. In Figure 2, we denote the feed-forward size as F_{dim}, the number of self-attention heads as A_{num}, and the total of Transformer layers as L_{num}. The Mockingjay representations can be extracted from the Transformer encoders' hidden state and labeled as $Hidden$, we explain how they are used as representations in Section 2.3.

Since Transformer encoders contain no recurrence and convolution, we use positional encoding to make our model aware of the input sequence order [1]. As direct addition of acoustic features to positional encoding may lead to potential training failure [14], the input frames are first projected linearly to the hidden dimension of H_{dim} before adding with positional encoding [15]. We use sinusoidal positional encoding instead of learnable positional embeddings [16] because acoustic features can be arbitrarily long with high variance [15]. We apply downsampling on input features to adapt our model to long sequences. To reduce the length of frames by a factor of R_{factor}, we use the reshape technique from [14, 15] by stacking R_{factor} consecutive frames into one step.

2.2. Masked Acoustic Modeling

We propose the Masked Acoustic Modeling task, where we randomly select 15% of the input frames, and the model predicts the selected frames based on its left and right context, as illustrated in Figure 1. During training, we add a prediction head consists of two layers of feed-forward network with layer-normalization, using the last encoder layer as it's input. We use L1 Loss to minimize reconstruction error between prediction and ground-truth frames on the selected 15%. The prediction head is not used once the model is trained.

During training, for the selected 15% of frames, 1) we mask it all to zero 80% of the time, 2) replace all with a random frame 10% of the time, and 3) leave the frames untouched 10% of the time.[1] We introduce this sub-random

[1]This process is similar to the Cloze task [17] and the Masked Language Model task from BERT [12], but we mask frames of speech to zero instead of using the MASK token.

process instead of always masking the frames to alleviate the mismatch between training and inference, as masked frames do not appear during inference time. Note that in contrast to BERT [12], where the sub-random process is performed token-wise on the i-th chosen token, our sub-random process is performed utterance-wise. In other words, our model may receive inputs as ground-truth frames for 3) 10% of the time, rather with some of the inputs always augmented as in [12].

To avoid the model exploiting local smoothness of acoustic frames, we propose additional consecutive masking where we mask consecutive frames C_{num} to zero. The model is required to infer on global structure rather than local information. We also use dynamic masking [18] where masking patterns are sampled from an uniform distribution every time we feed a sequence to the model, unlike the static mask employed in [12] where masking is performed during data preprocessing. We only use a single context prediction task to train our representation model, as suggested by [18]. Unlike BERT [12] and ALBERT [19] that needs two tasks to train their language model. In our preliminary experiments, we found that the sentence prediction task used in [12, 19] is not helpful, as additional tasks can potentially harm training behavior. We do not include details due to space limitations.

2.3. Incorporating with Downstream Tasks

Mockingjay representations are essentially the Transformer encoder hidden states. There are many ways to incorporate learned representations to downstream tasks. In this work, we mainly extract representations from the last layer. However, we also expose the deep internals of Mockingjay to downstream models, where we use a mixture of representations from all layers, similar to the ELMO [20] approach. In other words, we use a learnable weighted sum to integrate hidden states from all layers. Last but not least, a pre-trained Mockingjay model can be fine-tuned with downstream models to create improving results, we update the pre-trained Mockingjay together with random initialized downstream task models.

3. IMPLEMENTATION

In this work, we use two types of features as our model's output reconstruction target: the Mel-scale spectrogram and the linear-scale spectrogram. As Mel-scale spectrogram is a more concise acoustic feature when compared to linear-scale spectrogram, we propose two model settings: *BASE* and *LARGE*. Both of these models take Mel-features as input, and transform input Mel-features into high-level representations. They use the same hidden dimension size of H_{dim}=768, feed-forward size of F_{dim}=3072, and attention heads of A_{num}=12, with the exception of layer number L_{num}, downsample factor R_{factor}, and consecutive masking number C_{num}, the differences in model settings are listed in Table 1. We further analyze their differences in our experiment section.

Table 1. The proposed BASE and LARGE model settings

Model	BASE	LARGE
Target	Mel	Linear
L_{num}	3	12
R_{factor}	1	3
C_{num}	7	3
parameters	21.4M	85.4M

The proposed Mockingjay models are pre-trained on the LibriSpeech [21] corpus train-clean-360 subset. We use Adam [22] where learning rate is warmed up over the first 7% of 500k total training steps to a peak value of 4e-4 and then linearly decayed. A dropout [23] of 0.1 is applied on all layers and attention weights. For downstream task fine-tuning, most of the hyperparameters are the same as in pre-training, with the exception of a learning rate of 4e-3, and the number of training epochs is set to 2 (which is approximately 50k steps). We train with a batch size of 6 using a single 1080Ti GPU. We provide pre-trained models with our implementation, they are publicly available for reproducibility[2].

4. EXPERIMENT

Following previous works [2, 3, 4, 5, 6, 7, 8], we evaluate different features and representations on downstream tasks, including: phoneme classification, speaker recognition, and sentiment classification on spoken content. For a fair comparison, each downstream task uses an identical model architecture and hyperparameters despite different input features.

We report results from 5 of our settings: 1) *BASE* and 2) *LARGE* where Mockingjay representations are extracted from the last encoder layer, 3) the *BASE-FT2* where we fine-tune *BASE* with random initialized downstream models for 2 epochs, and 4) the *BASE-FT500* where we fine-tune for 500k steps, and finally 5) the *LARGE-WS* where we incorporate hidden states from all encoder layers of the *LARGE* model through a learnable weighted sum. We did not fine-tune the *LARGE* model, as it is meant for extracting representations. Empirically we found that even with supervised training, a random initialized Mockingjay model followed by any downstream model is hard to be trained from scratch. This shows that the proposed pre-training is essentially indispensable.

4.1. Comparing with other representations

The proposed approaches are mainly compared with APC [6] representations, as they also experiment on phone classification and speaker verification. As reported in [6], the APC approach outperformed CPC representations [5, 7, 9] in both two tasks, which makes APC suitable as a strong baseline. APC uses an unidirectional autoregressive model. We compare the proposed approach with APC to show that our bidirectional approach has advantages in speech representation

[2]https://github.com/andi611/Mockingjay-Speech-Representation

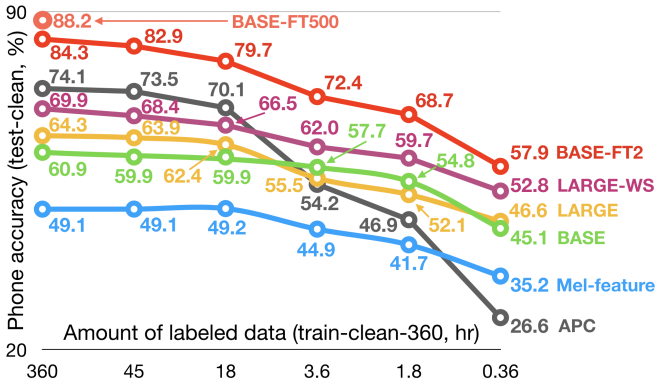

Fig. 3. Comparing representations with phone classification accuracy across different amount of transcribed data.

learning. For fair comparison, we pre-train APC using their official implementations with the reported ideal parameters and settings, but expand the model's hidden size to H_{dim}=768 to match ours. We also report results on 160-dimensional log Mel-features, which helps evaluate the accessibility of speech information from regular acoustic features.

4.2. Phoneme Classification

To measure the accessibility of phonetic information, we train linear phone classifiers using Mel-features, APC and Mockingjay representations from the LibriSpeech train-clean-360 subset. We obtain force-aligned phoneme sequences with the Montreal Forced Aligner [24], where there are 72 possible phone classes. Testing results on the LibriSpeech test-clean subset are presented in Figure 3. In the case where all 360 hours of labels are used to train the classifier, *BASE* and *LARGE* representations increase 11.8% and 15.2% accuracy from Mel-features. The *BASE-FT2* model outperforms all representations after 2 epochs of fine-tuning, with 10.2% and 35.2% absolute improvement over APC and Mel-features, respectively. We observe that fine-tuning for 2 epochs is enough to reveal our method's potential as there is only a small gap (3.9%) between *BASE-FT2* and *BASE-FT500*. Furthermore, *LARGE-WS* improves over *LARGE*, just as we expected.

To demonstrate how pre-training on speech can improve supervised training in resource constrained scenarios where human labels are short, we train the classifier with reduced amount of training data. The performance variation of different methods are plotted in Figure 3, where we measure over various intervals of constrained training data to observe performance drop. Both *BASE* and *LARGE* increase accuracy over Mel-features across various amount of transcribed data. Whereas the APC approach performs well on the full resource but fails to generalize for limited amount of labeled data. In the case where there are only 0.36 hours of data available, we improve accuracy by 22.7% (an absolute improvement from Mel-features). Note that with only 0.36 hours (0.1%) of labels available, *BASE-FT2* (57.9% acc) even outperformed

Table 2. Comparing different methods with different tasks.

Methods	Speaker (acc)	Sentiment (acc)
Mel-Features	70.06	64.63
APC	85.88	65.97
Base	94.54	67.38
BaseFT2	**98.05**	68.45
Large	96.26	70.07
LargeWS	96.40	**71.05**

Mel-features (49.1% acc) that uses all 360 hours (100%) of labeled data. We conclude that pre-training Mockingjay on speech substantially improves the performance on supervised task that requires human annotations.

4.3. Speaker Recognition

To demonstrate that the proposed approach performs constantly for all SLP downstream tasks, we report speaker recognition results on the LibriSpeech 100 hour selected subset, where train/test split is performed randomly with a 9:1 ratio, and there are 63 possible speakers. We trained a simple one-layer RNN classifier for speaker recognition using different representations, results are listed in Table 2 for comparison. The proposed *BASE* and *LARGE* representations outperformed both APC and Mel-Features. *BASE-FT2* further improves upon *BASE* while achieving the highest accuracy, whereas *LARGE-WS* also outperforms *LARGE*.

4.4. Sentiment Classification on Spoken Content

To demonstrate domain invariant transferability of the proposed representation across different datasets, the Mockingjay model is pre-trained on LibriSpeech and applied on the MOSEI [25] dataset. We also use a simple one-layer RNN classifier, where the model is trained to extract linguistic meanings from speech and discriminates between sentiments. The results listed in Table 2 lead to an identical conclusion as in the speaker recognition task discussed above. Except that in the case of sentiment classification, *LARGE-WS* achieved the highest score without the need of fine-tuning, demonstrating that a deeper model has great potential for extracting general speech representations. To conclude this section, we claim that the proposed representations are general and can be used on datasets with various unseen domains.

5. CONCLUSION

The proposed representation contains a variety of knowledge, including but not limited to phonetic, speaker, and sentiment information. We improve performance for a wide range of downstream tasks, and show promising results in low resource settings, as the learned speech representations are robust and can be transferred to different tasks across different datasets. In future work, we will investigate and deploy Mockingjay representations on more downstream SLP tasks, including ASR, voice conversion, and speech translation.

6. REFERENCES

[1] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," 2017.

[2] Jan Chorowski, Ron J. Weiss, Samy Bengio, and Aaron van den Oord, "Unsupervised speech representation learning using wavenet autoencoders," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 27, no. 12, pp. 2041–2053, Dec 2019.

[3] Yu-An Chung and James Glass, "Speech2vec: A sequence-to-sequence framework for learning word embeddings from speech," *Interspeech 2018*, Sep 2018.

[4] Yu-An Chung, Chao-Chung Wu, Chia-Hao Shen, Hung-Yi Lee, and Lin-Shan Lee, "Audio word2vec: Unsupervised learning of audio segment representations using sequence-to-sequence autoencoder," 2016.

[5] Aaron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," 2018.

[6] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," *Interspeech*, Sep 2019.

[7] Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli, "wav2vec: Unsupervised pre-training for speech recognition," *Interspeech*, Sep 2019.

[8] Anonymous authors, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR 2020 Conference Blind Submission*, 2020.

[9] Anonymous authors, "Unsupervised learning of efficient and robust speech representations," in *ICLR 2020 Conference Blind Submission*, 2020.

[10] Andy T. Liu, Po chun Hsu, and Hung yi Lee, "Unsupervised end-to-end learning of discrete linguistic units for voice conversion," 2019.

[11] Tomáš Mikolov, Martin Karafiát, Lukáš Burget, Jan Černockỳ, and Sanjeev Khudanpur, "Recurrent neural network based language model," in *Eleventh annual conference of the international speech communication association*, 2010.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[13] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton, "Layer normalization," 2016.

[14] Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel, "Self-attentional acoustic models," *Interspeech 2018*, Sep 2018.

[15] Ngoc-Quan Pham, Thai-Son Nguyen, Jan Niehues, Markus Muller, and Alex Waibel, "Very deep self-attention networks for end-to-end speech recognition," *arXiv preprint arXiv:1904.13377*, 2019.

[16] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin, "Convolutional sequence to sequence learning," 2017.

[17] Wilson L Taylor, ""cloze procedure": A new tool for measuring readability," *Journalism Bulletin*, vol. 30, no. 4, pp. 415–433, 1953.

[18] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[19] Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut, "Albert: A lite bert for self-supervised learning of language representations," 2019.

[20] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer, "Deep contextualized word representations," 2018.

[21] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 5206–5210.

[22] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2014.

[23] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014.

[24] Michael McAuliffe, Michaela Socolof, Sarah Mihuc, Michael Wagner, and Morgan Sonderegger, "Montreal forced aligner: Trainable text-speech alignment using kaldi.," 2017.

[25] AmirAli Bagher Zadeh, Paul Pu Liang, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency, "Multimodal language analysis in the wild: CMU-MOSEI dataset and interpretable dynamic fusion graph," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia, July 2018, pp. 2236–2246.