Joining datasets via data augmentation in the label space for neural networks

Jake Zhao (Junbo) [*1] **Mingfeng Ou** [*2 3] **Linji Xue** [2] **Yunkai Cui** [2] **Sai Wu** [1] **Gang Chen** [1]

Abstract

Most, if not all, modern deep learning systems restrict themselves to a single dataset for neural network training and inference. In this article, we are interested in systematic ways to join datasets that are made of similar purposes. Unlike previous published works that ubiquitously conduct the dataset joining in the uninterpretable latent vectorial space, the core to our method is an augmentation procedure in the label space. The primary challenge to address the label space for dataset joining is the discrepancy between labels: non-overlapping label annotation sets, different labeling granularity or hierarchy and etc. Notably we propose a new technique leveraging artificially created knowledge graph, recurrent neural networks and policy gradient that successfully achieve the dataset joining in the label space. Empirical results on both image and text classification justify the validity of our approach.

1. Introduction

The advances of deep learning (LeCun et al., 2015) arise in many domains, such as computer vision (Krizhevsky et al., 2017), natural language processing (Sutskever et al., 2014), speech (Oord et al., 2016), games (Silver et al., 2017) and etc. In particular, the most popular paradigm to date is the so-called *end-to-end* learning paradigm. Its recipe can normally be summarized as follows: (i)-prepare a dataset consists of numerous {input-target} groups; (ii)-feed the dataset to a model, or coupled with a pre-trained model; (iii)-optimizing the model by a gradient-based method and finally, (iv)-inference on testing data points. In spite of its massive successes, we argue that a given dataset should not be only used once for one phase of task. **Simply put, why train your neural network using just one dataset?** Instead, its versatility can be substantially enhanced by a novel framework of dataset joining.

In real world applications, we often have the choice of multiple datasets for the same task. Perhaps some datasets can easily be combined, but some cannot. The main bottleneck to join datasets together, we argue, is the label discrepancy — that is, the discrepancy caused by inconsistent label set, different semantic hierarchy or granularity. Prior work attempting to solve this problem has ubiquitously focused on mixing methodologies in the vectorial hidden space, enabled by transfer learning algorithms (He et al., 2019), adversarial training (Ganin et al., 2016) and etc. These works, however, suffer from a lack of interpretability and oftentimes an inadequate exploitation of the semantic information of the labels. To the best of our knowledge, how to combine different datasets directly in the label space remains an untouched research challenge.

In this article, we aim to propose a new framework to join datasets and directly address the label space mixing or joining. Unlike the conventional deep learning paradigm where for each input the model is optimized towards predicting the corresponding label, our paradigm extends to predicting a trajectory on a *label graph* that traverses through the targeted label node. The construction of the label graph is the core element in our paradigm. Later in this paper, we will describe how to construct it in detail; briefly, the label graph can be perceived as an integration of all the labels (as nodes) from considered datasets, in addition to a set of augmented nodes which are used to bridge the isolated label nodes.

In a nutshell, the label graph is knowledge-driven. The construction of it simulates the human process of a decision. We take cat breed classification as an example. Traditional paradigm may only have delaminated cat breeds label such as ⟨*british-shorthair*⟩, ⟨*ragdoll*⟩, and etc. By contrast, in our paradigm, the constructed label graph would not only consist of all the cat breed labels as nodes, but also compose several additional *augmented nodes*. These augmented nodes are functional to indicate *certain features* revealed in its descendant nodes, such as the color feature ⟨*tabby-color*⟩ being the ancestor node of cat breed label nodes like ⟨*egyptian-mau*⟩, ⟨*bengal*⟩ and ⟨*maine coon*⟩. Likewise, the

[*]Equal contribution [1]College of Computer Science and Technology, Zhejiang University, Zhejiang, China [2]Graviti Inc., Shanghai, China [3]Department of Software Engineering, Tongji University, Shanghai, China. Correspondence to: Gang Chen <cg@zju.edu.cn>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021.

augmented nodes can also contain hair features, eye color features and etc. An illustration is displayed in Figure 1.

We postulate our system to have following general merits: (i)-it systematically unifies different label systems from different datasets, tackling the inconsistent label sets; (ii)-enhanced transparency and interpretability. Unlike traditional single-label end-to-end prediction being widely criticized as black-box, our paradigm offers a "decision process" thanks to the trajectory prediction mechanism.

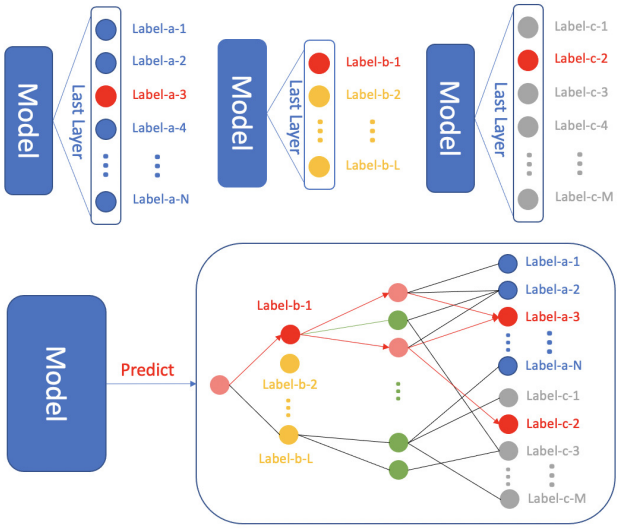

Figure 1. Single label prediction paradigm (top) vs. Label graph prediction paradigm (bottom). Three different colors (i.e., blue, yellow and gray) indicate their source of datasets (blue for dataset-a, yellow for dataset-b and gray for dataset-c). The green and pink ones are the augmented nodes while the red nodes denote the groundtruth targets.

We instantiate this paradigm by using a recurrent neural network trained with a combination of teacher forcing (Sutskever et al., 2014) and REINFORCE (Williams, 1992) handling different types of augmented nodes. The contributions of this work are:

- Propose a new paradigm based on label space augmentation that directly joins datasets in the label space;
- Develop a novel instantiation and training algorithm concept-proofing the validity of the paradigm, exhibiting promising results on both image and text classification;
- Enhanced interpretability and causal traceability for the proposed approach than the conventional end-to-end paradigm.

The rest of the paper is organized as follows. Section 2 outlines the related work throughout the literature. In section 3, we first formally describe a more abstracted version of the system and then go to a more concrete model instance. Section 4 shows the experimental results. Section 5 concludes the paper.

2. Related Work

In this section, we examine the existing related work.

2.1. Pre-trained models

The concept of pretrained models has been very popular in recent years. From computer vision (Misra & Maaten, 2020; He et al., 2019; Chen et al., 2020), to natural language processing (Devlin et al., 2018; Yang et al., 2019; Radford et al.), pretrained models demonstrate promising results for performance gain on a variety of domain and tasks.

Fundamentally, the pretrained models are often obtained from a separate and large-scale dataset, and trained under certain criteria (mostly unsupervised). This can be seen as merging the knowledge from this large-scale dataset to the (often much smaller) dataset involved in the downstream task. Notice that this joining or merging procedure is enacted in the feature space and enabled by gradient-based techniques.

2.2. Transfer learning

The idea of transfer learning is to reuse the past experience from a source task to enhance models' performance on a target task, most commonly including parameter finetuning or feature reusing. This area has been studied for many years (Pratt et al.; Kornblith et al., 2019a; Huh et al., 2016; Yosinski et al., 2014; Bozinovski, 2020; Yu et al., 2017). As early as it is in 1976, Bozinovski (2020) offers a mathematical and geometrical model of transfer learning in the neural network. Modern work includes (Yu et al., 2017) investigating transfer learning with noise, (Huh et al., 2016) examining the underlying reason of ImageNet (Deng et al., 2009) being a good dataset for transfer learning and (Yosinski et al., 2014) addressing the transfer capacity at different layers in a neural network. More recently, (Raghu et al., 2019) assessed reused feature in the meta learning framework like MAML (Finn et al., 2017). Neyshabur et al. (2020) argued that the features from the latter layer from a neural network have a better influence for transfer learning. Some negative results are also well conveyed from the community. (Kornblith et al., 2019b) challenges the generalizability of a pretrained neural network obtained from ImageNet. (He et al., 2019) shows that transfer learning does not necessarily yield performance gains.

Nonetheless, these works are generally complementary to ours. The primary transfer mechanism is conducted in the feature space, either from parameter porting or gradient-

based finetuning. Our paradigm aims directly at the (often discrete) label space joining.

2.3. Knowledge distillation and pseudo labels

Recently, Pham et al. (2020) extends from the original pseudo label idea (Yarowsky, 1995; Lee, 2013) to reach superior results on ImageNet. The idea of pseudo label relies on a teacher network to generate *fake* labels to teach a student network. While we find this line of work similar to ours owing to the label space manipulation, it differs from our approach in two aspects: (i)-prior work on pseudo label is often limited to an unsupervised setup where the pseudo labels are tagged with unlabeled datasets; (ii)-in our paradigm we do not have any machine-generated labels but we rely on a domain knowledge-based label graph. In addition, we will compare our approach against a revised version of supervised pseudo label in section 4.

2.4. Label Relationship and Classification

Label Relationship has shown promising potential in classification tasks exhibited by existing works. Deng et al. (2014); Ding et al. (2015) were among the first to point out the concept of label relation graph, in which the authors proposed to construct a label relation graph and used set-based or probabilistic classifiers. However, this line of work cannot deal with nondeterministic paths which commonly exist in an off-the-shelf label graph nor discuss the datasets joining scheme. Ristin et al. (2015) adopted Random Forests and propose a regularized objective function that takes into account unique parent-child relations between categories and subcategories to improve classification by coarser labeled data. (Wang et al., 2016) proposed a method that recurrently lets the prediction flow through a same set of ground truth labels, so as to exploit co-occurrence dependencies among multi objects in an image and improve multi-classes classification performance. Similarly, Hu et al. (2016) pointed out visual concepts of an image can be divided into various levels and proposed an rnn based framework to capture inter-level and intra-level label semantics.

Our work is similar to some of the above works mainly from a methodology perspective, but differs in two main aspects: (i) instead of leveraging label relations in a single set(or rather in a single image), our framework is capable of incorporating labels from both internal and external set, and across all hierarchies or granularities; (ii)-our framework can resolve the multi-datasets joining in the label space.

3. Method

3.1. Setup

3.1.1. Dataset joining problem

Let $D^A = \{(\mathbf{x}_1^a, y_1^a), (\mathbf{x}_2^a, y_2^a) \cdots (\mathbf{x}_{N_a}^a, y_{N_a}^a)\}$ and $D^B = \{(\mathbf{x}_1^b, y_1^b), (\mathbf{x}_2^b, y_2^b) \cdots (\mathbf{x}_{N_b}^b, y_{N_b}^b)\}$ be the targeted datasets respectively. The annotated labels for two datasets may or may not overlap.

Conventional deep learning system relies on an end-to-end scheme where two functions are trained separately to map X to Y:

$$f^a(\mathbf{X}^a) \rightarrow Y^a, \quad f^b(\mathbf{X}^b) \rightarrow Y^b.$$

In this work, we are interested in combining two datasets via a label space joining algorithm. For example, in a classification setup, we intend to overcome the main bottleneck — the discrepancy between the label sets of the datasets. The traditional softmax-based classifiers are inefficient for the joining. The foundation of a softmax function is to introduce "competition" into the prediction process. The joined label set commonly exists labels that are not mutually exclusive in the taxonomy.

3.2. Method

3.2.1. Knowledge-driven label graph

We consider the most challenging and perhaps most common situation where the label sets have no overlap, $\cap(\mathcal{Y}_a, \mathcal{Y}_b) = \varnothing$.

First and foremost, we construct a label graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ covering the label sets of both datasets, where $\mathcal{V}$ is the set of N_v nodes, $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of N_e edges connecting M pairs of nodes in $\mathcal{V}$. Specifically, we construct the graph based on the following steps:

1. start an empty graph with only a root node v_0, $\mathcal{V} = \{v_0\}$;
2. add labels from both datasets as nodes to the graph, $\mathcal{V} = \{v_0\} \cup \mathcal{Y}^a \cup \mathcal{Y}^b$;
3. based on the labels' semantic information and domain knowledge, we add augmented nodes $\mathcal{Y}_i$ to the graph, $\mathcal{V} = \{v_0\} \cup \mathcal{Y}^a \cup \mathcal{Y}^b \cup \mathcal{Y}^{\text{aug}}$;
4. add links $\mathcal{E}$ to connect the nodes;

Oftentimes, when constructing the label graphs, we primarily want to base it on the domain knowledge that is accumulated across several decades for the considered task. Take the pet breed classification as an example. We tweaked slightly the above steps into: (i)-we crawl a complete knowledge graph K from a domain website like Purina; (ii)-we start with an empty graph G and place the root node $\langle \textit{animal} \rangle$; (iii)-going top-down the taxonomy of K, we select related nodes $\langle \textit{cat} \rangle$ and $\langle \textit{dog} \rangle$ and place them under the root; (iv)-repeat (iii), going further down through K, we place the augmented nodes in K representing the

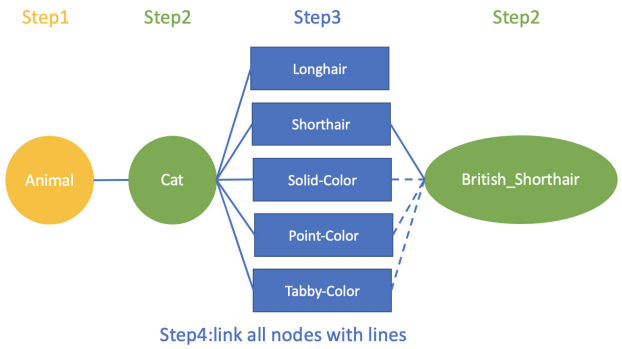

Figure 2. A subset example of constructed Label Graph. The root node is colored yellow. Two label nodes from $\mathcal{Y}^a$ and $\mathcal{Y}^b$ are colored green. Augmented nodes in $\mathcal{Y}^{\text{aug}}$ are blue.). The dashed line indicates nondeterministic paths, see text for more details.

visual features describing a pet like hair, ear, tail features, etc. (v)-repeat the previous steps until the finest granularity of K.

The exemplar label subgraph is plotted in Figure 2.

Note that the label graphs are inherently obtained from the domain knowledge which has been accumulated for decades across different fields. In particular, the label graphs we used for the experiments are all off-the-shelf with minor filtering and modification. For instance, we got the Pet label graph from websites like Purina [1], and the Arxiv label graphs are collected by its website routing logics. See section 4 for more details.

In terms of the **scalability** or **expandability** of our approach, we argue that extending a label graph is much cheaper than enlarging the annotated dataset itself because the latter requires significantly more human efforts in annotation.

3.2.2. Prediction Paths

In this subsection, we define *prediction paths*. Unlike the conventional deep learning paradigm maximizing the log-likelihood of the predicted label conditioned on its input, our system optimizes the log-likelihood of an entire prediction path (or trajectory) that traverses from the root node to the groundtruth label node.

More formally, we define a prediction path P obtained by running a graph traversal algorithm: $v_0 \rightarrow v_1 \rightarrow v_2 \rightarrow v_{\langle \text{eop} \rangle}$, where v_0 is the root node and "eop" indicates the end of a path. In the remainder of the paper, we may abbreviate this form to $v_0 \circ v_1 \circ v_2 \circ v_3 \circ v_{\langle \text{eop} \rangle}$. As we mentioned in the previous subsection, some of the nodes on P might be a label node belonging to $\mathcal{Y}^a$ or $\mathcal{Y}^b$ while some of the others might be augmented ones $\mathcal{Y}^{\text{aug}}$. In addition, when one or several nodes on the path are label nodes, we may also write this prediction path down as $P(v_j)$, when v_j is a label node.

Let us give a concrete example. Given a data point sampled from either datasets, (x, y). We first locate the label node v representing y on the label graph $\mathcal{G}$, then we run a graph traversal algorithm to obtain all possible prediction paths traversing from v_0 to v. This process winds up with a set of prediction paths, $(P_0(v), P_1(v), \cdots P_M(v))$. The loss objective in the system is to maximize the log-probability of all collected prediction paths. In the remainder of the paper, we may also call these paths as groundtruth prediction paths, or in short groundtruth paths.

3.2.3. Paths Categorization

Briefly we categorize the groundtruth paths into two types: the *deterministic* and *nondeterministic* path.

Definition 3.1 (Competing nodes) *We define u and w are competing nodes when u and w share a same ancestor node, and they are mutually exclusive in the taxonomy.*

Definition 3.2 (Deterministic path) *Given a path anchoring label node v, $P(v)$, the path is deterministic if there does **not** exist another path $P'(v)$ such that the following conditions are met: node $u \in P(v)$, node $w \in P'(v)$, u and w are competing nodes, and $u \neq w$.*

Definition 3.3 (Nondeterministic path) *Given a path anchoring label node v, $P(v)$, the path is nondeterministic if there does exist another path $P'(v)$ such that the following conditions are met: node $u \in P(v)$, node $w \in P'(v)$, u and w are competing nodes, and $u \neq w$.*

Let us give a concrete example. In Figure 2, there are two label nodes colored green and five intermediate nodes colored blue. Among these intermediate nodes, $\langle$*Shorthair*$\rangle$ and $\langle$*Longhair*$\rangle$ are competing nodes which can be named as *Hair* group, and every two nodes among $\langle$*Solid-Color*$\rangle$, $\langle$*Tabby-Color*$\rangle$ and $\langle$*Point-Color*$\rangle$ are also competing nodes which can be named as *Color Pattern* group. Thus, these nodes form one deterministic path (*Animal*$\circ$*Cat*$\circ$*Shorthair*$\circ$*British-Shorthair*) because all samples of $\langle$*British-Shorthair*$\rangle$ is short hair, and three nondeterministic paths (e.g., *Animal*$\circ$*Cat*$\circ$*Solid-Color*$\circ$*British-Shorthair*) because the color pattern of $\langle$*British-Shorthair*$\rangle$ samples can be any node in *Color Pattern* group.

It is relatively straightforward to tackle the prediction of the deterministic paths. It can be implemented by a standard seq2seq (Sutskever et al., 2014) model trained by maximum log-likelihood estimation (MLE) assisted by the teacher forcing mechanism. Note that, different groups of competing nodes use softmax independently because we

[1] https://www.purina.com/

assume that nodes in different groups are independent. On the other hand, when facing nondeterministic paths, teacher forcing cannot be applied due to the uncertainty from the intermediate nodes. Therefore, we use the policy gradient algorithm to estimate the gradient from the nondeterministic paths.

3.2.4. BLOCK SOFTMAX

For general softmax, its output represents the relative probability between different categories, which means that all the categories are competing against each other. However, in our architecture, the competitive relationship only exists among competing nodes. Thus, in order to handle this situation, we adapted softmax as follows:

$$p_u = \frac{e^{z_u}}{\sum_{w \in \mathbb{S}_u} e^{z_w}}$$

where $\mathbb{S}_u$ denotes the set of competing nodes of node u, and z denotes the input to the softmax. This means that we limit the calculation of relative probability within each competing node groups. We call this revised version as block softmax function.

More concretely, for the Pet problem, there are nodes describing hair feature and color pattern feature respectively. The nodes inside each of these two groups (or blocks) are *competing nodes*. Softmax inherently introduces competition inside considered predicted labels, hence we place two softmax classifiers for these two groups (or blocks) and the nodes across the groups do not influence each other.

3.2.5. DETERMINISTIC PATH PREDICTION

Essentially we treat each of the groundtruth deterministic paths as a sequence and let a decoder predict each token (e.g. node) autoregressively. Because every token on a deterministic path is guaranteed to be the groundtruth label against their competing nodes respectively, this allows us to borrow the flourishing literature from the sequence decoding (Sutskever et al., 2014; Cho et al., 2014; Chung et al., 2014). That being said, we adopt teacher forcing as our training standard.

Formally, given a deterministic path $P = (v_0, v_1, v_2, ...v_N)$, we feed the sequence into a recurrent unit and adopt the teacher-forcing strategy (Williams & Zipser, 1989) during training. Therefore we can gather the hidden state as follows:

$$\mathbf{f}_t = g(\mathbf{e}_t, \mathbf{f}_{t-1}), \tag{1}$$

where g is a compositional function such as gated recurrent units (Cho et al., 2014), $\mathbf{f_t}$ denotes the feature vector at the token step t, $\mathbf{e}$ is a learned node embedding matrix.

At the top layer of the recurrent unit, we extract the feature $\mathbf{f}$ and then carry it to a block softmax predictor at each position. The recurrent decoder maximizes the conditional log-probability $\log p(v_0, v_1, v_2 \cdots v_{\langle \text{eop} \rangle}) \mid \mathbf{x})$ Note this training process is analogous to many natural language processing tasks, such as machine translation (Vaswani et al., 2017). At each step t, the overall objective for the certain paths can be derived in an autoregressive fashion:

$$\mathcal{L}_d = -\sum_t \log p(v_{t+1} | v_t, \mathbf{f}_t) \tag{2}$$

The gradient from predicting the deterministic paths can be easily and tractably computed using backpropagation.

3.2.6. NONDETERMINISTIC PATHS

Predicting nondeterministic paths is generally more challenging, due to the nondeterministic nodes in the groundtruth path blocking the usage of normal sequence training techniques like teacher forcing. To cope with this, we first define a reward function.

$$r(\hat{P}) = \frac{1}{|S|} |\hat{P} \cap S|, \tag{3}$$

where $\hat{P}$ is a generated path sampled by the model (from a multinomial distribution produced at each node step), S is a set composed by the groundtruth label nodes corresponding to the input, $|\cdot|$ denotes the size of a set.

Then we write down the loss function for the nondeterministic path prediction:

$$\mathcal{L}_{nd} = -\sum \log p(\hat{P}) r(\hat{P}) = -\mathbb{E}_{\hat{P} \sim p(P)} r(\hat{P}), \tag{4}$$

where $p(P)$ is the path prediction produced by the model and the $r(\hat{P})$ is the associated reward. In practice, we approximate the expectation via a single sample generated from the distribution of action (node choices).

Without too much details, the gradient of L_{nd} can be estimated by:

$$\frac{\partial \mathcal{L}_{nd}}{\partial \mathbf{f}_t} = (r(\hat{P}) - b(r)) p(\hat{v}_{t+1} | \hat{v}_t, \mathbf{f}_t), \tag{5}$$

where $b(r)$ is an average baseline estimator. For simplicity we omit the details of the gradient derivation, we refer the readers to Williams (1992) for more details.

3.2.7. MODEL INSTANTIATION

To this end, we finalize the instantiation of a model instance fitting in this paradigm. As we described earlier, our model is devised to predict a node path rather than a single node.

The overall model instantiation in computer vision tasks is illustrated in Figure 3. Briefly, this model employs a seq2seq alike structure (Sutskever et al., 2014); we use an EfficientNet-B4 (Tan & Le, 2019) as our encoder backbone,

Algorithm 1 Training algorithm

Input: images $\{x^{(k)}\}_{k=1}^{m}$, paths $\{P^{(k)}\}_{k=1}^{m}$ ($P^{(k)} = \{v_1^{(k)}, ..., v_n^{(k)}\}$), uncertain sample indexes $I_{pg} = \{a_{k'}\}_{k'=1}^{q} (0 \leq q \leq m, 1 \leq a_{k'} \leq m)$

Parameter: LabelGraph $\mathcal{G}$, MaxLength n, Teacher Force Rate $r_{tf} (0 \leq r_{tf} \leq 1)$

Let loss $\mathcal{L} = 0$
/*1.Train deterministic path by teacher forcing*/
Let Sample $coin \sim \mathcal{U}(0, 1)$
Let token $t_0 = \{\langle START \rangle^{(k)}\}_{k=1}^{m}$
Define encoder function g_{enc}, decoder function g_{dec}
Compute feature $f = g_{\text{enc}}(\{x^{(k)}\}_{k=1}^{m})$
Initial $\mathbf{f}_0$ by f
for $i = 1$ **to** n **do**
  Let probabilities $p_i, \mathbf{f}_i = g_{\text{dec}}(\mathcal{G}, f, \mathbf{f}_{i-1}, t_{i-1})$
  Compute loss at time step i and accumulate: $\mathcal{L} = \mathcal{L} + (-\log p(t_i | t_{i-1}, \mathbf{f}_i))$
  if $coin \leq r_{tf}$ **then**
    $t_i = \{v_i^{(k)}\}_{k=1}^{m}$
  else
    $t_i =$ Indexes of max values in p_i
  end if
end for

/*2.Train nondeterministic path by policy gradient*/
if I_{pg} *is not empty* **then**
  Sample $\{x^{(k)}\}_{k=1}^{m}$ according to I_{pg} and assign to x^{pg}
  Sample $\{P^{(k)}\}_{k=1}^{m}$ according to I_{pg} and assign to P^{pg}
  Let $t_0 = \{\langle START \rangle^{(k')}\}_{k'=1}^{q}$
  Compute feature $f = g_{\text{enc}}(x^{pg})$
  Let path probabilities $pp = [\,]$
  for $j = 1$ **to** n **do**
    Let probabilities $p_j = g_{\text{dec}}(\mathcal{G}, f, t_{j-1})$
    Sample nodes $t_j \sim p_j$
    Append sampled nodes' probabilities $p_j^{t_j}$ at time step j to pp
  end for
  Compute loss $\mathcal{L} = \alpha * \mathcal{L} + \beta * (-\sum \log pp * r(t_{n-1}))$
end if
Backpropagate $\mathcal{L}$, and update g_{enc}, g_{dec}.

Algorithm 2 Decoding algorithm

Input: image x
Parameter: LabelGraph $\mathcal{G}$, MaxLength n
Output: Path based on LabelGraph $\mathcal{G}$

Let token $t_1 = \langle START \rangle$
Let path $p = [\,]$
Define encoder function g_{enc}, decoder function g_{dec}
Compute feature $f = g_{\text{enc}}(x)$
for $i = 1$ **to** n **do**
  Let $prob = g_{\text{dec}}(\mathcal{G}, f, t_i)$
  $t_i =$ Index of Max value in $prob$
  if $t_i == \langle EOP \rangle$ **then**
    break
  else
    Append t_i to p
  end if
end for
return p

and a gated recurrent unit in the decoder. As we described earlier, the overall model is trained by a combination of techniques: gradient descent, backpropagation, teacher forcing and policy gradient. Our training procedure is compiled Algorithm 1. Meanwhile, during the inference phase, we simply apply a greedy decoding algorithm, showed in Algorithm 2. Analogously, in natural language processing task, we simply replace the EfficientNet structure with BERT (Devlin et al., 2018).

4. Experimental Evaluation

In this section, we show the empirical results using our presented architecture.

4.1. Setup

Datasets To validate our approaches, we experiment with two modalities of data: images and natural language. And just for proof-of-concept, we choose the most conventional task — image and text classifications. The datasets statistics are shown in Table 1. In the domain of computer vision, we use (i)-the Oxford-IIIT Pet dataset (Parkhi et al., 2012) and Dogs vs. Cats[2] as a group; (ii)-102 Category Flowers dataset (Nilsback & Zisserman, 2008) and the 17 Category Flowers dataset (Nilsback & Zisserman, 2006) as another group. In particular, the chosen dataset groups can be characterized as having one dataset very fine-grained annotated with the other being much coarser. To expand the horizon of the experiments, in group (i) there is **no** label overlap between the datasets, while for the group (ii) **8** labels are shared in the sets. Notice that, for evaluation purposes, we aim at the performance on the finer-grained datasets. It is relatively easy to enhance the coarser-level performance by fusing the finer-grained labels (through a label set mapping table for example). So we focus on the significantly more challenging task where we intend to enhance the performance on finer-grained datasets by utilizing its coarsely annotated counterpart.

On the other line of natural language processing experiments, we use the Arxiv dataset (Clement et al., 2019). The downloadable version of this dataset involves a gigantic number of articles released on arxiv throughout the entire 28 years. In this work, we only take a subset to conduct our experiments. The goal for this set of experiments is for a hierarchical multi-label text classification, thanks to the naturally hierarchical labels from Arxiv[3], such as "cs.machine-learning". In particular, we artificially prepare two datasets: an *Arxiv Original* dataset with 50,000 samples

[2]https://www.kaggle.com/c/dogs-vs-cats
[3]https://arxiv.org/

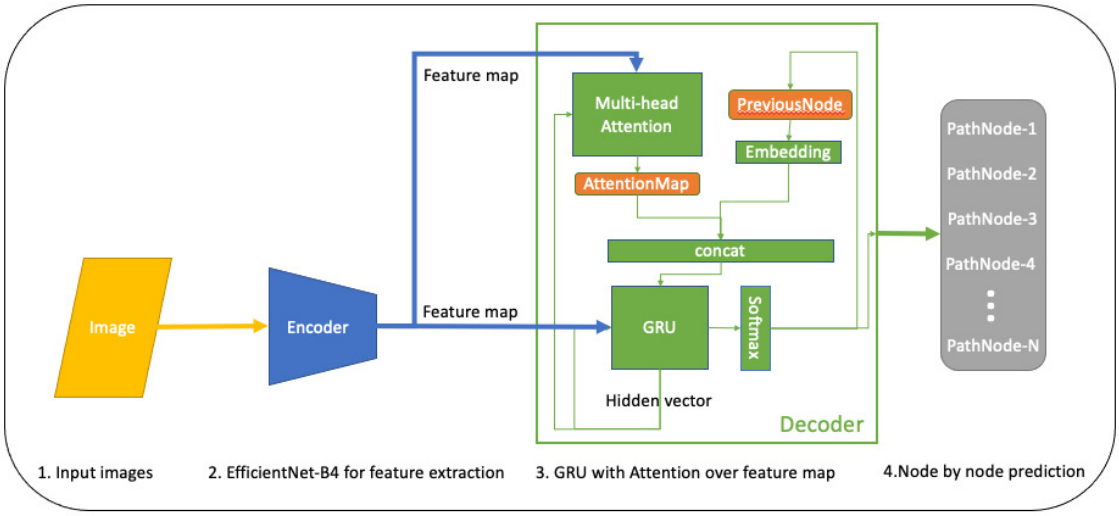

Figure 3. Our model architecture

and an *Arxiv Augment* dataset with 50,000 samples but only maintaining the coarser-level labels (such as "cs"). Similar to computer vision tasks, we conduct the experiments on both the *Arxiv Original* dataset and the combined dataset dubbed Arxiv Fusion.

Baseline setup In computer vision tasks, we adopt the state-of-the-art model EfficientNet (Tan & Le, 2019) as the baseline results that we compare against. The reasons are the following: (i)-EfficientNet framework is one of the state-of-the-art pretrained frameworks in computer vision. Its performance on the Pet datasets is ranked the top on the board. (ii)-As we directly use a pretrained EfficientNet framework as our encoder, it is very straightforward to see if our techniques of label augmentation and dataset fusion are effective. We use EfficientNet-B4 version instead of their biggest B7 version due to time and space complexity issues.

In addition, we compare against the pseudo label method (Lee, 2013). Take the pet group datasets as an example. We first train a standard fine-grained classification model, and use it to generate *pseudo labels* on the images provided by the coarser dataset, i.e. Dogs vs Cats. We additionally use the datasets' coarse labels (dogs versus cats) to conduct a filtering (if the produced pseudo label is incorrect and detected by the filter, we remove the data sample from the fused set). Thereby we have a combined dataset that possesses a unified fine-grained label system. We argue that this revision improves from the original unsupervised version of pseudo label (Lee, 2013) because we further leverage some weak supervision derived from the coarsely annotated labels.

Besides the above baseline setups, we conduct a multi-label classification setting dubbed as *Label Set*, adopted from one of the pivotal experiments in Deng et al. (2014). Specifically, we flatten all the labels on a ground-truth path into a set of labels and train a set-predictive multi-label classification network. Essentially this setup contains the same volume of information as our approach but lacking the label taxonomy information. We hope to use this comparison to demonstrate the necessity of the label graph.

Notably, the natural language processing experiments are established by two settings: one with a pretrained BERT model and the other one adopts an LSTM-based encoder trained on-the-fly.

4.2. Main results

We report our results in Table 3 and Table 2 respectively for vision and text classification. We may conclude from the scores that: (i)-the augmentation strategy in the label space enables dataset joining. The models we obtained from the fused dataset perform substantially better than the end-to-end learning paradigm. It also performs better than the improved version of the weakly-supervised pseudo label method; (ii)-Even without the help from the extra dataset, simply augmenting the label into label graph coupling with our training strategy still offers certain performance gains. We reveal the experimental details in the Appendix.

4.3. Interpretability results

As we mentioned in the Introduction section, an appealing by-product we characterize our framework with is much-enhanced interpretability than a black-box end-to-end system. It can be perceived that the augmented label graph offers a "decision process" of our model, when performing

Table 1. Dataset statistics. K denotes the number of classes. Note that the testing data for datasets with coarser annotation is **not** used.

DATASET	*#Train Data*	*#Test Data*	*K*
Oxford-IIIT Pet	3680	3669	37
Dogs vs. Cats	12,500	-	2
PetFusion	28,680	3669	39
102 Category Flower	1020	6149	102
17 Category Flower	680	-	17
FlowerFusion	1180	6149	102
ARXIV ORIGINAL	50000	50000	149
ARXIV AUGMENT	50000	-	21
ARXIV FUSION	50000	50000	149

Table 2. Text classification results, reported in F1.

MODEL	*Arxiv original*	*Arxiv Fusion*
BERT + FFN	75.5	76.9
BERT + label-aug (ours)	77.4	**79.3**
LSTM + FFN	69.9	–
LSTM + label-aug (ours)	**72.4**	–

an inference pass. In Figure 4 we plot three augmented nodes ($\langle$*Tabby-Color*$\rangle$, $\langle$*Point-Color*$\rangle$ and $\langle$*Solid-Color*$\rangle$) with some of the corresponding images. More specifically, when conducting an inference pass on displayed images on each row, the decoded path will traverse through the listed augmented (feature) node on the far left.

Furthermore, the color of the blocks indicates the property of the predicted path obtained during inference on the test set. The blue rectangle indicates the prediction is accomplished on deterministic paths while the green ones are nondeterministic paths. This result indicates the validity of our training scheme involving policy gradient and normal sequence training techniques.

4.4. Path correctness evaluation

To further scrutinize the trained model, we manually check the predicted paths' correctness. In particular we look into the predictions of nondeterministic paths, because for these paths, there aren't groundtruth associated with the ambiguous nodes, and sampling and policy gradient introduce a certain amount of uncertainty.

We center our inspection on the node $\langle$*Tabby-Color*$\rangle$. This node appears on the deterministic paths leading to label nodes including $\langle$*Abyssinian*$\rangle$, $\langle$*Egyptian Mau*$\rangle$ and $\langle$*Bengal*$\rangle$, and it turns out to be nondeterministic for $\langle$*Maine Coon*$\rangle$ and $\langle$*British-Shorthair*$\rangle$. Among the predicted nondeterministic paths traversing through $\langle$*Tabby-Color*$\rangle$ node, more than **90%** of these samples' color pattern does conform to be a tabby color.

To us, this is a very exciting result because it shows that our model is indeed capable of performing some reasoning on the label graph and resolving the ambiguity existing on the nondeterministic paths. We hope to leave the exploration of this line to the future work.

4.5. Ablation study

In this section, we attempt to identify the key factors influencing the model performance. We conduct a series of ablation study on the following factors:

- **the size of the label graph**. We compare our best result with the same architectural setup built on trimmed label graphs. Specifically, we obtain a *medium*-size label graph by trimming down the augmented nodes by 36%, and likewise, a *small*-size label graph by trimming down the augmented nodes by 63%.
- **minibatch construction**. In our training paradigm, each sample may correspond to several possible groundtruth paths. In practice, we could choose to sample (at most) N_p target paths, and implement the gradient with a *mean* or *sum* pooling operator towards each input sample. We could also just sample a *random* target path to do the training. Experiments show that the *mean* operation performs the best (in Table 3).

The results are obtained from the Pet dataset group, displayed in Table 4.

5. Outlooks

Why train your neural network using just one dataset?

In this article, we study the problem of dataset joining, more specifically in label set joining when there is a labeling system discrepancy. We propose a novel framework tackling this problem involving label space augmentation, recurrent neural network, sequence training and policy gradient. The trained model exhibits promising results both in performance and interpretability. Furthermore, we also tend to position our work as a preliminary attempt to incorporate the rich domain knowledge (formatted as knowledge graphs of the labels) to boost the connectivism (such as a neural network classifier). At last, we hope to use this work to motivate research for the multi-dataset joining setup for different tasks, and knowledge-driven label graphs with higher efficiency to be incorporated into deep learning.

Acknowledgement

Jake Zhao, Wu Sai and Chen Gang are supported by the Key RD Program of Zhejiang Province (Grant No. 2020C01024). JZ also thank Yiming Zhang for completing the preliminary

Table 3. Image classification results reported in accuracy (%). These results are obtained on the finer-grained testing datasets. ("×" denotes the considered model cannot be directly employed for the setup while "–" indicates the experiment is less prioritized.)

MODEL	*Oxford-IIIT Pet*	*PetFusion*	*102 Category Flower*	*FlowerFusion*
EfficientNet-B4 (Tan & Le, 2019) + FFN	93.84	×	90.91	91.01
EfficientNet-B4 + Label Set (Deng et al., 2014)	92.58	–	85.28	–
EfficientNet-B4 + Pseudo Labels	×	91.50	×	×
EfficientNet-B4 + label-aug (ours)	**94.66**	**94.95**	**92.80**	**93.27**

Figure 4. Interpretability results. The testing images at each row triggered the model inference process to traverse through the augmented node listed at far left (respectively for each row). The blue and green rectangles indicate deterministic and nondeterministic path prediction respectively.

Table 4. Results of the ablation study. See text in section 4.5 for details. The table shows the accuracy difference (%) from the best results in Table 3.(*OXF-PET* is short for *Oxford-IIIT Pet*.)

		OXF-PET	PETFUSION
GRAPH SIZE			
	MEDIUM GRAPH	-0.14	-
	SMALL GRAPH	-0.19	-0.1
MINIBATCH CONSTRUCTION			
	SUM	-0.71	–
	RANDOM	-0.36	–

baseline coding work for the ArXiv experiment.

References

Bozinovski, S. Reminder of the first paper on transfer learning in neural networks, 1976. *Informatica*, 44(3), 2020.

Chen, M., Radford, A., Child, R., Wu, J., Jun, H., Luan, D., and Sutskever, I. Generative pretraining from pixels. In *International Conference on Machine Learning*, pp. 1691–1703. PMLR, 2020.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

Chung, J., Gülçehre, Ç., Cho, K., and Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling. *CoRR*, abs/1412.3555, 2014. URL http://arxiv.org/abs/1412.3555.

Clement, C. B., Bierbaum, M., O'Keeffe, K. P., and Alemi, A. A. On the use of arxiv as a dataset, 2019.

Deng, J., Li, K., Do, M., Su, H., and Fei-Fei, L. Construction and Analysis of a Large Scale Image Ontology. Vision Sciences Society, 2009.

Deng, J., Ding, N., Jia, Y., Frome, A., Murphy, K., Bengio,

S., Li, Y., Neven, H., and Adam, H. Large-scale object classification using label relation graphs. In *European conference on computer vision*, pp. 48–64. Springer, 2014.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Ding, N., Deng, J., Murphy, K. P., and Neven, H. Probabilistic label relation graphs with ising models. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1161–1169, 2015.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 1126–1135, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR.

Ganin, Y., Ustinova, E., Ajakan, H., Germain, P., Larochelle, H., Laviolette, F., Marchand, M., and Lempitsky, V. Domain-adversarial training of neural networks. *The Journal of Machine Learning Research*, 17(1):2096–2030, 2016.

He, K., Girshick, R., and Dollár, P. Rethinking imagenet pre-training. In *Proceedings of the IEEE international conference on computer vision*, pp. 4918–4927, 2019.

Hu, H., Zhou, G.-T., Deng, Z., Liao, Z., and Mori, G. Learning structured inference neural networks with label relations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2960–2968, 2016.

Huh, M., Agrawal, P., and Efros, A. A. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Kornblith, S., Norouzi, M., Lee, H., and Hinton, G. Similarity of neural network representations revisited. *arXiv preprint arXiv:1905.00414*, 2019a.

Kornblith, S., Shlens, J., and Le, Q. V. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2661–2671, 2019b.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.

LeCun, Y., Bengio, Y., and Hinton, G. Deep learning. *nature*, 521(7553):436–444, 2015.

Lee, D.-H. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

Misra, I. and Maaten, L. v. d. Self-supervised learning of pretext-invariant representations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6707–6717, 2020.

Neyshabur, B., Sedghi, H., and Zhang, C. What is being transferred in transfer learning? *arXiv preprint arXiv:2008.11687*, 2020.

Nilsback, M.-E. and Zisserman, A. A visual vocabulary for flower classification. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pp. 1447–1454, 2006.

Nilsback, M.-E. and Zisserman, A. Automated flower classification over a large number of classes. In *Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008.

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. V. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

Pham, H., Xie, Q., Dai, Z., and Le, Q. V. Meta pseudo labels. *arXiv*, pp. arXiv–2003, 2020.

Pratt, L. Y., Mostow, J., and Kamm, C. A. Direct transfer of learned information among neural networks.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners.

Raghu, A., Raghu, M., Bengio, S., and Vinyals, O. Rapid learning or feature reuse? towards understanding the effectiveness of maml. *arXiv preprint arXiv:1909.09157*, 2019.

Ristin, M., Gall, J., Guillaumin, M., and Van Gool, L. From categories to subcategories: large-scale image classification with partial class label refinement. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 231–239, 2015.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Sutskever, I., Vinyals, O., and Le, Q. V. Sequence to sequence learning with neural networks. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems*, volume 27, pp. 3104–3112. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Tan, M. and Le, Q. V. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., and Polosukhin, I. Attention is all you need. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30, pp. 5998–6008. Curran Associates, Inc., 2017.

Wang, J., Yang, Y., Mao, J., Huang, Z., Huang, C., and Xu, W. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2285–2294, 2016.

Williams, R. J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256, 1992.

Williams, R. J. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270–280, 1989.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R. R., and Le, Q. V. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pp. 5753–5763, 2019.

Yarowsky, D. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pp. 189–196, 1995.

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H. How transferable are features in deep neural networks? In *Advances in neural information processing systems*, pp. 3320–3328, 2014.

Yu, X., Liu, T., Gong, M., Zhang, K., Batmanghelich, K., and Tao, D. Transfer learning with label noise. *arXiv preprint arXiv:1707.09724*, 2017.