IMPROVED REPRESENTATION LEARNING FOR ACOUSTIC EVENT CLASSIFICATION USING TREE-STRUCTURED ONTOLOGY

Arman Zharmagambetov[1]*, *Qingming Tang*[2], *Chieh-Chi Kao*[2], *Qin Zhang*[2], *Ming Sun*[2], *Viktor Rozgic*[2], *Jasha Droppo*[2], *Chao Wang*[2]

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

ABSTRACT

Acoustic events have a hierarchical structure analogous to a tree (or a directed acyclic graph). In this work, we propose a structure-aware semi-supervised learning framework for acoustic event classification (AEC). Our hypothesis is that the audio label structure contains useful information that is not available in audios and plain tags. We show that by organizing audio representations with a human-curated tree ontology, we can improve the quality of the learned audio representations for downstream AEC tasks. We use consistency training to use large amounts of unlabeled data for structured representation manifold learning. Experimental results indicate that our framework learns high quality representations which enable us to achieve comparable performance in discriminative tasks as fully supervised baselines. Moreover, our framework can better handle audios with unseen tags by confidently assigning a super-category (internal node like "animal" in Fig. 1) tag to the audio.

Index Terms— Acoustic event classification, Representation learning, Audio ontology, Decision tree

1. INTRODUCTION

Acoustic Event Classification (AEC) focuses on detecting if certain events exist in a short sound snippet. It is relevant to many applications including home security [1] and smart homes [2] and is becoming increasingly more important with the expansion of virtual assistant technologies such as Amazon Alexa, Google Assistant, Apple Siri, and etc. State-of-the-art AEC models are based on deep neural architectures trained with large amounts of labeled data [3]. Such a model development paradigm is not ideal because it is difficult and expensive to collect an audio corpus for AEC with high diversities in event granularity and acoustic environment. Additionally, it is unfriendly to new acoustic event discovery or feature expansion.

Recent advances in *self-supervised* learning for speech and audio processing [4, 5, 6, 7] have been proven useful in the settings with limited annotated data. In particular, representation networks pre-trained on a large amount of unlabeled data can learn general representations that can be used by a wide range of downstream tasks. Task-specific supervised fine-tuning based on the pre-trained representation networks can typically achieve similar levels of performance with much fewer labeled data. Besides these works which learn a sequence of representations for audios, there is another line of research which focus on learning per-instance fixed-dimensional representations that can benefit downstream tagging tasks [8, 9, 10].

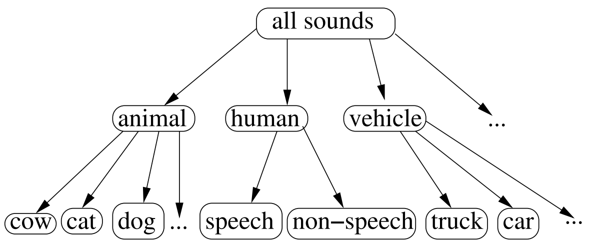

Fig. 1. Example of the tree-structured ontology of various audio events. Hierarchical structure allows one to group similar audio events under the same "super-category" (i.e. "animal").

All of the aforementioned self-supervised learning approaches ignore the domain knowledge, which contains rich information that can be used to improve model performance. Motivated by this, we capitalize on the idea of using a tree-structured ontology to guide the training of the representation network, as there exist numerous human-curated ontologies (e.g. [11]) for acoustic event classification. Tree-based ontology has a special hierarchical structure that merges similar audio events under the same subtree (see Fig. 1) while keeping dissimilar sounds distant in the hierarchy. Encouraging the audio representation manifold to align with a label ontology is beneficial for organizing the audio representations in a more reasonable fashion. For example, as shown in Fig. 1, the label ontology does not only assist in distinguishing between leaf nodes like "cow" and "cat", it also ensures that "cow" and "cat" fall into a loose super-category that is "far away" from "human" and "vehicle" sounds. We expect such hierarchically organized audio embedding spaces can further benefit the performance of downstream tagging tasks, and enhance the generalization to novel or rare events as well.

In this paper, we develop a 2-module joint model consisting of a representation neural network and a decision tree based on a pre-defined tree-structured ontology. We train the model using a large set of unlabeled data and a limited amount of labeled data. We utilize the large amount of unlabeled data by data augmentation and consistency training [12] . More specifically, we encourage the different augmented views of the same audio to predict the same root-to-leaf path on the ontology tree. We show that tree-based modeling not

*Work performed during a Summer Internship at Amazon.

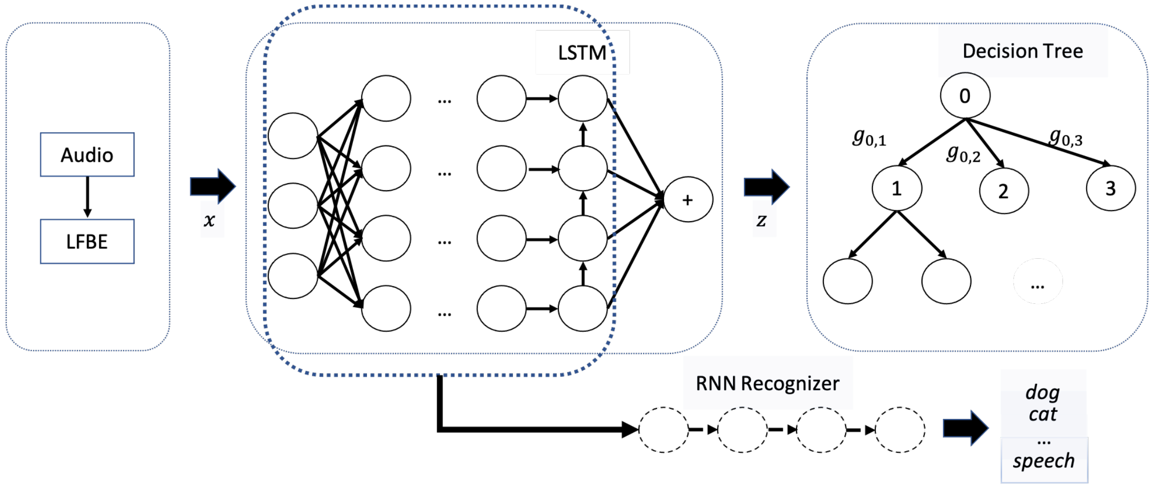

Fig. 2. Representation network (CNN-LSTM, CRNN for short, see Section 4.1 for details) takes log mel-filterbank energy (LFBE) $\mathbf{x}$ as input and produces an embedding vector $\mathbf{z}$ (average of the LSTM outputs). Decision tree operates on embedding space and has its own learnable parameters at each node. The entire representation architecture is trained end-to-end. When training the one-layer RNN AEC recognizer on top of the CRNN, we detach the pooling layer and freeze the CRNN encoder.

only significantly improves the model performance on events seen during the training stage, but also helps the learned representations generalize to super-categories that are semantically correct. Such a capability is beneficial for discovering novel sound types while maintaining good performance on seen events.

2. RELATED WORKS

Using label structure to improve the model performance is widely used in many domains, including unsupervised speech recognition [13, 14] and semantic segmentation [15]. Specific to the AEC domain, [16] shows that textual embedding contains rich structure information that can be used to do zero-shot learning for AEC. Label-aware AEC also draws attention recently in fully-supervised scenarios. For instance, [17] uses a 2-level ontology of audio classes for AEC and [18] uses graph convolutional networks to learn label embeddings. Different from the aforementioned works, our work combines semi-supervised representation learning and tree-based modeling. Our motivation is to better organize the audio representations by leveraging human knowledge, which is under-exploited. We find that the consistency training significantly solves the lack-of-data problem of AEC. More importantly, the method we propose shows potential in discovering novel acoustic events.

3. TREE-STRUCTURED ONTOLOGY FOR REPRESENTATION LEARNING

The model architecture we propose is presented in Fig.2. We first extract spectrogram features (input $\mathbf{x}$) and pass them to a representation network (encoder). The encoder we used is a shallow CNN-LSTM (CRNN for short) model which features one LSTM layer on top of a CNN module. We do not experiment with other model architectures in this paper as it is not the focus of this work. The transformation of the encoder is denoted as $\mathbf{z} = f(\mathbf{x}; \Theta)$ where Θ represents the encoder (e.g. CRNN) parameters, and $\mathbf{z} \in \mathcal{R}^{128}$. The representation vector $\mathbf{z}$ is the input to the tree module. For each internal node i we define a gating function $g_i(\cdot)$ in Eq. 1 with parameters $\mathbf{W}_i \in \mathcal{R}^{128 \times C_i}$ and $\mathbf{b}_i \in \mathcal{R}^{C_i}$, where $C_i \geq 2$ is the number of children of node i.

$$g_i(\mathbf{z}; \mathbf{W}_i, \mathbf{b}_i) = \sigma(\mathbf{W}_i^T \mathbf{z} + \mathbf{b}_i) \tag{1}$$

The above softmax gating function outputs probability distribution over the children nodes of i. In our implementation, a probability distribution that traverses from root to a leaf node simply factorizes as:

$$P(y|\mathbf{z}) = g_{\mathbf{c}_0,\mathbf{c}_1}(\mathbf{z}) \cdot g_{\mathbf{c}_1,\mathbf{c}_2}(\mathbf{z}) \dots g_{\mathbf{c}_{l-1},\mathbf{c}_l}(\mathbf{z}) \tag{2}$$

where $\mathbf{C} = \{\mathbf{c}_0 = \text{root}, \mathbf{c}_1, \mathbf{c}_2, \dots, \mathbf{c}_l = y\}$ is the set of nodes along the root-to-leaf path, and $g_{\mathbf{c}_{i-1},\mathbf{c}_i}$ indicates the probability of transition from node $\mathbf{c}_{i-1}$ to its child $\mathbf{c}_i$. We denote $\mathbf{W} = \{\mathbf{W}_i, \mathbf{b}_i\}_{i=1}^{K}$, where K is the number of non-leaf nodes. Note that the described implementation can also handle directed acyclic graphs if we enumerate all the root-to-leaf paths. As an initial exploration, we simplify our setting to tree structure and only handle one root-to-leaf path.

The tree ontology we use (see Fig. 1 for illustration, and Section 4.1 for setup) originates from [11], and it covers all of the events (leaf nodes) that appear in the training data. Our tree-based training does not only encourage $\mathbf{z}$ to be classified as a concrete event (e.g "dog barking"), but also organizes $\mathbf{z}$ in a hierarchical fashion so it can also be semantically closer to other "animal" sounds.

3.1. Tree-based optimization

Given a training set $(X, Y) = \{(\mathbf{x}_n, y_n)\}_{n=1}^{N}$ consisting of input $\mathbf{x}$ and the corresponding list of ground truth labels y, we minimize the negative log-likelihood loss shown below:

$$L_s(\mathbf{W}, \Theta) = - \mathop{\mathbb{E}}_{\mathbf{x},y} \frac{\sum_{j=1}^{|y|} \log P(y_j | f(\mathbf{x}))}{|y|} \tag{3}$$

where y_j indicates the j^{th} label of the sample $\mathbf{x}$ if it has multiple labels. Here, one label corresponds to one leaf node in the pre-defined tree ontology. We implement a stochastic approximation to Eq. 3, where we randomly select one of the ground truth labels of sample x, and treat x as single-event audio during each update.

Please note that, given the pre-defined ontology, our tree-based modeling is significantly simplified compared to Soft Decision Trees (SDT) or related methods that combine trees and neural networks [19, 20]. For example, vanilla SDT training involves weighted averaging over all leaves, whereas here we maximize the probability of the correct root-to-leaf path only. However, we refer to our tree-based optimization as *SDT* in this paper for simplicity.

3.2. Consistency training for unlabeled data

Eq. 3 is not directly applicable for unlabeled data. In order to use unlabeled data, we apply *consistency training* [12], one semi-supervised learning technique. Recent works have shown that data augmentation plays a critical role in semi-supervised learning [21]. We use common augmentation methods, such as SpecAugment [22], to apply transformations on the LFBE surface feature. In a nutshell, assuming we are given a portion of data indexed in set $\{\mathbf{x}_m\}_{m=1}^{M}$, the unsupervised consistency loss is defined as:

$$L_c(\mathbf{W}, \Theta) = \mathop{\mathbb{E}}_{\mathbf{x}} \left\{ \mathop{\mathbb{E}}_{\mathbf{c} \in \mathbf{C}} \{\mathcal{D}(g_c(\mathbf{z}), g_c(\hat{\mathbf{z}}))\} \right\} \tag{4}$$

where $\mathbf{C} = \{\mathbf{c}_0 = \text{root}, \mathbf{c}_1, \mathbf{c}_2, \ldots, \mathbf{c}_l\}$ is the set of internal nodes along the most likely root-to-leaf path of the non-corrupted sample $\mathbf{x}$ (see more context from Eq.2 and Eq.1). $\mathbf{z}$ and $\hat{\mathbf{z}}$ are the representations of $\mathbf{x}$ and $\hat{\mathbf{x}}$ respectively. We use cross-entropy in our implementation, but $\mathcal{D}$ can be other losses like KL divergence.

For input perturbations (e.g. $\hat{\mathbf{x}}$), we use four types of augmentation that are applied sequentially: 1) adding Gaussian noise (max standard deviation $= 0.05$), 2) audio time shift, 3) frequency mask (with max frequency=10 applied at random location) and 4) time mask (two masks with time frame=50 applied at random locations). The final loss is given by combining Eq. 3 and Eq. 4 with a weighting hyper-parameter λ:

$$L(\mathbf{W}, \Theta) = L_s(\mathbf{W}, \Theta) + \lambda L_c(\mathbf{W}, \Theta) \tag{5}$$

Here, L_c considers both labeled and unlabeled data, whereas L_s applies to labeled data only. We use $\lambda = 1$ in all our experiments and we denote this modified version of SDT as *SDTC*.

4. EXPERIMENTS

4.1. Data and setup

We use AudioSet [11] which has its own human-curated ontology for our experiments. AudioSet contains $5.8k$ hours of audios from 527 sound classes. We only select audio events with annotation accuracy more than 80% to ensure label quality. Moreover, to simplify the ontology into the tree structure, we remove the nodes with several parents. These operations lead to 127 "leaf events" and 79 "super-categories" (internal nodes) after filtering (206 events in total). We further divide the leaf events into 110 "seen" events and keep the remaining 17 as "unseen" events (i.e. events will only be used in evaluation, see section 4.3). Overall, we have $444,752$ audio clips belonging to "seen events" with the following partition: 70% for train, 15% for dev and 15% for test. We manually pick the following events as *unseen*: "Purr", "Cluck", "Conversation", "Narration, monologue", "Baby cry, infant cry", "Child singing", "Pant", "Strum", "Hammond organ", "Double bass", "Tabla", "Clarinet", "Stream", "Skidding", "Subway, metro, underground", "Buzzer" and "Cash register". The remaining 110 "seen" classes can be obtained after the pre-processing steps described above.

Table 1. Comparison of different methods on a downstream classification task (on a subset of AudioSet) given fixed representations. F1 score is the average of 5 runs.

Method	Label fraction 1%	5%	10%
	Test F1 macro		
supervised	0.612		
supervised (with aug.)	0.638		
LFBE	0.201	0.442	0.508
SimCLR+APC	0.244	0.491	0.551
SimCLR+APC → Fine-tune	0.329	0.540	0.587
SDT	0.288	0.531	0.580
SDT + consistency (SDTC)	0.410	0.557	0.600
SDTC + APC (SDTCA)	**0.417**	**0.561**	**0.609**

We extract the log mel-filterbank energy (LFBE) features for each 10-seconds audio clip (sample rate $= 16k$): we use a window size of 25 ms, a hop size of 10 ms, and the number of mel coefficients is 64. This effectively gives us audio features of dimensions 998×64. We do not apply global mean and variance normalization. As for the encoder architecture, we use the same CRNN-based network for all methods in this paper. The architecture consists of a 3-layer $2D$ CNN with number of filters 24,48 and 96, respectively. Each CNN kernel is followed by ReLU, Max Pooling (except the last CNN layer) and Batch Norm. The 2D kernel and stride are designed such that the CNN module outputs $30 \times 96 \times 2$ features which is further reshaped to 30×192, and then fed to LSTM layer of hidden size 128. The LSTM output (30×128) is used for RNN classifier training, and the average of the LSTM output is the audio representation used as the input to the tree module. For all representation learning approaches described in the next sections, we apply the following optimization configurations: we use Adam optimizer with initial learning rate 0.001, norm gradient clipping 1.0 and batch size 256 (1024 for contrastive learning). We train all models for 10 epochs and use the model at the epoch with the lowest validation loss for downstream tasks.

4.2. Main Results: Audio Event Classification (AEC)

We compare CRNN encoders pre-trained by different methods on AEC. We pick a subset of 20 classes from seen events and train a single layer LSTM classifier on top of the pre-trained CRNN encoder. When training all the classifiers (except fully supervised baselines using 100% of the training data) in Table 1, we ***freeze the CRNN encoder***. We leverage only a small subset of labeled data (e.g. 1%, 5% and 10%) to train the classifier. The remaining 90% of the data serve as unlabeled data during the representation learning phase. We compare our **SDT** representation learning method with "APC" [4] and "SimCLR [8]". To make a fair comparison, all encoders use *the same shallow CRNN*. APC adopts language model style pre-training where future frames are generated based on past frames. We adapt APC a bit such that the summary of each convolutional window is used to predict all the frames that will appear in the next convolutional window. The choices of the CRNN architecture and the adapted APC loss are based on the consideration that that models with smaller footprint are more friendly to modern edge devices. SimCLR is another self-supervised baseline where the main idea is to maximize the similarity between representations obtained from the same image. Our ablation study (not shown in the table due to space limit) shows that combining SimCLR and APC losses with weighting factor $\beta = 0.6$, i.e. $\beta \cdot$SimCLR$+(1-\beta)\cdot$APC, yields better performance than both SimCLR and APC, so we only show "SimCLR+APC" in the table.

Table 1 shows the dependency of the F1 macro score on the test set with respect to the percentage of the labeled data used for training. One thing to notice is that all representations learned using the 90% remaining data out-perform simple LFBE features. The fully supervised baseline achieves the best performance, which is consistent with our expectation as it can be considered as a "theoretical" upper bound. Another interesting aspect is that SDT actually learns more discriminative representations than "SimCLR+APC" or the plain LFBE classifier. This strongly suggests that the hierarchical structure information hidden in the tree ontology is beneficial for AEC tasks. Note that, if the classifier is trained with $\alpha\%$ of labeled data, the SDT (and also the "Fine-tune" based on $128D$ vector for SimCLR+APC) is also trained using this $\alpha\%$ of data. The benefit of SDT over SimCLR+APC diminishes if the encoder of SimCLR+APC is also fine-tuned (without using tree ontology).

Finally, by using consistency training, SDT training can also utilize the remaining 90% of the unlabeled data in a weakly supervised fashion by matching the predicted paths of different augmented copies. We can see the clear benefit of **SDTC** over all other methods in Table 1. Interestingly, we even observe that SDTC approaches the fully supervised baseline (without data augmentation). We further add APC loss to Eq. 5 (**SDTCA** in the table) as APC loss learns temporal patterns that could be missed by SDT and consistency losses. Keep in mind that the APC is only used as a multitask loss rather than for pre-training purposes here. We observe minor benefits by adding this loss.

4.3. Accuracy at the level of super-categories

Finally, we evaluate the representation quality of SDTCA (trained using 10% labeled data) at the level of super-categories using the $128D$ representations. To elaborate more, assuming that we have an audio clip of "dog barking", then, by using the tree ontology, we can infer the ancestors of this clip: "dog" $\rightarrow$ "domestic animal" $\rightarrow$ "animal". Following the same logic, we can determine if this clip is correctly classified as dog, domestic animal, etc. This is especially important for unseen events (see definition in section 4.1) since they do not appear as leaves in the ontology thus softmax functions do not consider these events during training. For comparison, we use the prediction provided by "SimCLR+APC $\rightarrow$ Fine-tune". A fine-tuned model uses the softmax output to make a prediction on the known label dictionary, and we can match that prediction with our ontology to find all ancestors.

Table 2. Avg accuracy by levels where "Level 1" checks if a leaf node is correctly classified, parent and grandparent for "Level 2" and "Level 3". Note that, the setting is different from Table 1 as we use representation vector $\mathbf{z}$ to make predictions.

	Seen		Unseen	
Avg. acc.	SDT	Baseline	SDT	Baseline
Level 1	0.46	0.35	-	-
Level 2	0.61	0.44	0.59	0.41
Level 3	0.79	0.62	0.78	0.57

Table 2 summarizes our results. We measure the accuracy as follows: following the example of "dog barking", we collect all audio clips which belong to this event. We measure how many of them are correctly classified as "dog" (Level 1), as "domestic animal"(Level 2), etc. We measure the performance on both seen (same 20 events used in Section 4.2) and unseen events. According to the results, our method significantly outperforms the baseline in Level 1, and does a better job in identifying the super-categories. This illustrates that the trained embeddings respect the structure we learned from ontologies. We also find that, SDTCA yields a larger improvement than non-SDT approaches on Level 2 and 3 for unseen events, compared to the seen event scenario, which supports the claim that our proposed method has the potential to benefit novel event discovery.

5. CONCLUSION

We leverage a tree-structured ontology of audio events for representation learning for acoustic event classification. Since it is not trivial to embed such information into the learning pipeline, we propose a parametric tree model which can be jointly trained with a representation encoder. Moreover, we apply a semi-supervised learning scheme based on consistency training that can be used to handle label scarcity. To the best of our knowledge, this is the first attempt to use consistency training for tree-based models in the AEC domain. Experimental results suggest that SDT-based semi-superivsed learning can convey the structural information hidden in the label ontology to learned audio embeddings, and thus further improve AEC performance. Also, such kinds of learning schema shows its potential in more confidently classifying unseen events to their correct super-categories.

6. REFERENCES

[1] Danish Chowdhry, Raman Paranjape, and Paul Laforge, "Smart home automation system for intrusion detection," in *2015 IEEE 14th Canadian Workshop on Information Theory (CWIT)*. IEEE, 2015, pp. 75–78.

[2] Andrey Temko, Robert G. Malkin, C. Zieger, Dusan Macho, and C. Nadeu, "Acoustic event detection and classification in smart-room environments: Evaluation of CHIL project systems," in *The IV Biennial Workshop on Speech Technology*, 2006.

[3] 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 *Proc. ICASSP*, 2017, pp. 131–135.

[4] Yu-An Chung, Wei-Ning Hsu, Hao Tang, and James Glass, "An unsupervised autoregressive model for speech representation learning," in *Proc. Interspeech*, 2019, pp. 146–150.

[5] Yu-An Chung, Hao Tang, and James Glass, "Vector-quantized autoregressive predictive coding," in *Proc. Interspeech*, 2020, pp. 3760–3764.

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

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," in *Advances in Neural Information Processing Systems (NeurIPS'20)*, 2020.

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton, "A simple framework for contrastive learning of visual representations," in *International Conference on Machine Learning (ICML'20)*, 2020, pp. 1597–1607.

[9] Aaqib Saeed, David Grangier, and Neil Zeghidour, "Contrastive learning of general-purpose audio representations," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3875–3879, 2021.

[10] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *ArXiv*, vol. abs/2006.09882, 2020.

[11] 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 *Proc. ICASSP*, 2017.

[12] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le, "Unsupervised data augmentation for consistency training," in *Advances in Neural Information Processing Systems (NeurIPS'20)*, 2020, vol. 33, pp. 6256–6268.

[13] Chih-Kuan Yeh, Jianshu Chen, Chengzhu Yu, and Dong Yu, "Unsupervised speech recognition via segmental empirical output distribution matching," *ArXiv*, vol. abs/1812.09323, 2019.

[14] Alexei Baevski, Wei-Ning Hsu, Alexis Conneau, and Michael Auli, "Unsupervised speech recognition," arXiv:2105.11084, May 24 2021.

[15] Mohammadreza Mostajabi, Michael Maire, and Gregory Shakhnarovich, "Regularizing deep networks by modeling and predicting label structure," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5629–5638, 2018.

[16] Huang Xie and Tuomas Virtanen, "Zero-shot audio classification based on class label embeddings," *2019 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pp. 264–267, 2019.

[17] Abelino Jimenez, Benjamin Elizalde, and Bhiksha Raj, "Sound event classification using ontology-based neural networks," in *NeurIPS Workshop on Interpretability, Robustness in Audio, Speech and Language (IRASL)*, 2018.

[18] Yiwei Sun and Shabnam Ghaffarzadegan, "An ontology-aware framework for audio event classification," in *Proc. ICASSP*, 2020, pp. 321–325.

[19] Peter Kontschieder, Madalina Fiterau, Antonio Criminisi, and Samuel Rota Buló, "Deep neural decision forests," in *Proc. 15th Int. Conf. Computer Vision (ICCV'15)*, Santiago, Chile, Dec. 11–18 2015, pp. 1467–1475.

[20] Arman Zharmagambetov and Miguel Á. Carreira-Perpiñán, "Learning a tree of neural nets," in *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 3140–3144.

[21] Kihyuk Sohn, David Berthelot, Chun-Liang Li, Zizhao Zhang, Nicholas Carlini, Ekin Dogus Cubuk, Alexey Kurakin, Han Zhang, and Colin Raffel, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *ArXiv*, vol. abs/2001.07685, 2020.

[22] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019, pp. 2613–2617.