The Missing Link: Finding label relations across datasets

Jasper Uijlings*, Thomas Mensink*, and Vittorio Ferrari

Google Research
{jrru, mensink, vittoferrari}@google.com

Abstract. Computer vision is driven by the many datasets available for training or evaluating novel methods. However, each dataset has a different set of class labels, visual definition of classes, images following a specific distribution, annotation protocols, etc. In this paper we explore the automatic discovery of visual-semantic relations between labels across datasets. We aim to understand how instances of a certain class in a dataset relate to the instances of another class in another dataset. Are they in an *identity*, *parent/child*, *overlap* relation? Or is there no link between them at all? To find relations between labels across datasets, we propose methods based on language, on vision, and on their combination. We show that we can effectively discover label relations across datasets, as well as their type. We apply our method to four applications: understand label relations, identify missing aspects, increase label specificity, and predict transfer learning gains. We conclude that label relations cannot be established by looking at the names of classes alone, as they depend strongly on how each of the datasets was constructed.

1 Introduction

Progress in computer vision is fueled by the availability of many different datasets, covering a wide spectrum of appearance domains and annotated for various task types, like ImageNet for classification [6], Open Images for detection [15], and KITTI for semantic segmentation of driving scenes [8]. Each of these datasets has its own set of class labels, its own visual definition for each class, its own set of images following a specific distribution, its own annotation protocols, and was labeled by a different group of humans annotators. As a result, the visual-semantical meaning of a certain label in a particular dataset is unique [22,26]. A few examples: (1) a `sofa` in ADE20k refers to the same visual concept as a `couch` in COCO, even though their class label is different; (2) ADE20k distinguishes `stool`, `armchair`, and `swivel chair` whereas COCO has a single concept `chair`. Moreover it is unclear if instances of `stool` would adhere to the annotation definition of the `chair` class in COCO; (3) ADE20k has the labels `floor` and `rug` whereas COCO distinguishes `floor-wood` and `rug-merged`. These are two ways of categorizing the visual world which are not fully compatible: a full-floor carpet is both a `floor` and a `rug-merged`, while a wooden floor is only a `floor` and a doormat is only a `rug-merged` (see also Fig. 1c).

* Equal contribution.

(a) identity (b) parent / child (c) overlap

Fig. 1: Examples of relations: (a) *identity*: both bicycle labels contain similar instances; (b) *parent/child*: the `animal` class contains instances which are either `cat` or `dog`; (c) `floor` and `rug-merged` *overlap* in the middle instance. But each label contains instances which are incompatible with the other label.

In this paper we want to automatically discover relations between labels across datasets. We aim to determine if the ADE20k `lake` and COCO `water-other` labels are related in their visual semantics. More specifically, are there visual instances which can be described by both labels? And what is the *type* of their relation? Do they represent the same visual concept? Are they in a parent/child relation? Or do they overlap like `floor` and `rug-merged`? Establishing such relations would enable combining datasets. This is useful for training on larger dataset with more visual concepts and more samples per concept, and also for evaluation purposes.

Joining datasets cannot be done by simply looking at the class label names; how labels across datasets are related really depends on the idiosyncrasies of each dataset involved. Indeed, Lambert et al. [16] recently proposed to unify multiple datasets into a single and consistent label space. This required a tremendous amount of manual work: matching all labels, visually verifying whether labels actually point to the same visual concepts, and re-annotating significant portions of each dataset into a single, mutually exclusive label space. Essentially [16] manually solved some of the kind of problems we want to address automatically. But also their result is prone to similar issues as described, their result depends on choices made for the definitions of labels, the annotation protocol, *etc.* Moreover as the number of datasets continues to grow, such heroic manual joining operation becomes infeasible and it will be necessary to do this automatically.

In this paper, we present methods for the automatic discovery of relations between labels across dataset. We distinguish different relation types (Tab. 1): *identity* (e.g. ADE20k `bicycle` and COCO `bicycle`), *parent/child* (e.g. ADE20k `animal` and COCO `dog`), and *overlap* (e.g. ADE20k `floor` and COCO `rug-merged`). We introduce methods to establish these relations by leveraging language cues, visual cues, and a combination of both.

In short, this paper presents an exploration into the discovery of how labels across datasets relate to each other. Our contributions are as follows: (1) We introduce a variety of methods to discover the existence of relations between labels across datasets, as well as their type (Sec. 3). These methods include vision, language, and their combination. (2) We demonstrate that we can effectively and automatically discover label relations between three semantic segmentation datasets: COCO [5,17,13], ADE20k [34], and Berkeley Deep Drive (BDD) [31]

(Sec. 4). To evaluate this quantitatively we leverage the MSeg annotations [16] to establish ground-truth label relations between these datasets. Additionally, we show that we can discover relations between different *types* of datasets by applying our method to ILSVRC12 image classification and COCO segmentation (supp. mat. Sec. A). (3) We demonstrate the usefulness of our method in four applications: *Understand label relations* (Sec. 5.1), in which we gain a deeper understanding of what types of relations exist and why they arise in practice; *Identify missing aspects* (Sec. 5.2), where we determine how datasets vary in covering appearance variability of a class; *Increase label specificity* (Sec. 5.3), where we can relabel instances of a class at a finer-grained level; *Predict transfer learning gains* (supp. mat. Sec. B), where our label relations can predict the gains brought by transfer learning.

2 Related Work

Dataset creation and evolution. In computer vision there is a long standing history to create datasets for training and benchmarking methods. There are too many to recall here, but interestingly many popular dataset have evolved over time, either by growing the number of images, like ImageNet [6], the number of classes, like PASCAL-VOC [7] from 4 classes in 2005 to 20 in 2007, or in the types of annotation, like COCO [17] to COCO-stuff [4] to COCO-panoptic [12]. Other datasets evolve by merging, for example the SUNRGB-D [35] dataset combined imagery from among others NYU-depth-v2 [25] and SUN3D [30], while the ADE20K [34] dataset contains imagery from SUN [29] and Places [35].

In this paper we use the COCO-panoptic dataset [5,12,17], ADE20K [34] and BDD [32]. Instead of considering these dataset individually, we explore how the visual concepts in these dataset *relate* to each other. The relations we find could be used when aiming to combine these datasets or when aiming to train more generic models across different datasets.

Learning over diverse image domains. Any single dataset has issues by its design [22], bias [26], or evaluation robustness [33]. Therefore a recent trend is to train or evaluate algorithms over multiple datasets. For example in the Robust Vision Challenge [1] participants are asked to evaluate a single trained model over multiple datasets and the winner is based on the average performance. To facilitate this, collection of datasets have been introduced, for example, Visual Decathlon for image classification [24], Meta-Dataset for few-shot learning [27], and MSeg for semantic segmentation [16].

Training tactics to successfully use multiple datasets differ, from training a single model with different heads over all datasets jointly [14], to learn in stages, *i.e.*, first on ImageNet, then tune on COCO and finally fine-tune on PASCAL-VOC [19]; and from using manually merged labels [2,16], to post-hoc merging of labels for detection [36]. In contrast to these approaches, our aim is not to train a new model with better classifiers, but we aim to analyze more fundamentally how datasets relate to each other.

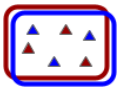
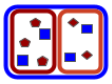
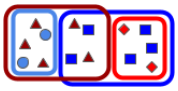
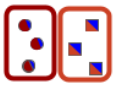
Identity Label a in one dataset indicates the same visual concept as label b in another dataset. For example `sofa` in ADE20k and `couch` in COCO represent the same visual concept.	
Parent/child A subcategory relationship. For example, `animal` in ADE20k is the parent of `cow` in COCO.	
Overlap Label a in one dataset and label b in another describe visual concepts which are not the same even though their sets of instances intersect. For example, the ADE20k `floor` and COCO `rug-merged` both describe a floor-covering carpet. Yet both concepts are broader in a mutually exclusive way: `floor` also includes a wooden floor which is not a `rug-merged`. Conversely, `rug-merged` also includes a rug which can be picked up which is not a `floor`.	
Part-of Label a in one dataset captures parts of instances of label b in another dataset. For example, `roof` in COCO describes part of an instance of `house` in ADE20k.	

Table 1: Definition of types of label relations we distinguish. In this paper we aim to automatically identify all relations (Sec. 3) except the *part-of* relation.

Zero-shot and open set segmentation. For both zero-shot and open set segmentation the goal is to obtain pixel-wise predictions for never-seen labels using zero training examples [3,9]. Both aim to learn classifiers which generalize the set of training classes to a fixed set of never seen labels [3] or open vocabulary queries [9]. This works by establishing (language) based relations between seen and unseen classes, for example based on large scale contrastive pre-training on images and textual queries [11,23]. In contrast to these methods, our aim is not to train generalizable classifiers, but to find the relations between visual concepts in both datasets, for which we can make use of the available annotations.

3 Method

In this paper we want to automatically discover relations between class labels across two given datasets A and B. We consider all possible pairs $\langle a, b \rangle$ of labels a in A and b in B. For each pair we want to determine if they are *related*, i.e. where there are visual instances which are covered by both the definition of a and b, and we also want to determine the *type* of the relation (Tab. 1).

We distinguish *identity*, *parent/child*, *overlap*, and *part-of* (focusing mostly on the first three). Importantly, the existence of a relation between two labels and its type cannot be derived simply by considering their names. Instead, they are specific to the pair datasets from which they originate, because they depend on the design and construction of each dataset.

3.1 Discovering relations using visual information

We first discuss how we discover relations and their type using purely visual information, as illustrated in Fig. 2). We do this in the context of semantic seg-

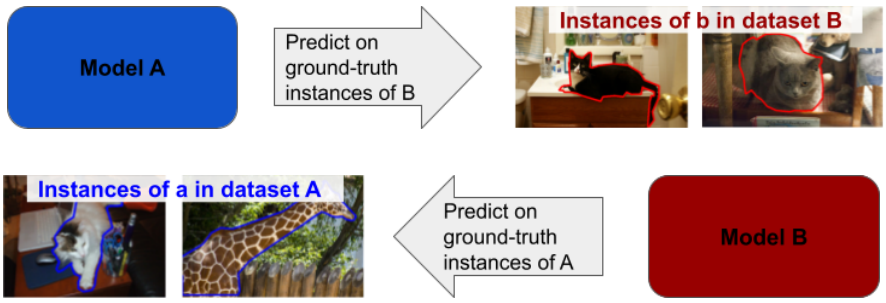


Fig. 2: Illustration of how to obtain label link scores between ADE20k `animal` and COCO `cat`, we estimate $S_{a\to b}(\texttt{animal}, \texttt{cat})$ using the model trained on ADE20K and $S_{b\to a}(\texttt{cat}, \texttt{animal})$ using the model trained on COCO.

mentation, but our method would also work for object detection. To determine whether there exist a relation between label a in dataset A and b in dataset B, we use annotated *instances*[1] of these classes in their respective datasets.

We use a model p_A trained on dataset A to obtain predictions $p_A(a|i_b)$ for label a for an instance i_b with label b from dataset B. Next, we average these predictions:

$$S_{a\to b} = \frac{1}{n_b} \sum_{i_b \in B} p_A(a|i_b) \tag{1}$$

where n_b is the number of instances of label b in dataset B. Intuitively, this measures how likely it is that the instances of $i_{b=\texttt{cat}}$ from the COCO dataset (B) would be called $a = \texttt{animal}$ according to the model trained on ADE20k (dataset A). Similarly we obtain $S_{b\to a}$ by aggregating predictions of p_B over instances of dataset A. The final score is the average: $R_{a,b} = (S_{a\to b} + S_{b\to a})/2$. To determine whether there is a relation between label a and label b, we simply threshold $R_{a,b}$. This results in a set $\mathcal{R}$ of binary relations.

Experimentally we evaluate two different prediction models $p_A(a|i_b)$:

- *Pixel Probabilities*: applying a segmentation model trained on dataset A directly on instances of dataset B. To convert to instance probabilities we average the pixel-wise probabilities over all pixels of the instance;
- *Visual Embeddings*: we extract instance visual features for both dataset A and dataset B by aggregating the pixel-wise visual features, using the same segmentation model (trained on dataset A) without the classification head. Then we use a 1-Nearest Neighbour classifier. This results in a binary prediction *i.e.* $p_A(a|i_b)$ is either 1 or 0. We do the analogue for $p_B(b|i_a)$.

Training details. We train semantic segmentation models using an HRNetV2-W48 [28] backbone with a linear pixel-wise prediction head and a softmax-loss. This results in a strong model for semantic segmentation [16,19,28]. We unify the training setup to make the models compatible across datasets, using color

[1] An instance is either a single object (for thing classes, e.g. `cat`, `car`), or the union of all regions of a stuff class (e.g. `grass`, `water`), following the panoptic definition [13].

normalization, horizontal flipping, random crop and resize to 713×713. We optimize using SGD with momentum, with lr $= 0.01$ decreased by a factor 10 after 2/3rd of the number of training steps (optimized per dataset).

While for semantic segmentation typically the `background` class is ignored during training and evaluation we find it useful to incorporate it explicitly. The `background` prediction can be interpreted as the model predicting *none of the classes from my label space*. Moreover, we find it beneficial to only aggregate over *easy* instances to factor out errors introduced by miss-classification of difficult instances. To do so we use instances which are classified correctly by the model trained on the same dataset. More specifically, we define instances to be easy for the pixel probability method if $p_B(b|i_b) > 0.5$. They are easy for the visual embedding method if $p_B(b|i_b) = 1$.

3.2 Relation type discovery

We estimate the *type* of relation (Tab. 1) in two different ways, one based on set theory and the other on the degree of asymmetry between $S_{a\to b}$ and $S_{b\to a}$.

Set theory. To derive the relation types we make two assumptions: (1) There is only a relation between label a and label b if there are instances which can be categorized as both a and b, so $\langle a, b\rangle \in \mathcal{R}$; (2) Labels from the same dataset are mutually exclusive. Then we derive the types between a_k and b_l as follows:

- *identity*: a_k and b_l have an identity relation when neither a_k nor b_l has a relation with another label. More formally, $\langle a_k, b_l\rangle \in \mathcal{R}$, but $\nexists a_m, \langle a_m, b_l\rangle \in \mathcal{R}, a_m \neq a_k$ and $\nexists b_n, \langle a_k, b_n\rangle \in \mathcal{R}, b_n \neq b_l$.
- *parent/child*: A label a_k is a parent if it is related to at least two labels in B (including b_l), which are not related to any other label in A. More formally, for at least two labels b_l and b_n, $b_l \neq b_n$, it holds that $\langle a_k, b_l\rangle \in \mathcal{R}$ and $\langle a_k, b_n\rangle \in \mathcal{R}$. Yet, $\nexists a_m, [\langle a_m, b_l\rangle \in \mathcal{R} \vee \langle a_m, b_n\rangle \in \mathcal{R}], a_m \neq a_k$. Analogously, a_k is a child of b_l if their roles are reversed.
- *overlap*: both labels a_k and b_l are used in multiple relations. Formally, $\langle a_k, b_l\rangle \in \mathcal{R}$ and $\exists a_m, \langle a_m, b_l\rangle \in \mathcal{R}, a_m \neq a_k$ and $\exists b_n, \langle a_k, b_n\rangle \in \mathcal{R}, b_n \neq b_l$.

Score Asymmetry. We exploit the asymmetry between $S_{a\to b}$ and $S_{b\to a}$ to provide the type of the relation. Intuitively, for a *parent-child* relation, we expect that an `animal` classifier gives high scores on `cat` instances, while the `cat` classifier only gives high scores on *some* of the `animal` instances. Therefore, a large asymmetry between $S_{a\to b}$ and $S_{b\to a}$ suggests that the labels are in a *parent-child* relation. Given a pair of labels $(a, b) \in \mathcal{R}$ we derive the label as follows: 1) a is a *parent* of b, if $\frac{S_{a\to b}}{S_{b\to a}} > T$; else 2) a is a *child* of b, if $\frac{S_{b\to a}}{S_{a\to b}} > T$; otherwise 3) a and b are in an *identity* relation. Note this method cannot predict *overlap*.

3.3 Predicting relation types using language

We introduce two baseline methods which use language to discover relations.

WordNet. We use the WordNet [21] taxonomy and its graphical structure. Specifically, we map each class label to a WordNet noun-synset. Then, if a and

b map to the same synset, they are in an *identity* relation. When the synset of a is an ancestor of the synset of b, then a is a *parent* of b. If two synsets share at least one descendant, they are in an *overlap* relation. For example, in WordNet `car` and `truck` overlap since they both have `minivan` as a descendent.

For each pair of labels (a,b) we estimate the `path similarity` between the two synsets, which is based on the proximity of their nearest common ancestor. Then we add 1 if a and b have a relation according to the taxonomy. This yields a dense matrix R, with pairs discovered as *identity* have a strength of 2, as *parent*, *child*, or *overlap* have a strength between 1 and 2, and the rest between 0 and 1.

Word2Vec. Our second baseline uses Word2Vec [20], based on the publicly available model trained on Wikipedia [10]. This maps each word to a 500-D embedding vector. The score between each paper of labels a, b is based on the cosine similarity between their embeddings. Since this is a symmetric similarity, we can only use the *set theory* method to determine relation types.

3.4 Discovering relations by combining vision and language

We combine our Visual Embeddings method with our WordNet method. We multiply the strength of the visual relation $R_{a,b}$ by a constant factor n if the synset of a and the synset of b are related according to the taxonomy (i.e. we discover *identity*, *parent*, *child*, or *overlap*).

To discover the relation type, we combine the visual *asymmetry* method and the WordNet predictions: If according to WordNet a and b are in an *identity* relation, we enlarge threshold T of the *asymmetry* method by a factor m. This makes it more likely that *identity* will be predicted. Similarly, when according to WordNet a and b are in a *parent*/*child* relation, we reduce T by a factor m.

3.5 Evaluation

To evaluate how well we are able to automatically discover relations between labels across datasets, we first establish ground-truth relations[2]. We leverage the MSeg dataset [16], who manually constructed a unified label space across a variety of different datasets, which we refer to as *MSeg labels* (Fig. 3). Based on this, we first map dataset A and dataset B to the MSeg label space, and then create direct relations between labels in A and B.

Establishing relations to the MSeg label space. The MSeg dataset provides for all dataset which they cover a new ground-truth in the unified MSeg label space. This MSeg ground-truth covers a different set of labels than the original ground-truth for each dataset; for each dataset the authors merged some classes and re-annotated other classes to obtain a consistent labeling of each dataset according to the MSeg label space [16]. We compare the MSeg ground-truth with the original ground-truth to establish relations.

In particular, we count how many times an instance with a particular label in the original label space is relabeled to each MSeg label. For an instance to

[2] Available at: https://github.com/google-research/google-research/tree/master/missing_link

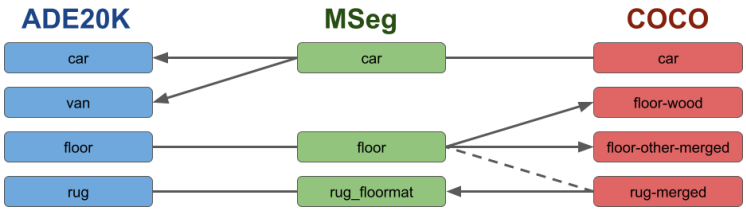


Fig. 3: To create relations between ADE20k and COCO, we first establish relations between each individual dataset and MSeg. Based on set theory (Sec. 3.1) we establish *identity* (—), *parent/child* (→), and *overlap* (- -) relations. Afterwards, through MSeg we derive direct relations between ADE20k and COCO.

count being relabeled, more than 50% of its pixels need to have been relabeled. This makes the process robust against small manual corrections made during the MSeg relabeling effort. We manually inspect all label pairs with a positive count, and remove them when this is caused by a human error (low counts typically help identify these cases). For example, a few instances of COCO `tent` have been relabeled to the MSeg `kite`, while these labels are clearly unrelated. All remaining pairs are considered as related in our ground-truth.

To derive the type of relation, we apply the *set theory* method from Sec. 3.1, and then manually investigate all relations. We found that almost no human correction was needed at this stage. The only exception was that a few relations were changed to *part-of*, which the set theory method cannot automatically produce. For example, COCO `roof` is *part-of* the MSeg `building`.

Establishing relations between A and B. Through the MSeg labels, we can directly relate the original labels between datasets (Fig. 3). The type of relation depends on the type of the two individual relations with MSeg. When both relations are *identity*, the resulting relation is that as well. Two consecutive *child* relations or one *identity* and one *child* relation result in a *child* relation. For example, ADE20k `van` is a *child* of COCO `car`. The *parent* relation is analogous to *child*. If one relation is *part-of*, the resulting relation is *part-of* as well. For all other cases, we manually inspect visual examples to determine the relation type. Often this happens for *overlap* relations. But for example both COCO `person` and the ADE20k `person` have been sub-categorized by MSeg in `person`, `bicyclist`, `motorcyclist`, and `rider_other`. It requires manual inspection to verify that both `person` labels represent the same concept and hence have an *identity* relation. As before, after these steps we perform a final quality control by manually inspecting visual examples of label pairs.

Quantitative evaluation. For two datasets, we compare our automatically predicted relations with the ground-truth we just established. We evaluate how good our methods are in predicting whether any relation is present, regardless of its type. To do so we order all possible label pairs according to their predicted strength, and calculate a Precision-Recall (PR) curve and its associated Area Under the Curve (AUC). We also measure how well our methods predicts relation types, where we also consider *no relation* predictions. We measure accuracy for each predicted type and average them to obtain an overall accuracy.

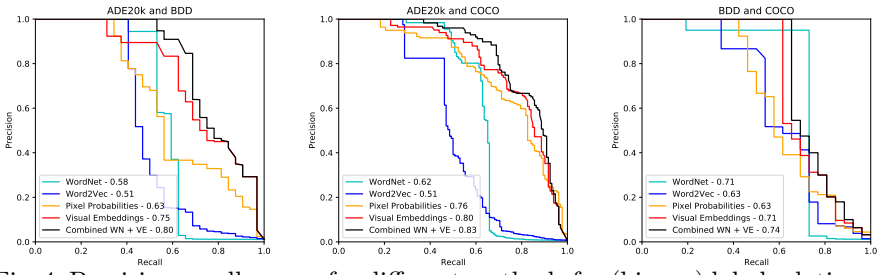


Fig. 4: Precision recall curves for different methods for (binary) label relation predictions. The visual methods perform (much) better than language-only methods and combining vision and language gives best performance.

4 Results

As our main experiment we apply and evaluate our method on all three possible pairs of the following semantic segmentation datsets: (1) ADE20k [34], a dataset of consumer photos, where we consider the 150 most frequent class labels as is common practice; (2) COCO Panoptic [5,17,13], which also contains consumer photos, with 133 classes; (3) Berkeley Deep Drive [32], a driving dataset containing 19 classes. For ease of exposition, we write the names of class labels for ADE20k in `blue`, for COCO in `red`, and for BDD in `violet`.

Additionally, in Sec. A of the supp. mat. we demonstrate that we can establish relations between labels of different types of datasets by applying our method to ILSVRC classification and COCO segmentation.

Relation discovery. The Precision-Recall curves in Fig. 4 show that the language-based models are generally outperformed by the vision-based models. The model based on WordNet works better than Word2Vec, because Word2Vec gives high scores for labels which are semantically related but do not refer to the same object. For example, the Word2Vec cosine similarity between `shower` and `toilet` is 0.72 while these classes are really disjoint. The WordNet-based method has high accuracy for labels in an *identity* or *parent/child* relation according to the taxonomy, but a low recall for many other relations. Among the vision models, the Visual Embeddings method consistently outperforms the Pixel Probability method (Sec. 3.1). Finally, we obtain the best performance when combining WordNet with Visual Embeddings.

Relation type classification. Before we can determine relation types, we note that the *Set Theory* and *Score Asymmetry* methods have thresholds (Sec. 3.2). We establish these by optimizing accuracy with respect to the predictions made by the WordNet *taxonomy*. While the WordNet *taxonomy* method may not be fully accurate, as long as it is an unbiased estimate its optimal thresholds will also hold for the real ground-truth - which is indeed what we found.

The results in Tab. 2 generally align with our previous observations: WordNet is the best language-based model but the vision-based models work even better. Again, the Visual Embeddings method outperforms all others. From the two ways to determine the relation type, the one based on *Score Asymmetry* works best. Intuitively, it makes sense that this is a powerful mechanism: we expect an

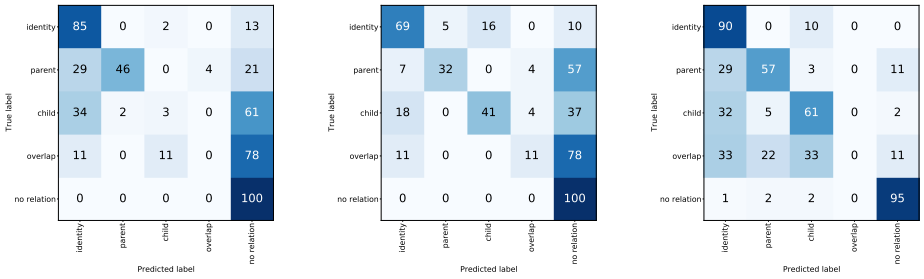


Fig. 5: Confusion matrices for relation types between ADE20K and COCO using WordNet-taxonomy (*left*), Embeddings with Set Theory (*middle*), and Embedding with Asymmetry (*right*).

	Language		Vision				Vision+Language
	WordNet	Word2Vec	Pixel Predictions		Visual Embeddings		WordNet+Embeddings
	taxonomy	*set theory*	*set theory*	*asymmetry*	*set theory*	*asymmetry*	*taxonomy+asymmetry*
ADE20k, BDD	46	37	56	54	56	55	57
ADE20k, COCO	47	38	47	60	51	61	62
BDD, COCO	46	38	46	48	49	51	53
average	46	38	50	54	52	56	57

Table 2: Accuracy (in percentage) of estimating relation types. Our vision-based models outperform language-only models for all pairs of datasets. and combining works best.

`animal` model to always yield high scores on `cat` instances, whereas a `cat` model will not give high scores to all `animal` instances. As before, the combination of WordNet and Visual Embeddings gives the best results.

In Fig. 5 we show the full confusion matrices for relations between ADE20k and COCO discovered by the WordNet *taxonomy*, Visual Embeddings with *Set Theory*, and Embeddings with *Asymmetry* methods. We can see that the WordNet taxonomy predicts both the identity and 'no relation' pretty well, but tends to over-predict 'no relation'. The Embeddings with *Set Theory* also over-predicts 'no relation', it is slightly worse in 'identity' but better in 'child' and 'overlap'. Embeddings with *Asymmetry* is significantly better in parent and child relations. However, it cannot predict 'overlap'.

5 Applications

We apply our method to four applications (Sec. 5.1, 5.2, 5.3, supp. mat Sec. B).

5.1 Understand Label Relations

To gain insights into why and how labels relate, we visually inspect instances of labels with high-scoring relations, but whose labels do not exactly match. We do this for relations between ADE20k and COCO which we visualize in Fig. 6.

Fig. 6: Examples of instances of classes in ADE20k (in blue) and COCO (in red). The top rows shows examples for labels for which we find a relation. The second row shows how stove and oven categorize the visual world differently. The bottom shows different types of water which are difficult to distinguish, and different types of screens which are labeled inconsistently.

Identity. One of the highest scoring identity relations with non-matching labels is the ADE20k sofa and COCO couch. These are synonyms and indeed represent the same visual concept (see Fig. 6 *top left*). More interestingly, we also identify a relation between ADE20k minibike and COCO motorcycle. Semantically these are different concepts: usually a minibike denotes a tiny motorbike which

is not higher than one's knees. But here both represent a full-sized motorcycle (Fig. 6 *top center*). Finally, another interesting, high-scoring identity relation we found is between `stove` and `oven` (Fig. 6 *second row*). In ADE20k the `stove` refers to the cooking panel on which you can put pots and pans, while including the oven underneath if it exists. In COCO, the `oven` refers to the closed heating compartment, including the stove if it exists. So even while `stove` and `oven` are synonyms and mostly represent the same visual concept, one could argue that the true relation is not identity but overlap, because there are instances which are `stove` but not `oven` (2nd row, left) and vice-versa (2nd row, right).

Parent/child. One example of parent/child is between ADE20k `animal` and COCO `elephant` (Fig. 6 *top-right*). Others include `hill` and `mountain-merged`, `wall` and `wall-tile`. We also correctly identify that the ADE20k `tent` is a child of the COCO `tent`, since the latter also includes the ADE20k `awning`. Language alone would never be able to identify that `tent` and `tent` have a *child* relation.

Overlap. Here we look at several overlap relations found by our embeddings and logic method. It correctly identifies the overlap between `floor` and `rug-merged`. This overlap relation exists because both labels use a different reference frame of the world: `floor` emphasizes that the concept is *stuff* and not an *object*, while `rug-merged` emphasizes the function and type of material (*e.g.* fabric to walk on), see Fig. 1. We also predict an overlap relation between `water` and `water-other`, where the ground-truth relation is *child*. When visually inspecting examples, we found many examples where it was unclear what type of water the image depicts (Fig. 6 *bottom left*). Arguably, `water`, `river`, `sea`, `river`, and `water-other` all overlap, mostly caused by the visual ambiguity in images with these labels.

Inconsistencies. Finally, we found strong relations not only between `television receiver` and `tv`, but also between `crt screen`, `monitor`, `computer` and `tv`. Looking at instances, these labels often point to the same visual concepts (Fig. 6 *bottom right*). So strictly speaking, these labels visually overlap. However, this overlap is caused by labeling errors and inconsistencies in both datasets. In COCO, all displays (including computer monitors) are labeled as `tv`. Instead, in ADE20k computer monitors are alternatively labeled as `crt screen`, `monitor`, and `computer`. These concepts overlap even within ADE20k which makes the common assumption of mutually exclusive labels *within a dataset* invalid.

5.2 Identify Missing Aspects

We want to identify which appearance aspects of a class are common between two datasets, and which are covered by only one of them. Discovering this would enable combining examples from different datasets to cover the full range of visual appearances of a class. This could help train better recognition models.

For this experiment we focus on the COCO `car` and BDD `car` classes. We use the model trained on the COCO dataset and extract features for both COCO `car` and BDD `car` instances, which we aggregate per instance. We use these features to create a 2D visualisation using UMAP [18] in Fig. 7 and extract 6 different clusters for further analysis in Fig. 8.

Each of the shown clusters has some particular visual coherence, for example:

Fig. 7: Embedding of `car` instances (COCO with blue box, BDD with red).

Fig. 8: Embedding with clusters of COCO `car` (orange dots) & BDD `car` (blue dots) instances (*left*) and example images sampled from each cluster (*right*).

- Three clusters of cluttered streets differing in the shapes: the back of the car in the center, partial cars at the image border, and partial occluded cars.
- Two clusters with imagery captured at night, but with different instance shapes. Those clusters are mostly filled with images from the BDD dataset.
- A cluster with *parked cars next to sports*, filled with only COCO images.

From this visual analysis we observe that there is a significant overlap in the kind of `car` segments: both datasets contain instances with a rear or side view, partially occluded instances, and instances at the edge of the image. However, we also find interesting differences in the imagery contained in the datasets: BDD is a driving dataset and hence the diversity in viewpoints of scenes is limited to the viewpoint from the dashboard. COCO, on the other hand, is a very diverse consumer dataset, where street imagery is present with much more viewpoints. That explains why we see cars near sport fields in COCO, but not in BDD.

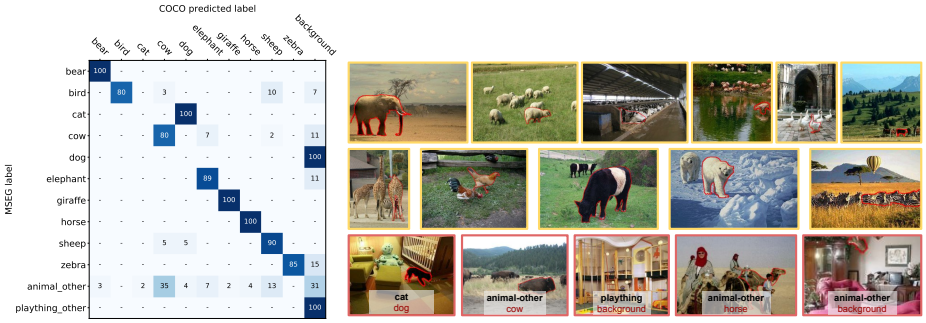


Fig. 9: Confusion matrix (*left*) and example (*right*) evaluating re-annotation of ADE20k `animal` using the predictions of related child COCO classes.

5.3 Increase Label Specificity

In this experiment we illustrate how the discovered label relations could be used to annotate images with a finer level of annotation. Here we relabel the ADE20k `animal` instances into the related COCO classes: {`cow`, `dog`, ..., `zebra`} using the model trained on COCO, using the established label relation indicating that ADE20k `animal` is a parent class of these COCO classes.

For this experiment we use the model trained on COCO and use this model to predict fine-grained annotations on the instances belonging to the ADE20k `animal` class. In order to quantitatively evaluate these new annotations we make use of the MSeg annotations. These provide ground-truths for the ADE segments, which we use to evaluate the top-1 accuracy per class.

Fig. 9 (*left*) shows the confusion matrix between MSeg ground-truths and COCO predictions on ADE instances. Fig. 9 (*right*) shows examples of correctly and incorrectly classified segments. From the results we observe that for most labels the finer annotations are accurate and the errors are easily explainable.

6 Conclusion

In this paper we investigated the relations of labels across datasets. We introduced several methods to automatically discover relations and their types. Our experiments showed that our vision-based models outperformed our language-based models by a significant margin, demonstrating that relying on the semantics of the label names alone is insufficient for establishing such relationships.

We demonstrated the usefulness of establishing *visual-semantic* relationships on four applications. Among our findings, we discovered that the definition of labels across datasets can vary in subtle ways. Understanding these subtle relations is important when using multiple datasets, such as when training on a combination of datasets, when fine-tuning on a target dataset, or when merging two datasets. We hope that our work inspires more researchers to study how different datasets relate to each other and how to exploit these relations to address computer vision problems.

References

1. Robust vision challenge. http://www.robustvision.net/
2. Bevandić, P., Oršić, M., Grubišić, I., Šarić, J., Šegvić, S.: Multi-domain semantic segmentation with overlapping labels. In: Proc. WACV (2022)
3. Bucher, M., Vu, T., Cord, M., Pérez, P.: Zero-shot semantic segmentation. In: NeurIPS (2019)
4. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff dataset. http://calvin.inf.ed.ac.uk/datasets/coco-stuff (2018)
5. Caesar, H., Uijlings, J., Ferrari, V.: COCO-Stuff: Thing and stuff classes in context. In: CVPR (2018)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-fei, L.: ImageNet: A large-scale hierarchical image database. In: CVPR (2009)
7. Everingham, M., Eslami, S., van Gool, L., Williams, C., Winn, J., Zisserman, A.: The PASCAL visual object classes challenge: A retrospective. IJCV (2015)
8. Geiger, A., Lenz, P., Stiller, C., Urtasun, R.: Vision meets robotics: The KITTI dataset. International Journal of Robotics Research (2013)
9. Ghiasi, G., Gu, X., Cui, Y., Lin, T.: Open-vocabulary image segmentation. Tech. rep., ArXiV (2021)
10. Google: Wiki words 500 with normalization - a 500 dimensional wor2vec skip-gram model trained on english wikipedia. https://tfhub.dev/google/Wiki-words-500-with-normalization/2
11. Jia, C., Yang, Y., Xia, Y., Chen, Y., Parekh, Z., Pham, H., Le, Q.V., Sung, Y., Li, Z., Duerig, T.: Scaling up visual and vision-language representation learning with noisy text supervision. In: ICML (2021)
12. Kirillov, A.: Panoptic challenge intro. COCO+Mapillary Joint Recognition Challenge Workshop, http://presentations.cocodataset.org/ECCV18/COCO18-Panoptic-Overview.pdf
13. Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P.: Panoptic segmentation. In: CVPR (2019)
14. Kokkinos, I.: Ubernet: Training a 'universal' cnn for low-, mid-, and high- level vision using diverse datasets and limited memory. In: CVPR (2017)
15. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Duerig, T., Ferrari, V.: The Open Images Dataset V4: Unified image classification, object detection, and visual relationship detection at scale. IJCV (2020)
16. Lambert, J., Liu, Z., Sener, O., Hays, J., Koltun, V.: MSeg: A composite dataset for multi-domain semantic segmentation. In: CVPR (2020)
17. Lin, T.Y., Maire, M., Belongie, S., Bourdev, L., Girshick, R., Hays, J., Perona, P., Ramanan, D., Zitnick, C.L., Dollár, P.: Microsoft COCO: Common objects in context. In: ECCV (2014)
18. McInnes, L., Healy, J., Saul, N., Grossberger, L.: UMAP: Uniform manifold approximation and projection. The Journal of Open Source Software (2018)
19. Mensink, T., Uijlings, J., Kuznetsova, A., Gygli, M., Ferrari, V.: Factors of influence for transfer learning across diverse appearance domains and task types. IEEE Trans. on PAMI (2021)
20. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. In: ICLR workshop (2013)
21. Miller, G.: WordNet: a lexical database for English . Communications of the ACM **38**(11), 39–41 (1995)

22. Ponce, J., Berg, T.L., Everingham, M., Forsyth, D.A., Hebert, M., Lazebnik, S., Marszalek, M., Schmid, C., Russell, B.C., Torralba, A., Williams, C.K.I., Zhang, J., Zisserman, A.: Dataset issues in object recognition. In: Toward Category-Level Object Recognition (2006)
23. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: ICML (2021)
24. Rebuffi, S.A., Bilen, H., Vedaldi, A.: Learning multiple visual domains with residual adapters. In: NeurIPS (2017)
25. Silberman, N., Hoiem, D., Kohli, P., Fergus, R.: Indoor segmentation and support inference from rgbd images. In: ECCV (2012)
26. Torralba, A., Efros, A.: An unbiased look on dataset bias. In: CVPR (2011)
27. Triantafillou, E., Zhu, T., Dumoulin, V., Lamblin, P., Evci, U., Xu, K., Goroshin, R., Gelada, C., Swersky, K., Manzagol, P., Larochelle, H.: Meta-dataset: A dataset of datasets for learning to learn from few examples. In: ICLR (2020)
28. Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W., Xiao, B.: Deep high-resolution representation learning for visual recognition. IEEE Trans. on PAMI (2020)
29. Xiao, J., Hays, J., Ehinger, K., Oliva, A., Torralba, A.: SUN database: Large-scale scene recognition from Abbey to Zoo. In: CVPR (2010)
30. Xiao, J., Owens, A., Torralba, A.: SUN3D: A database of big spaces reconstructed using SfM and object labels. In: ICCV (2013)
31. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020)
32. Yu, F., Chen, H., Wang, X., Xian, W., Chen, Y., Liu, F., Madhavan, V., Darrell, T.: Bdd100k: A diverse driving dataset for heterogeneous multitask learning. In: CVPR (2020)
33. Zendel, O., Honauer, K., Murschitz, M., Humenberger, M., Fernandez Dominguez, G.: Analyzing computer vision data - the good, the bad and the ugly. In: CVPR (2017)
34. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ADE20K dataset. In: CVPR (2017)
35. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In: NeurIPS (2014)
36. Zhou, X., Koltun, V., Krähenbühl, P.: Simple multi-dataset detection. In: CVPR (2022)