UNSUPERVISED AND SEMI-SUPERVISED FEW-SHOT ACOUSTIC EVENT CLASSIFICATION

Hsin-Ping Huang[1,2] *Krishna C. Puvvada*[2] *Ming Sun*[2] *Chao Wang*[2]

[1] University of California, Merced [2] Amazon Alexa

ABSTRACT

Few-shot Acoustic Event Classification (AEC) aims to learn a model to recognize novel acoustic events using very limited labeled data. Previous works utilize supervised pre-training as well as meta-learning approaches, which heavily rely on labeled data. Here, we study unsupervised and semi-supervised learning approaches for few-shot AEC. Our work builds upon recent advances in unsupervised representation learning introduced for speech recognition and language modeling. We learn audio representations from a large amount of unlabeled data, and use the resulting representations for few-shot AEC. We further extend our model in a semi-supervised fashion. Our unsupervised representation learning approach outperforms supervised pre-training methods, and our semi-supervised learning approach outperforms meta-learning methods for few-shot AEC. We also show that our work is more robust under domain mismatch.

Index Terms— acoustic event classification, few-shot learning, representation learning, semi-supervised learning

1. INTRODUCTION

Acoustic Event Classification (AEC) aims to identify whether certain events occur in an audio clip. Models achieving high performance typically rely on a large amount of labeled data [1]. However, scenarios such as rare event classification require building models with limited labeled data. Further, building personalized audio event detectors for smart home assistants such as Alexa, Google Home or Siri requires building models with extremely limited data (e.g., < 10 samples). Few-shot AEC via meta-learning has been proposed as a solution for building models in low data regimes [2].

Briefly, few-shot learning methods strive to build classifiers that generalize well with a few examples for classes not seen during training. Popular methods for few-shot learning follow episodic training paradigm of meta-learning, wherein training proceeds by mimicking the few-shot problems that need to be solved at test time [3, 4, 5]. Another area of deep learning which deals with none, or limited labeled data is unsupervised or semi-supervised learning. After the initial wave of few-shot learning studies, several works attempt to augment few-shot learning with the benefit of learning from unlabeled data. Examples include augmenting the labeled set with additional unlabeled samples through pseudo-labeling [6, 7], as well as including auxiliary tasks along with few-shot classification in a multi-task paradigm [8].

While the added benefit of unlabeled data in few-shot learning is clear, the general consensus is that unsupervised or semi-supervised learning alone is not as efficient as meta-learning methods for few-shot classification tasks. However, given the recent developments in speech/language sequence modeling [9, 10, 11, 12], we visit the problem of few-shot AEC using unsupervised and semi-supervised learning approaches. We make the following contributions: (1) We introduce a new unsupervised framework for few-shot AEC, built upon unsupervised representation learning and (2) We propose a simple semi-supervised extension to our framework. Quantitative results show that the proposed approach achieves improvement over existing supervised meta-learning approaches for few-shot AEC.

2. OVERVIEW

Acoustic Event Classification (AEC) is a multi-label classification problem. Given an input audio clip $\mathbf{x}$, the goal is to train a model F to predict a multi-hot vector $\mathbf{y} = [y^1, y^2, ..., y^C]$, where y^i is a binary value that indicates whether or not event i occurs in the audio clip $\mathbf{x}$. C is the total number of target events. Supervised few-shot acoustic event classification [2] aims to train a model using a large set of labeled base class data C_{train}. The goal is to detect novel class data C_{test} with only few labeled samples for each novel event available. A validation set C_{valid} is used for hyper-parameter tuning. Here C_{train}, C_{valid} and C_{test} are non-overlapping, so the model should learn to generalize across novel classes.

At test time, for an N-way K-shot few-shot classification task, we sample N classes from the novel class data C_{test} to form an N-class classification task t. The task t consists of a labeled support set S including K positive samples from each class and a total of K^- negative samples that do not belong to any of the N sampled classes. The support set S is used to train the model. We also form a query set Q sampled from the same N classes. The model is evaluated by classifying the query set samples into N classes. The performance is

reported as average AUC of T sampled few-shot classification tasks.

3. METHODS

We briefly describe the supervised pre-training and the best performing meta-learning approach (prototypical network) [2] in Section 3.1 and Section 3.2 respectively. We introduce the proposed unsupervised approach for few-shot AEC in Section 3.3 and its semi-supervised extension in Section 3.4.

3.1. Supervised pre-training

Supervised pre-training method trains encoder f_E and classifier f_C performing multi-label classification task with cross-entropy loss using the whole base class data C_{train}. Here f_E maps an audio sample into latent space, and the classifier f_C^i maps the latent embedding to probability of class i.

$$L_{cls} = -\sum_{(\mathbf{x},\mathbf{y})} \sum_{i=1}^{C_{train}} y^i \log f_C^i(f_E(\mathbf{x})) + (1-y^i)\log(1-f_C^i(f_E(\mathbf{x}))) \tag{1}$$

At test time, we discard the classifier f_C and only use the encoder f_E as the few-shot classification model F. The model F directly classifies the query set samples based on their distance to the support set samples. Specifically, for an N-way K-shot task, for each novel class i in total N novel classes, the support set consists of K positive sample $\mathbf{x}^{s+}$ and K^- negative samples $\mathbf{x}^{s-}$. As shown in (2), we use the model F to project samples into embedding space. We then calculate the average distance of a query sample $\mathbf{x}^q$ to each positive sample $\mathbf{x}^{s+}$ and each negative sample $\mathbf{x}^{s-}$. We use the average distance of $\mathbf{x}^q$ to $\mathbf{x}^{s+}$ to represent how far $\mathbf{x}^q$ is from being class i, denoted as $\hat{d}_i^+$; We use the average distance of $\mathbf{x}^q$ to $\mathbf{x}^{s-}$ to indicate how far $\mathbf{x}^q$ is from not being class i, denoted as $\hat{d}_i^-$. Finally, we apply softmax function to obtain the binary probability $\hat{y}^i$ of detecting that class i occurs in the query sample $\mathbf{x}^q$. Here we use cosine similarity as the distance metric D.

$$\hat{d}_i^+ = \frac{1}{K}\sum_{K} D(F(\mathbf{x}^{s+}), F(\mathbf{x}^q)),\ \hat{d}_i^- = \frac{1}{K^-}\sum_{K^-} D(F(\mathbf{x}^{s-}), F(\mathbf{x}^q))$$
$$p(\hat{y}^i = 1) = \frac{exp(-\hat{d}_i^+)}{exp(-\hat{d}_i^-) + exp(-\hat{d}_i^+)} \tag{2}$$

3.2. Prototypical network

Prototypical network (ProtoNet) samples support set and query set from the base class C_{train} to form few-shot training tasks to train the model $F = f_E$. Note that there is no classifier f_C in prototypical network training. Specifically, for an N-way K-shot task, the N classes are sampled from the base class data C_{train} to form an N-class classification task t including a labeled support set S and a labeled query set Q. For each query set sample $\mathbf{x}^q$, we calculate the probability of whether class i occurs in it or not by (2). The model F is trained to optimize the cross-entropy loss for the query set samples. At test time, given the few-shot task sampled from C_{test}, we also use (2) to classify the novel class events.

$$L_{proto} = -\sum_{(\mathbf{x},\mathbf{y})} \sum_{i=1}^{N} y^i \log p(\hat{y}^i) + (1-y^i)\log(1-p(\hat{y}^i)) \tag{3}$$

3.3. Unsupervised few-shot AEC

In our unsupervised few-shot AEC approach, we adopt unsupervised representation learning method to learn the model $F = f_E$ for mapping raw audio signals $\mathbf{x}$ into latent embeddings $F(\mathbf{x})$. At test time, the latent embeddings $F(\mathbf{x})$ are then used to classify query samples into novel classes following the procedure in (2). We utilize the vq-wav2vec along with BERT model as our unsupervised representation learning method. The vq-wav2vec model [11] first turns the raw audio signal into discrete indices (Section 3.3.1). These discrete indices are in turn used as inputs to the BERT model to obtain deep contextual representations (Section 3.3.2).

3.3.1. Vq-wav2vec

Wav2vec model takes raw audio signal $\mathbf{x}$ as input and applies two modules. The encoder network $\mathbf{E}$ encodes the audio signal $\mathbf{x}$ into a sequence of signal embeddings $\{\mathbf{z}\}$, and the context network $\mathbf{C}$ combines multiple time-steps of the embedding $\mathbf{z}$ to obtain contextualized representations $\{\mathbf{c}\}$. Vq-wav2vec model has the same $\mathbf{E}$ and $\mathbf{C}$ as wav2vec model, albeit with a vector quantization module $\mathbf{Q}$ inserted between $\mathbf{E}$ and $\mathbf{C}$ that turns the signal embedding $\{\mathbf{z}\}$ into discrete indices and the quantized embedding $\{\hat{\mathbf{z}}\}$. The context network $\mathbf{C}$ then combines multiple time-steps of $\hat{\mathbf{z}}$ to obtain contextualized representations $\{\mathbf{c}\}$.

The vq-wav2vec model learns vector-quantized representations for raw audio signals by performing future signal prediction task. We use $\mathbf{z}_t$ to denote the embedding of the audio signal at current time step, and $\mathbf{z}_{t+k}$ is the embedding of audio signal that is k steps into the future. The networks $\mathbf{E}$, $\mathbf{Q}$, and $\mathbf{C}$ are optimized by minimizing the contrastive loss of predicting whether a given sample embedding is a positive sample $\mathbf{z}_{t+k}$ or a negative sample $\tilde{\mathbf{z}}$. $\tilde{\mathbf{z}}$ is chosen uniformly from other time instances of the same audio clip.

As shown in (4), given an audio signal at current time step $\mathbf{x}_t$, the model first extracts the signal context $\mathbf{c}_t$. Then, it applies an affine transformation layer h_k to obtain the transformed context $h_k(\mathbf{c}_t)$. The probability of predicting a positive example is calculated by $\mathbf{z}_{t+k} \cdot h_k(\mathbf{c}_t)$ and the probability of predicting a negative sample is calculated by $\tilde{\mathbf{z}} \cdot h_k(\mathbf{c}_t)$. For each time step k we have different affine transformation layers h_k, and the loss function is summed up over different step

sizes k, $L_{wav2vec} = \sum_{k=1}^{K} L_{wav2vec}^{k}$. T denotes the sequence length and λ is set to 10, indicating there are 10 negative samples for each positive sample.

$$L_{wav2vec}^{k} = -\sum_{t \in T}(\log \sigma(\mathbf{z}_{t+k} \cdot h_k(\mathbf{c}_t)) + \sum_{\lambda} \log(\sigma(-\tilde{\mathbf{z}} \cdot h_k(\mathbf{c}_t)))) \tag{4}$$

The vector quantization module $\mathbf{Q}$ turns the sequence of continuous embeddings $\{\mathbf{z}\}$ into a sequence of quantized vectors. Each of the quantized vectors $\hat{\mathbf{z}}$ with indices $\mathbf{w}$ is selected from a learnable codebook that contains $\mathbf{W}$ codeword vectors of the same size as $\mathbf{z}$. We utilize the Gumbel-softmax method [13] to train $\mathbf{Q}$. Note that both wav2vec and vq-wav2vec models can be used to extract representations $\mathbf{c}$ for few-shot testing, and are included as baselines in Section 5.1.

3.3.2. BERT

Once trained, the encoder network $\mathbf{E}$ and the quantization module $\mathbf{Q}$ of vq-wav2vec model are used to transform input audio $\mathbf{x}$ to a sequence of codebook indices $\{\mathbf{w}\}$. These discrete token sequences $\{\mathbf{w}\}$ are used as inputs to train the BERT model. BERT [14] has a few transformer layers that build a deep contextualized representation. Following masked language modeling paradigm, some input tokens are randomly masked out during training, and the model is trained to predict the missing tokens. As it is too trivial for the BERT model to predict a single missing token due to smooth varying nature of input audios, we choose 5% tokens to form the start indices of the masks, and mask 20 consecutive tokens following [11]. Once the BERT model is trained, we use it as model $F = f_E$ that maps the discrete audio token sequences $\{\mathbf{w}\}$ into embedding space. We concatenate outputs of 3 transformer layers of the BERT model as the final embeddings for few-shot evaluation. We use mean function to aggregate the embeddings of different time steps.

3.4. Semi-supervised few-shot AEC

We further extend our model to a semi-supervised approach. After we train the model f_E using the unsupervised representation learning method described in Section 3.3, we append a feed-forward layer f_L on top of the model f_E. With f_E fixed, we train the feed-forward layer f_L using the labeled training set of base classes C_{train} by the supervised pretraining approach (Section 3.1). Finally, we use $F = \{f_E, f_L\}$ as our final model to map the raw audio signals $\mathbf{x}$ into the latent embeddings $F(\mathbf{x})$ for few-shot evaluation, following the procedure in (2).

4. EXPERIMENTAL SETUP

We use Audioset [15] comprising ∼2M 10-sec audio clips for our experiments. To evaluate the proposed unsupervised/semi-supervised approaches, we utilize part of Audioset as unlabeled and part as labeled data. In order to have a direct comparison, the labeled portion of our data is exactly the same as [2]. Briefly, the labeled data consists of 19.8k 10-sec audio clips split into train, valid and test partitions (referred to as 'Original' split in Table 2) with 99/21/22 non-overlapping classes respectively. The unlabeled portion of our data consists of randomly selected 1.6M audio clips from Audioset without any overlaps with labeled portion. Following the procedure in Section 3, we experiment on 5-way 1-shot setting with K=1/K^-=10, and 5-way 5-shot setting with K=5/K^-=50. For both settings, the query set is composed of 15 positive samples and 150 negative samples for each novel class. We sample 200 validation tasks and 200 test tasks. We measure the Area Under Curve (AUC) for Receiver Operating Characteristic (ROC), and report mean AUC over 200 few-shot test tasks along with 95% confidence interval. Higher AUC is better.

All experiments are conducted using Tesla V100 GPUs in Pytorch building upon Fairseq [16]. For the vq-wav2vec model, we use a codebook with 2 groups and 320 codewords per group, resulting in 640 logits. After training, we notice that 20K unique tokens are utilized out of a total of 100K codeword combinations. Tokenized audio clips are downsampled by a factor of 2, leading to 500 tokens per sequence to speedup BERT model training. For the BERT training, we use RoBERTa [12] base model. The input sequence size is set as 512. We train the model using 8 GPUs with batch size of 16 per GPU. The gradient is accumulated once every 16 batches resulting in an effective batch size of 2048. For the semi-supervised BERT model, we use a feed forward layer f_L to map the concatenated BERT embeddings from layers 7,8 and 9 (tuned using validation data) from size 2304 to 512. The feed forward layer is trained using Adam optimizer with learning rate 0.001 and batch size 32. We perform early stopping based on AUC of validation set. The early stopping usually happens at around 5 epochs.

5. RESULTS

5.1. Main results

Table 1(a) shows the performance of proposed (unsupervised and semi-supervised) and baseline (supervised) methods on test set. The baseline numbers are replicated from [2]. We notice that unsupervised vq-wav2vec+BERT outperforms supervised pre-training by 0.9% and 1.8% in 1-shot and 5-shot settings. Further, the proposed semi-supervised vq-wav2vec+BERT approach outperforms the supervised prototypical network approach by 1.3% and 0.9% for 1-shot and 5-shot settings respectively. This shows that the model learned in semi-supervised fashion can generalize well to unseen classes. It is interesting to note that the feed forward layer used in semi-supervised vq-wav2vec+BERT has no

Table 1. **Few-shot evaluation results.** We report average AUC (%) with 95% confidence interval. Higher is better.

(a) Original.

Model	1-shot	5-shot
Supervised		
Pre-training (spectrogram) [2]	77.0	85.8
ProtoNet (spectrogram) [2]	83.3	91.2
Pre-training (waveform)	74.1	83.5
ProtoNet (waveform)	72.3	82.1
Unsupervised		
Wav2vec	74.7 ± 0.764	83.2 ± 0.454
Vq-wav2vec	69.6 ± 0.771	78.6 ± 0.531
+BERT	$\mathbf{77.9 \pm 0.760}$	$\mathbf{87.6 \pm 0.427}$
Semi-supervised		
Wav2vec	80.7 ± 0.742	88.2 ± 0.368
Vq-wav2vec	76.8 ± 0.729	85.0 ± 0.422
+BERT	$\mathbf{84.6 \pm 0.736}$	$\mathbf{92.1 \pm 0.311}$

(b) Domain mismatch.

Model	1-shot		5-shot	
	Music	Animal	Music	Animal
Supervised				
Pre-training [2]	70.9	61.3	79.8	69.5
ProtoNet [2]	71.2	64.4	82.4	74.9
Unsupervised				
Wav2vec	66.7 ± 0.869	62.3 ± 0.901	76.0 ± 0.656	70.4 ± 0.686
Vq-wav2vec	63.3 ± 0.871	59.5 ± 0.808	70.5 ± 0.691	66.5 ± 0.635
+BERT	$\mathbf{73.2 \pm 0.918}$	$\mathbf{65.9 \pm 0.887}$	$\mathbf{84.3 \pm 0.575}$	$\mathbf{75.7 \pm 0.669}$
Semi-supervised				
Wav2vec	70.4 ± 0.918	65.1 ± 0.892	80.7 ± 0.643	74.4 ± 0.652
Vq-wav2vec	66.3 ± 0.908	62.3 ± 0.872	75.8 ± 0.658	70.3 ± 0.695
+BERT	$\mathbf{73.7 \pm 0.953}$	$\mathbf{68.5 \pm 0.883}$	$\mathbf{85.3 \pm 0.588}$	$\mathbf{78.8 \pm 0.629}$

non-linearity i.e., the representation space of semi-supervised vq-wav2vec+BERT is a simple linear transformation of representation space learned in unsupervised vq-wav2vec+BERT. This suggests that representations learned in unsupervised fashion are already good enough for disentangling unseen audio events.

We note that our models use waveforms as inputs and wav2vec architecture, whereas baselines [2] use mel spectrograms as inputs and 4-layer CNN architecture. We have replicated supervised baselines using waveforms as inputs and wav2vec as model architecture, and the results show poorer performance compared to original input and model architecture configuration in Table 1(a). Therefore, we hypothesize that the performance gain of our model is not simply due to the difference of inputs.

We also present the results for wav2vec and vq-wav2vec representations in both unsupervised and semi-supervised settings (Table 1(a)). In both cases, we notice that vector quantization hurts the performance of wav2vec. However, it provides a way to employ BERT masked language model training which increases the overall performance considerably.

5.2. Domain mismatch

We evaluate the robustness of our model if there is a domain mismatch occurs at test time. Note that few-shot setting does not assume the knowledge of test classes at training time and thus one cannot preclude the possibility of a novel test class being out of domain. Here, we compare the proposed method to baselines in such a potential domain mismatch scenario. We experiment with music and animal as target domains. With music as an example, we remove events related to music from the original labeled train and valid sets, and use all the events related to music as test set (Table 2).

The performance of all methods drop because of the mismatch (Table 1(b)), however, we notice that the proposed methods are more robust compared to baselines. Our unsupervised vq-wav2vec+BERT model outperforms the prototypical network by 2.0% and 1.9% for 1-shot and 5-shot settings on music setting, and 1.5% and 0.8% on animal setting. We hypothesize that the unsupervised model performs better under domain mismatch settings because it is trained with a larger set of unlabeled data instead of a limited set of labeled data in the source domain. Specifically, the unsupervised model performs better for the music setting. A possible reason is that the unlabeled training set contains ≈1M audio samples related to music while only 40K are related to animal, so the model learns better representations for the music domain.

Table 2. **Dataset splits.**

	Original	Music	Animal
Train	99	77	87
Valid	21	16	19
Test	22	32	15

The semi-supervised vq-wav2vec+BERT model outperforms the prototypical network by 2.5% and 2.9% respectively for 1-shot and 5-shot settings on music setting, and 4.1% and 3.9% on animal setting. Comparing the unsupervised and semi-supervised models, we observe a 0.5% and 1.0% improvement for 1-shot and 5-shot settings on music setting, and a 2.6% and 3.1% improvement on animal setting. We notice that the semi-supervised model does better on animal setting, which means the performance gain brought by the labeled training set is larger for the animal setting. We hypothesize that this could be related to the difference in the size of labeled training set (Table 2) used on music (77 classes) and animal settings (87 classes).

6. CONCLUSION

We studied unsupervised and semi-supervised approaches for few-shot AEC. Building upon the recent advances in sequence modeling for speech and language, we find that representations learned in semi-supervised fashion outperform the existing meta-learning approaches when used for few-shot AEC, and the performance gain is larger under domain mismatch.

7. REFERENCES

[1] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron Weiss, and Kevin Wilson, "Cnn architectures for large-scale audio classification," in *ICASSP*, 2017.

[2] Bowen Shi, Ming Sun, K. C. Puvvada, Chieh-Chi Kao, Spyridon Matsoukas, and Chao Wang, "Few-shot acoustic event detection via meta learning," in *ICASSP*, 2020.

[3] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra, "Matching networks for one shot learning," in *NIPS*, 2016.

[4] Jake Snell, Kevin Swersky, and Richard Zemel, "Prototypical networks for few-shot learning," in *NIPS*, 2017.

[5] Chelsea Finn, Pieter Abbeel, and Sergey Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *ICML*, 2017.

[6] Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele, "Learning to self-train for semi-supervised few-shot classification," in *NIPS*, 2019.

[7] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, B. Joshua Tenenbaum, Hugo Larochelle, and S. Richard Zemel, "Meta-learning for semi-supervised few-shot classification," in *ICLR*, 2018.

[8] Spyros Gidaris, Andrei Bursuc, Nikos Komodakis, Patrick Perez, and Matthieu Cord, "Boosting few-shot visual learning with self-supervision," in *ICCV*, 2019.

[9] Aäron van den Oord, Yazhe Li, and Oriol Vinyals, "Representation learning with contrastive predictive coding," in *Arxiv preprint*, 2018.

[10] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017.

[11] Alexei Baevski, Steffen Schneider, and Michael Auli, "vq-wav2vec: Self-supervised learning of discrete speech representations," in *ICLR*, 2020.

[12] 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," in *ArXiv preprint*, 2019.

[13] Eric Jang, Shixiang Gu, and Ben Poole, "Categorical reparameterization with gumbel-softmax," in *ICLR*, 2017.

[14] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," in *ArXiv preprint*, 2018.

[15] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter, "Audio set: An ontology and human-labeled dataset for audio events," in *ICASSP*, 2017.

[16] Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli, "fairseq: A fast, extensible toolkit for sequence modeling," in *NAACL Demonstrations*, 2019.