ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) | 978-1-7281-6327-7/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICASSP49357.2023.10097025

AN APPROACH TO ONTOLOGICAL LEARNING FROM WEAK LABELS

Ankit Shah ⋆ *, Larry Tang* †*, Po Hao Chou* † *,Yi Yu Zheng* †*, Ziqian Ge* †*, Bhiksha Raj* ⋆ ‡

⋆ Language Technologies, Carnegie Mellon University,
† ECE Department, Carnegie Mellon University,
‡ Mohammed bin Zayed University of Artificial Intelligence

ABSTRACT

Ontologies encompass a formal representation of knowledge through the definition of concepts or properties of a domain, and the relationships between those concepts. In this work, we seek to investigate whether using this ontological information will improve learning from weakly labeled data, which are easier to collect since it requires only the presence or absence of an event to be known. We use the AudioSet ontology and dataset, which contains audio clips weakly labeled with the ontology concepts and the ontology providing the "Is A" relations between the concepts. We first re-implemented the model proposed by [1] with modifications to fit the multi-label scenario and then expand on that idea by using a Graph Convolutional Network (GCN) to model the ontology information to learn the concepts. We find that the baseline Twin Neural Network (TNN) does not perform better by incorporating ontology information in the weak and multi-label scenario, but that the GCN does capture the ontology knowledge better for weak, multi-labeled data. We also investigate how different modules can tolerate noises introduced from weak labels and better incorporate ontology information. Our best TNN-GCN model achieves mAP=0.45 and AUC=0.87 for lower-level concepts and mAP=0.72 and AUC=0.86 for higher-level concepts, which is an improvement over the baseline TNN but about the same as our models that do not use ontology information.

Index Terms— Ontology, Graph Convolutional Network (GCN), Twin Neural Network (TNN), weakly labeled data

1. INTRODUCTION

Ontologies represent hierarchical concepts through categories and relationships of domain knowledge. For example when we hear a sound, even if we don't recognize the specific animal species making the sound, we still recognize that given sound is a type of animal sound. Like humans, a machine can utilize ontology information to help to classify an object it hasn't seen before as a higher-level label. Incorporating these hierarchical relations from ontologies can improve the classification of semantically different observations that appear similar or provide more general descriptors of ambiguous subclasses. Many recent works have explored using this external knowledge to extract better feature representations [2, 3] as well as how to embed the knowledge into model architectures [4, 5, 6, 7].

In this paper, we investigate whether the learning of ontological classes from weak labeled data can be improved by including external knowledge of these relationships. For example, an audio clip may contain a *dog howling*, a *baby crying*, and an *engine idling*, but is weakly labeled simply as a *dog howling* is present. We will predict the ontological information of that data point, which includes *animal sounds* and *living things* as the higher level concepts. The ontology knowledge base can be also be provided or incorporated into the neural network to aid in the classification task. One of the challenges in this paper is that the network should also be able to notice that there may be parts of the data that could belong to other ontological labels, or could be tagged in the same higher level ontological label but may not be labeled as such. For example, a piece of an audio clip containing both *dog howling* and *cat meowing* may only be weakly labeled as *dog howling*, while both *dog howling* and *cat meowing* could lead to the prediction of the higher level ontological label *animal sounds* or *living thing*. We would like to look into whether the network can use that hidden knowledge to improve the prediction for respective ontological categories.

The advantage of weakly labeled datasets is that weak labels are easier to collect since they require only the presence or absence of an event to be known, which leads to better scalability as demonstrated by Google's AudioSet [8]. However, they may introduce noise into the labels which makes it difficult for models to learn from. Investigating this problem is thus important because it may lead to improved performance for classification tasks in which it is difficult to obtain strongly labeled datasets. We look to improve upon some previous work in the space of ontology prediction, hierarchical learning, and knowledge graphs by implementing two different models which embed the ontology information to predict the ontology classes and hierarchies from weak labels. We hope to gain a better understanding of ontological embeddings, whether they can improve classification in the weak label scenario, and the advantages or disadvantages of using weak labels. The following sections give a more detailed background of recent literature and then discusses the techniques and models we implement and our final results and conclusions.

2. METHODS

2.1. Dataset and Ontology

We use the AudioSet dataset, which consists of an ontology of weakly labeled 632 audio event classes and 2,084,320 human-labeled 10-second sound clips drawn from YouTube videos. The 128-dimensional Audioset features and ontology files are made available by Google and accessed online from previous work [9], [8]. Feature embeddings provided by [8] are extracted from a VGGish model [10] followed by a PCA to reduce down to 128 dimensions and each ten second clip is represented by ten 128-dimension feature vectors, i.e. one feature vector represents one second of one clip. Each clip in the dataset can contain mutiple labels where the clsses encompass a wide variety of sounds like human noises, music, animals or vehicles.

We use balanced training set (about 22,160 10-sec clips) of Audioset to train our models. We then preprocess the corresponding labels of each audio clip to get the labels in the top two levels of the Audioset ontology. The validation set is about 20% of the training

set size and is drawn from the unbalanced training set of Audioset, and the Audioset test set contains about 20,383 audio clips. The same label processing is done for the validation and test set to obtain two levels of ontology labels. Although there are more levels to the Audioset ontology, we choose to focus on the top two levels to fit our model architectures.

To give a more general idea of the Audioset data, consider a weakly labeled audio clip which could have multiple labels with no timing information, but the labels could be related due to their ontology. For example, a human voice and hands could possibly co-occur because they are both natural sounds. The labels could also be related due to their dependency, such as a human crying could co-occur with sad music. On the other hand the labels could also not be related with each other. For instance, a dog barking sound and car engine sound could co-occur in a clip accidentally.

2.2. Framework

We consider multi-labeled training data $\{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$ where $\mathbf{x}_i \in \mathcal{X}$ is a single 128-dimension audio feature representation and $\mathcal{X}$ represents training data represented with embeddings and the corresponding $\mathbf{y}_i$ is a collection of labels $\{(y_1^1, y_1^2, ..., y_1^{T_1}), ..., (y_k^1, y_k^2, ..., y_k^{T_k})\}$ with k equal to the number of ontology levels and T_i equal to the number of labels for $\mathbf{x}$ at the i^{th} ontology level. Thus in our post-processed dataset, we have k = 2 and will refer to these as a subclass or "level 1" labels and superclass or "level 2" labels.

2.3. MLP Network without Ontology Information

To investigate whether incorporating the ontology information is beneficial to use with weakly labeled data, we first need to train a model which uses no ontology information. We implement a simple MLP network with three hidden layers and a final output layer that collectively predicts ontology classes. More concretely, the final layer has size $\sum_{i=1}^{k} T_i$ to predict all ontology labels. Each hidden layer is of size 512 followed by a BatchNorm layer, ReLU activation, and a Dropout layer. This model makes no use of the ontology information to aid in the prediction of the ontology labels, making it a good baseline to compare to our models which will incorporate the ontology information. The final outputs pass through a final sigmoid activation for the multi-label scenario and the model aims to minimize the binary cross entropy loss.

2.4. Twin Neural Network (TNN) model with Ontology-based embeddings and Ontological layer

The ontology-based model from [1] is a recent contribution that specifically focuses on audio data while also providing a method for the prediction of classes at different ontology levels. Much of the prior work on ontology prediction has been in different areas such as medicine (drug classification) or text classification, whereas [1] presents a framework that is more relevant and recent. This model will therefore serve as a baseline for the models which will incorporate ontology information. The framework with modification for multi-label scenarios is shown in Figure 1.

Now we present the components of the model and its mathematical formulation, which is an extension of the work in [1] to learn ontology-based embeddings for the classification of multi-labeled data. We use a Twin Neural Network (TNN) to separate the ontology-based embeddings by imposing the embedding distance close to 0 if the input pairs are from the same subclass, close to 1 if they are from different subclasses, but the same superclass, and close to 2 if they are from different superclasses. To fit in the multi-label scenario, we applied two kinds of sampling methods, the first one is that samples fall into the same sub (super) class category if and only if they have exactly the same subclasses; samples fall into different sub (super) class category if any of their sub (super) classes is different. The second approach is that samples fall into the same sub (super) class category if and only if there is any sub (super) class in their intersection; samples fall into different sub (super) class categories if there is no sub (super) class in their intersection.

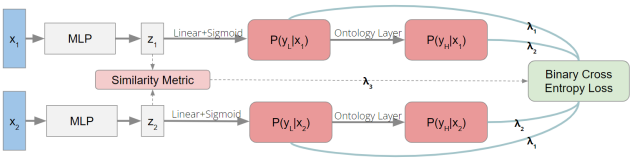

Fig. 1: Architecture of Twin Neural Network + Ontological Layer with modification to fit the multi-label task

The base model architecture of the Twin neural network net is the same MLP with three hidden layers from section 2.3, and each branch of the Twin neural network net shares the MLP parameters. Given a pair of input vectors $\mathbf{x_1}, \mathbf{x_2}$, the output of the MLP is an ontology embedding $\mathbf{z_1} = f(\mathbf{x_1}), \mathbf{z_2} = f(\mathbf{x_2})$. The embedding vectors $\mathbf{z_1}, \mathbf{z_2}$ produce subclass probabilities after a sigmoid activation, which is used for the prediction of multi-labeled data.

The ontological layer, $\mathbf{M}$, then relates the class probabilities of level 1 to those in level 2 of the ontology via the following relation:

$$\mathbf{p}(y_2|\mathbf{x}) = \mathbf{M} \cdot \mathbf{p}(y_1|\mathbf{x}) \tag{1}$$

We modify the ontological layer proposed in [1] in order to fit the multi-label data scenario, where $\mathbf{M}$ is constructed such that it averages the probabilities of the subclasses within a single superclass. Note that this layer is fixed and not trainable, it depends strictly on the ontology of the training data.

The final loss function incorporates the binary cross entropy loss of the level 1 class, $\mathcal{L}_1$, and the level 2 classes, $\mathcal{L}_2$, with the embedding loss, $D_w = (||\mathbf{z_1} - \mathbf{z_2}||_2 - d)^2$ where $d \in \{0, 1, 2\}$ according to the type of input pair.

$$\mathcal{L} = \lambda_1(\mathcal{L}_1^1 + \mathcal{L}_1^2) + \lambda_2(\mathcal{L}_2^1 + \mathcal{L}_2^2) + \lambda_3 D_w \tag{2}$$

The base network takes an input feature of dimension 128 before the output layer to the 42 classes in the first ontology level and then through the ontology layer to the 7 classes in the second level. The baseline paper does not actually report evaluation metrics for the Audioset data, as they focused on other audio datasets: Urban Sounds - US8K and data from the Making Sense of Sounds Challenge. These are both single-labeled datasets.

2.5. Graph Convolutional Network

To model both the co-occurrence information introduced from weak labels and the domain knowledge from ontology, we seek to utilize graph embedding approaches to extend the Twin Neural Network (TNN) architecture. In [11] and [12], a Graph Convolution Network (GCN) is shown to be effective for learning useful node representations through the information in a correlation graph. The essential idea is to update the node representations by aggregating information from neighboring nodes. Following prior work, [11, 12, 13], we

define the subclass labels and superclass labels as the nodes of the graph and aim to learn the label representation. The knowledge in the graph is encoded as a correlation matrix, which is a crucial part of the GCN. We will describe how it is constructed in Section 2.5.1.

We use one-hot encoding of labels as the initial node representation and use 2 GCN layers to extract embeddings with neighboring information. For each GCN layer, we use 2 linear layers to transform the input embedding of the node itself and a graph convolution to aggregate the embeddings from its neighbor. Given a label embedding $Z \in \mathbb{R}^{C \times d}$ (where C is the number of nodes and d is the dimensionality of node features), the graph convolution operation is:

$$Z^{l+1} = h(A' Z^l W_1^l W_2^l) \tag{3}$$

where $W_1^l \in R^{d \times d'}$ and $W2^l \in R^{d' \times d''}$ are 2 transformation matrices to be learned and $A' \in R^{C \times C}$ is the correlation matrix and $h(.)$ is a non-linear operation which is a LeakyReLU in our experiments. The dimensions of each the linear layers are 280 and 512 for the first GCN layer and 320 and 128 for the second GCN layer.

2.5.1. Correlation Matrix

The GCN learns node representations by collecting information from other nodes based on the correlation matrix provided. Thus, how we build the correlation matrix is crucial but also challenging for GCN. In this work, we referred to previous work [11, 12] and experimented on 3 different correlation matrices.

Labels Co-occurrence based Correlation Matrix: [12] proposed a way to model label dependency in the form of conditional probability, i.e. $P(L_j \mid L_i)$ denotes the probability of occurrence of L_j when L_i is present. To construct the correlation matrix, we count the co-occurrence of label pairs present in the training set to get a matrix $M \in R^{C \times C}$, where $M_{i,j}$ denotes the co-occurrence time of L_i and L_j. We then divide M by the occurrence time of L_i in the training set to get the conditional probability P. To prevent a long-tail distribution where some rare co-occurrences may be noisy, we binarize P by setting a tunable threshold t. The correlation matrix A is set to 1 if P are above the t and set to 0 when P is below the t, where $A \in R^{C \times C}$.

Ontology-based Method One: [11] proposed the correlation matrix A to denote label pairs who have same parents. $A_{i,j} = 1$ when L_i and L_j have same parents; $A_{i,j} = 0$ otherwise.

Ontology-based Method Two: [11] proposed the correlation matrix A to denote label pairs that have edges in between them. In the dataset we are using, edges only occur between parents and children, so we set $A_{i,j} = 1$ when L_i is a child of L_j or the other way around; otherwise, $A_{i,j} = 0$.

To prevent over-smoothing, after binarizing, we re-weight A to get the desired correlation matrix A' by setting $A'_{i,j} = p$ when $i = j$ and $A'_{i,j} = (1-p)/\sum_j A_j$ when $A_{i,j} = 1$, where p is probability.

2.6. Twin Neural Network with Graph Convolutional Network

The Graph Convolution Network module described in Section 2.5 can convert the label embedding from a one-hot representation to embedding aggregating neighbor information. We use matrix multiplication to get the similarity between a given audio clip embedding and every label's embedding. Given these similarity values, we then use a SoftMax activation and Binary Cross Entropy loss to calculate the loss between our outputs and the target labels. The overall framework is shown in Figure 2. Here we replace the baseline Ontological Layer with the label embeddings that the GCN generates. To combine it with TNN, we keep the λ_1, λ_2 and λ_3 for the loss term.

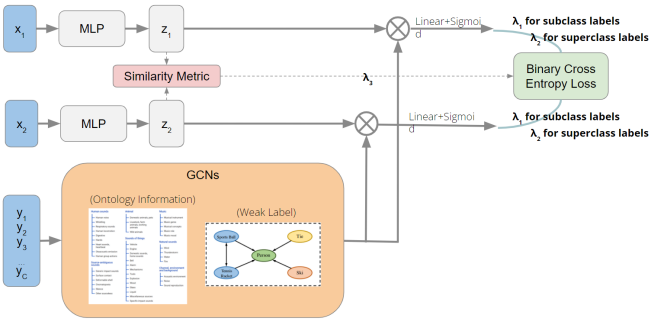

Fig. 2: The framework of Twin Neural Network + GCN

3. EXPERIMENTS

3.1. Evaluation Metrics

The performance of our models is evaluated using weighted average precision and AUC from predictions. All reported metrics are on the test set of the Audioset dataset. As discussed previously, we extract the labels for each segment from the top two levels of the Audioset ontology. The average precision (AP) metric is first computed for each class by considering the precision (fraction of true positive labels out of all predicted positive labels) - recall (fraction of true positive labels out of actual positive labels) curve at different thresholds. Thus it is an indication of how well the model can identify positive classes in the data. The other metric we consider is AUC, which considers negative labels by computing the area under the TPR (true positive rate) - FPR (false positive rate) curve and is an indication of how well the model can distinguish between classes. The AP/AUC metric for the classes on the same ontology level is combined through a weighted average based on the proportion of each class that is present in the training data to get a final weighted AP and weighted AUC score for each level.

3.2. Performance of MLP Model with no Ontology Information

The MLP model with no ontology information achieves a weighted AP/AUC score of 0.45/0.87, respectively for subclasses and a AP/AUC scores for the superclasses is 0.71/0.86, as shown in Table 1. We train the model for 70 epochs with a learning rate of 2e-3.

3.3. Performance of Twin Neural Network with Ontological Layer

The Twin neural network model with the Ontology layer achieves a subclass weighted AP/AUC = 0.36/0.81 and superclass weighted AP/AUC = 0.39/0.65. After hyperparameter tuning of the loss function, we were able to achieve results in Table 1 with $\lambda_1 = 1.5, \lambda_2 = 1, \lambda_3 = 0.25$. We found that these metrics can be tuned based on the λ values in the loss function to control the contributions from each level of the ontology.

Our results shows there should be more research work to apply TNN to a multi-label scenario, such as how to generate pairs of the same sub/superclass as well as their similarity metric. We further investigated this problem by considering two definitions: one in which a pair of data is of the "same" subclass if they have exactly the same labels and another in which "same" means that two pairs just have some intersection in their labels. Our experiments suggest that the latter definition is better for the multi-label scenario. However, the performance of this baseline TNN model which uses ontology in-

formation suggests that this external knowledge is still difficult to embed into a model for weakly labeled data. Furthermore, we believe that this definition of "same" or different" sub/superclass is still ambiguous and introduces even more noise into the model as it attempts to cluster pairs that are not really the "same" subclass. This can explain why we see such poor performance metrics compared to the MLP with no ontology information.

Model	Weighted AP		Weighted AUC	
	Subclass	Superclass	Subclass	Superclass
MLP	0.4509	0.7056	0.8706	0.8556
TNN + Ont.	0.3653	0.3876	0.8055	0.6505
TNN + GCN	0.4285	0.6790	0.8460	0.8280
MLP + GCN	**0.4590**	**0.7117**	**0.8751**	**0.8602**

Table 1: mAP and AUC Results of different models. TNN. = Twin Neural Network, Ont. = Ontology.

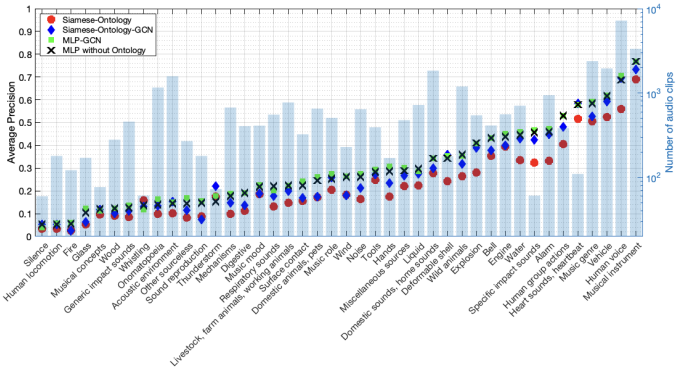

Fig. 3: mAP across different low-level labels

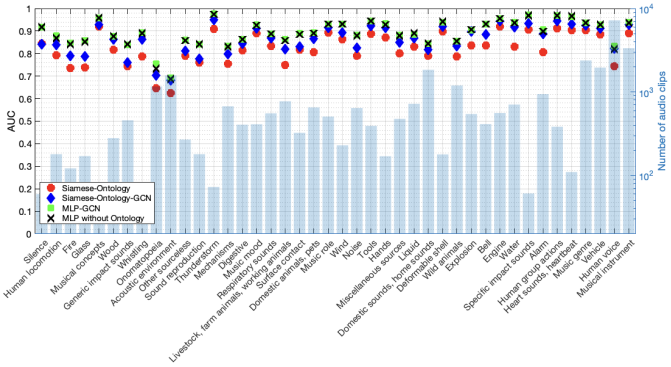

Fig. 4: AUC across different low-level labels

3.4. Performance of Twin Neural Network-GCN Model

The extension of the TNN model replaces the Ontology layer with a Graph Convolution network to embed ontology information. We trained the model with the Adam optimizer for about 30 epochs. We use 2 Linear Layer as the node embedding of GCN and apply 2 Layer GCN. The performance is sensitive to the hyper-parameters λ_1, λ_2 and λ_3 of the loss function because it would amplify or reduce the learning rate for different loss terms. We find the observation is that the greater λ is not necessarily equal to the greater learning rate.

The TNN with GCN tolerate noises introduced from weakly labeled data better than the baseline TNN with an un-trainable Ontological Layer. This suggests that the GCN can better capture the ontology hierarchy and that it can even overcome the noise introduced by attempting to find pairs for the TNN. Overall, the evaluation metrics are close to baseline MLP with no ontology information. [14]

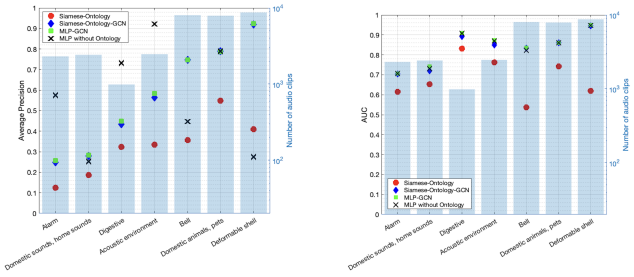

Fig. 5: AP (left) and AUC (right) across different superclass labels

3.5. Performance of MLP-GCN Model

To study the utility of the TNN we also implement an MLP-GCN using the same GCN structure as the one in the TNN-GCN. We investigate which correlation matrices can provide the best performance improvement. Among the 3 correlation matrices, we found label-co-occurrence-based correlation works best to model ontology information in the weak label scenario.

Consequently, we run experiments on the two trainable parameters: t, p. Experiments showed that $p = 0.2, t = 0.08$ gave the best results. The model performs better than previous models but improvement over an MLP without an Ontological Layer is limited. This suggests that even with the ontology information embedded into the GCN, it is still hard to get significant improvements in classification results. It further demonstrates that the architecture of the TNN net is not a great fit for the multi-label scenario. Using just a simple MLP to learn embeddings, it can already achieve relatively good performance with the GCN. Overall, we found that there were certain classes for which it was always difficult to achieve high AP or AUC scores as demonstrated in Figures 3, 4, 5. For these classes, e.g. glass, fire, or silence, we analyzed which data points contain those classes and found that they are often multi-labeled with more commonly found classes in the training set, such as human voice, or domestic sounds. This could make it difficult for the model to learn different representations for those classes.

4. CONCLUSIONS

To conclude, we observed that the Twin neural network model with the ontology layer has the worst performance, while a simple MLP obtains better results, and the combination of MLP with GCN works even better. It is possible for the TNN architecture to have a negative impact on prediction, introducing confusion through the construction of input pairs for data that is weakly multi-labeled. Although the GCN provides a slight improvement, it appears the model still has difficulty in differentiating ambiguous subclasses or using the hidden knowledge in weakly labeled data. The way to incorporate ontology information may differ with context (dataset, ontological architecture, network structure, etc.); additional investigation is needed to determine the best ways to use it. For a dataset like AudioSet, where each instance has multiple, or even wrong labels, having a simple ontology layer at the output of TNN might not be a good choice. So future works could narrow the scope of the datasets and try different approaches to embed ontological information and approaches that make use of deeper levels of ontology information.

5. REFERENCES

[1] Bhiksha Raj Abelino Jimenez, Benjamin Elizalde, "Sound event classification using ontology-based neural networks," *NIPS 2018 Workshop*, 2018.

[2] Konstantinos Sikelis, George E. Tsekouras, and Konstantinos Kotis, "Ontology-based feature selection: A survey," *Future Internet*, vol. 13, no. 6, 2021.

[3] Yuxia Geng, Jiaoyan Chen, Zhuo Chen 007, Jeff Z. Pan, Zhiquan Ye, Zonggang Yuan, Yantao Jia, and Huajun Chen, "Ontozsl: Ontology-enhanced zero-shot learning.," in *WWW '21: The Web Conference 2021, Virtual Event / Ljubljana, Slovenia, April 19-23, 2021*. 2021, ACM / IW3C2.

[4] George A. Miller, "Wordnet: A lexical database for english," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.

[5] T. Mitchell, W. Cohen, E. Hruschka, P. Talukdar, J. Betteridge, A. Carlson, B. Dalvi, M. Gardner, B. Kisiel, J. Krishnamurthy, N. Lao, K. Mazaitis, T. Mohamed, N. Nakashole, E. Platanios, A. Ritter, M. Samadi, B. Settles, R. Wang, D. Wijaya, A. Gupta, X. Chen, A. Saparov, M. Greaves, and J. Welling, "Never-ending learning," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI-15)*, 2015.

[6] M. Sheraz Anjum Kamran Munir, "The use of ontologies for effective knowledge modelling and information retrieval," *Applied Computing and Informatics 14 (2018)*, 2018.

[7] Gordon Wichern, Brandon Mechtley, Alex Fink, Harvey Thornburg, and Andreas Spanias, "An ontological framework for retrieving environmental sounds using semantics and acoustic content," *EURASIP J. Audio Speech Music Process.*, vol. 2010, no. 1, Dec. 2010.

[8] 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. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[9] Ankit Shah, Anurag Kumar, Alexander G Hauptmann, and Bhiksha Raj, "A closer look at weak label learning for audio events," *arXiv preprint arXiv:1804.09288*, 2018.

[10] 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 *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017.

[11] Harsh Shrivastava, Yfang Yin, Rajiv Ratn Shah, and Roger Zimmermann, "Mt-gcn for multi-label audio-tagging with noisy labels," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 136–140.

[12] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo, "Multi-label image recognition with graph convolutional networks," 2019.

[13] Thomas N. Kipf and Max Welling, "Semi-supervised classification with graph convolutional networks," 2017.

[14] Laurie M Heller, Benjamin Elizalde, Bhiksha Raj, and Soham Deshmuk, "Synergy between human and machine approaches to sound/scene recognition and processing: An overview of icassp special session," *arXiv preprint arXiv:2302.09719*, 2023.