

Recovering the Missing Link: Predicting Class-Attribute Associations for Unsupervised Zero-Shot Learning

Ziad Al-Halah

Makarand Tapaswi

Rainer Stiefelhagen

Karlsruhe Institute of Technology, 76131 Karlsruhe, Germany

{ziad.al-halah, makarand.tapaswi, rainer.stiefelhagen}@kit.edu

Abstract

Collecting training images for all visual categories is not only expensive but also impractical. Zero-shot learning (ZSL), especially using attributes, offers a pragmatic solution to this problem. However, at test time most attribute-based methods require a full description of attribute associations for each unseen class. Providing these associations is time consuming and often requires domain specific knowledge. In this work, we aim to carry out attribute-based zero-shot classification in an unsupervised manner. We propose an approach to learn relations that couples class embeddings with their corresponding attributes. Given only the name of an unseen class, the learned relationship model is used to automatically predict the class-attribute associations. Furthermore, our model facilitates transferring attributes across data sets without additional effort. Integrating knowledge from multiple sources results in a significant additional improvement in performance. We evaluate on two public data sets: *Animals with Attributes* and *aPascal/aYahoo*. Our approach outperforms state-of-the-art methods in both predicting class-attribute associations and unsupervised ZSL by a large margin.

1. Introduction

Large-scale object classification and visual recognition have seen rising interest in the recent years. Data sets such as ImageNet [10] have helped scale up the number of classes represented in tasks such as object classification or detection. Many methods based on deep convolutional neural networks [22, 39] have been developed recently to leverage the power of millions of training images distributed among thousands of classes. However, building a large data set, especially collecting large number of training images is very challenging and nevertheless this ends up representing only a fraction of the real visual world [7].

Transfer learning is a practical solution to bridge this gap as it allows to leverage knowledge and experience obtained from existing data to new domains. Specifically for object classification, knowledge from object categories which have labeled image samples can be transferred to new unseen cat-

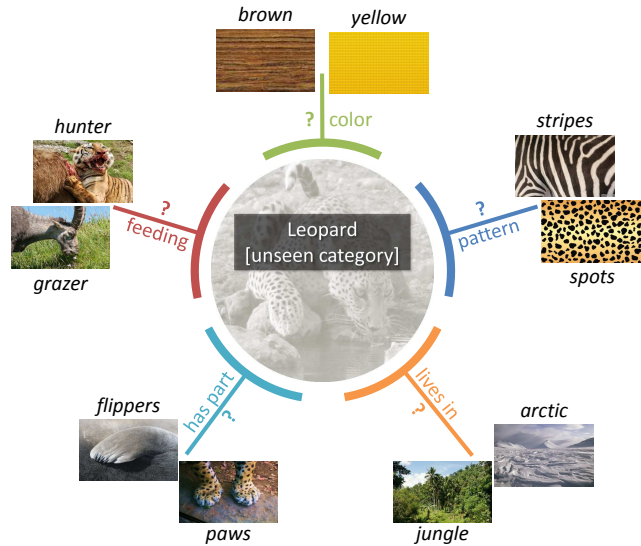


Figure 1: Given only the name of an unseen category, here *Leopard*, our method automatically predicts the list of attributes (e.g. yellow, spots) associated with the class through relationships (e.g. has_color, has_pattern). These predicted associations are leveraged to build category classifiers for zero-shot learning.

egories which do not have training images. This task is referred to as zero-shot learning (ZSL).

There exist many directions in the literature to perform ZSL. These primarily differ in the type of knowledge source they tap in order to establish the connection between the unseen classes and the available visual information [6, 23, 33]. Among these directions, attribute-based knowledge transfer shows impressive performance [4, 23, 25]. By learning an intermediate layer of semantic attributes (e.g. colors or shapes), a novel class is then described with a subset of these attributes and its model is constructed accordingly based on the respective attribute classifiers.

A major drawback of attribute-based approaches is that user supervision is needed to provide the description for each novel class. For example, for the new class “leopard” the user needs to describe it with a set of visual attributes in order to establish the semantic link with the learned visual vocabulary (e.g. the leopard has part paws, it exhibits

a spotted pattern but does not live in water). This amounts to providing manual class-attribute associations in the range of tens [15, 23] to hundreds [41] of attributes for each new category. This is not only time consuming but often also requires domain-specific or expert knowledge [23, 41] that the user is unlikely to have. It is more convenient and intuitive for the user to provide just the name of the unseen class rather than a lengthy description.

Our goal is to remove this need for attribute supervision when performing zero-shot classification. We aim to automatically link a novel category with the visual vocabulary and predict its attribute association without user intervention. Thereby, we answer questions such as: Does the leopard live in the jungle? Does it have a striped pattern? (see Fig. 1). To this end, we propose a novel approach that learns semantic relations and automatically associates an unseen class with our visual vocabulary (*i.e.* the attributes) based solely on the class name. Using the predicted relations, we are able to construct a classifier of the novel class and conduct unsupervised zero-shot classification. Moreover, we demonstrate that our model is even able to automatically transfer the visual vocabulary itself across data sets which results in significant performance improvements at no additional cost. We demonstrate the effectiveness of such a model against state-of-the-art via extensive experiments.

2. Related work

In ZSL the set of train and test classes are disjoint. That is, while we have many labeled samples of the train classes to learn a visual model, we have never observed examples of the test class (a.k.a. unseen class). In order to construct a visual model for the unseen class, we first need to establish its relation to the visual knowledge that is obtained from the training data. One of the prominent approaches in the literature is attribute-based ZSL. Attributes describe visual aspects of the object, like its shape, texture and parts [16]. Hence, the recognition paradigm is shifted from labeling to describing [9, 35, 36]. In particular, attributes act as an intermediate semantic representation that can be easily transferred and shared with new visual concepts [12, 15, 23]. In ZSL, attributes have been used either directly [15, 23, 25], guided by hierarchical information [4], or in transductive settings [18, 32].

However, most attribute-based ZSL approaches rely on the underlying assumption that for an unseen class the complete information about attribute associations are manually defined [15] or imported from expert-based knowledge sources [23, 41]. This is a hindering assumption since the common user is unlikely to have such a knowledge or is simply unwilling to manually set hundreds of associations for each new category.

Towards simplifying the required user involvement,

given an unseen class [42] reduces the level of user intervention by asking the operator to select the most similar seen classes and then inferring its expected attributes. [26, 34] go a step further and propose an unsupervised approach to automatically learn the class-attribute association strength by using text-based semantic relatedness measures and co-occurrence statistics obtained from web-search hit counts. However, as web data is noisy, class and attribute terms can appear in documents in different contexts which are not necessarily related to the original attribute relation we seek. We demonstrate in this work, that the class-attribute relations are complex and it is hard to model them by simple statistics of co-occurrence.

In an effort to circumvent the need for manually defined associations, [5, 11] propose to extract pseudo attributes from Wikipedia articles using TF-IDF based embeddings to predict the visual classifier of an unseen class. In theory, an article can be extracted automatically by searching for a matching title to the class name. However, in practice manual intervention is needed when there is no exact match or the article is titled with a synonym or the scientific name of the category as reported by [11].

In a different direction, unsupervised ZSL can be conducted by exploiting lexical hierarchies. For example, [33] uses WordNet [28] to find a set of ancestor categories of the novel class and transfer their visual models accordingly. Likewise, [4] uses the hierarchy to transfer the attribute associations of an unseen class from its seen parent in the ontology. In [3], WordNet is used to capture semantic similarity among classes in a structured joint embedding framework. However, categories that are close to each other in the graph (*e.g.* siblings) often exhibit similar properties to their ancestors making it hard to discriminate among them. Moreover, ontologies like WordNet are not complete. Many classes (*e.g.* fine-grained) are not present in the hierarchy.

Recently, [17, 37] proposed to learn a direct embedding of visual features into the semantic word space of categories. They leverage a powerful neural word embedding [19, 27] that is trained on a large text corpus, and learn a mapping from the space of visual features to the word representation. At test time, they predict an image class by looking for the nearest category embedding to the one estimated by the neural network. [17] shows impressive results of this approach for large-scale ZSL. [30] improves upon [17] by considering a convex combination of word embeddings weighted by classifiers confidences to estimate the unseen class embedding. However, we show in our evaluation that such word embedding approaches are less discriminative than their attribute-based counterpart.

We propose an approach that goes beyond using web statistics, predefined ontologies and word embedding estimation. We provide an automatic framework to learn complex class-attribute relations and effectively transfer knowledge across domains for unsupervised zero-shot learning.

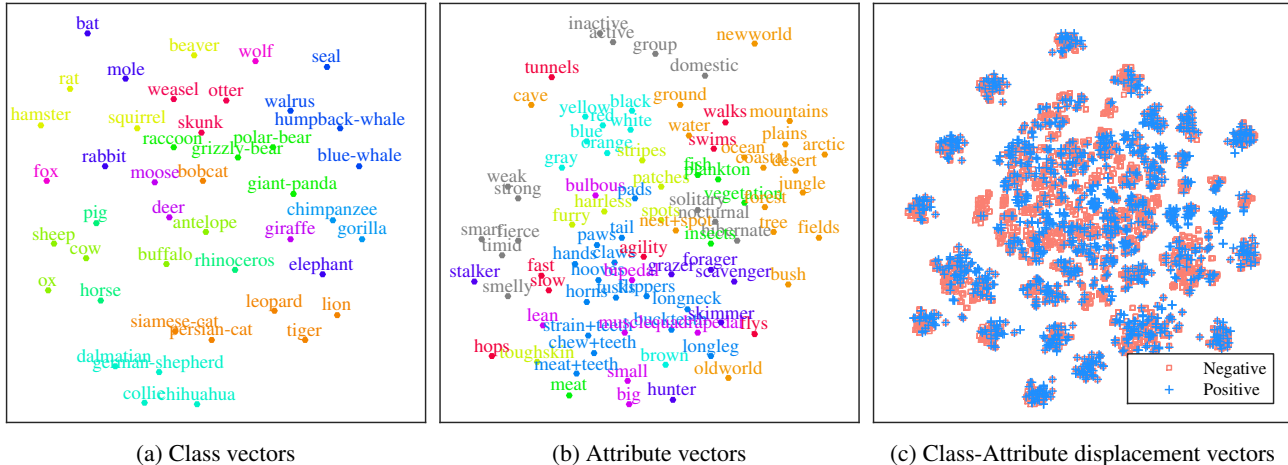


Figure 2: t-SNE representation of (a) class embeddings: colors indicate similar classes based on the super category in the WordNet hierarchy (e.g. dalmatian, collie, and other dog breeds are all colored in cyan); (b) attribute embeddings: colors indicate attributes which are grouped together to form class-attribute relations (e.g. has_color relationship clusters all colors yellow, black, etc. which are represented in cyan); and (c) class-attribute pair-wise displacement vectors (e.g. $v(\text{dolphin}) - v(\text{ocean})$) which show that encoding relationships using vector operations is a difficult task. This figure is best viewed in color.

3. Approach

We present an end-to-end approach to automatically predict class-attribute associations and use them for zero-shot classification. We begin by (i) finding suitable vector representations for words and use the learned embedding as a way to mathematically relate class and attribute names. These representations form the basis to model semantic relationships between classes and attributes. (ii) We formulate the learning of these relations in a tensor factorization framework (see Fig. 3) and offer key insights to adapt such a model to our problem. Finally, (iii) for an unseen class we show how to predict the set of its most confident attribute associations and carry out zero-shot classification. We start by defining the notation used throughout this paper.

Notation Let $\mathcal{C} = \{c_k\}_{k=1}^K$ be a set of seen categories that are described with a group of attributes $\mathcal{A} = \{a_m\}_{m=1}^M$. The vector representation of a word is denoted by $v(\cdot)$, and we use $v(c_k)$ and $v(a_m)$ for class c_k and attribute a_m respectively. The categories and attributes are related by a set of relations $\mathcal{R} = \{r_j\}_{j=1}^N$ such that $r_j(c_k, a_m) = 1$ if c_k is connected to a_m by relation r_j and 0 otherwise (e.g. $\text{has_color}(\text{sky}, \text{blue}) = 1$). Given only the name of an unseen class $z \notin \mathcal{C}$, our goal is to predict the attributes that are associated with the class (e.g. $\text{has_color}(\text{whale}, \text{blue}) = ?$) and conduct ZSL accordingly.

3.1. Vector space embedding for words

In order to model the relations between classes and attributes, we require a suitable representation that transforms names to vectors while at the same time preserves the semantic connotations of the words. Hereof, we use the skip-gram model presented by Mikolov *et al.* [27] to learn vector

space embeddings for words. The skip-gram model is a neural network that learns vector representations for words that best help in predicting the surrounding words. Therefore, words that appear in a similar context (neighboring words) are represented with vectors that are close to each other in the embedding space.

Fig. 2 visualizes the obtained word vector representation for few classes and attributes in our data set using t-SNE [40]. Even in such a low-dimension it is clear that classes related to each other appear closer. This is evident for example from the group of dog breeds or feline in Fig. 2a. Similarly, we also see clusters in the attribute label space corresponding to colors, animal parts, and environment (see Fig. 2b).

Relations in embedding space The skip-gram embeddings have gained popularity owing to their power in preserving useful linguistic patterns. An empirical evaluation [27] shows that syntactic/semantic relations can be represented by simple vector operations in the word embedding space. A great example is $v(\text{king}) - v(\text{man}) + v(\text{woman}) \approx v(\text{queen})$, where $v(\text{king})$ is the embedding for “king”. In other words, the relation between “king” and “man” modeled by their displacement vector is similar to the displacement between “queen” and “woman”.

However, modeling class-attribute relations by simple vector operations is inadequate. Fig. 2c presents the t-SNE representation for *displacement vectors* between each class-attribute pair (e.g. $v(\text{sky}) - v(\text{blue})$). We see that displacement vectors for both positive existing relations *and* negative non-existing relations are inseparable. We empirically show in Sec. 4.1 that class-attribute relations are more complicated and are not easily represented by simple vector operations.

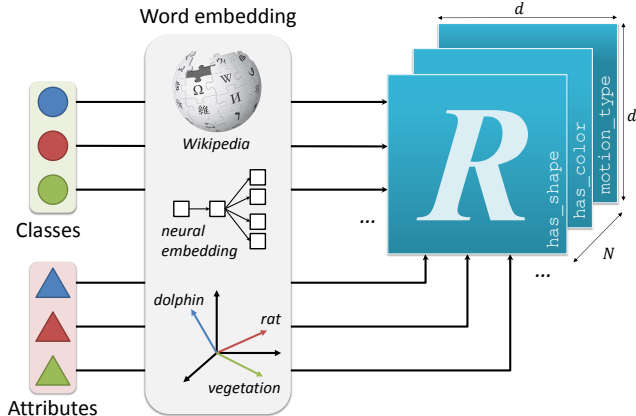


Figure 3: Our model couples class and attribute embeddings using the tensor \mathbf{R} . Each slice \mathbf{R}_j captures a relationship like `has_shape` or `motion_type`. The embeddings are obtained from a neural network trained on a large text corpus.

To address this challenge we adopt a more sophisticated and comprehensive method to learn these relations while at the same time effectively exploit the powerful word embedding representation.

3.2. Learning class-attribute relations

We now model the complex relations between categories and their corresponding visual attributes. Leveraging information based on these relations, we can predict the associations between a novel unseen class and our attribute vocabulary and build the corresponding ZSL classifier.

We propose to model the class-attribute relations using a tensor factorization approach [29, 38]. We represent the relations using a three dimensional tensor $\mathbf{R} \in \mathbb{R}^{d \times d \times N}$ where d is the dimension of the word embedding and N the number of relations (see Fig. 3). Each slice $\mathbf{R}_j \in \mathbb{R}^{d \times d}$ in the tensor models a relation r_j (e.g. `has_color`) as a bilinear operator. The likelihood of class c_k being associated with attribute a_m through relation r_j is:

$$p(r_j(c_k, a_m)) = \sigma(v(c_k)^T \mathbf{R}_j v(a_m)), \quad (1)$$

where $v(x) \in \mathbb{R}^d$ is the vector embedding of word x and $\sigma(\cdot)$ is the logistic function. We learn \mathbf{R} by minimizing the negative log-likelihood of both positive (\mathcal{P}) and negative (\mathcal{N}) class-attribute associations for each slice \mathbf{R}_j :

$$\min_{v(\mathcal{A}), \mathbf{R}_j} - \sum_{(j,k,m) \in \mathcal{P}} \log(p(t_{k,m}^j = 1)) - \sum_{(j,k,m) \in \mathcal{N}} \log(p(t_{k,m}^j = 0)),$$

where $t_{k,m}^j = r_j(c_k, a_m)$ (2)

Note that there are two key components in Eq. 2. Firstly, we take advantage of the powerful representation of skip-gram and learn word embeddings on a large text corpus to initialize the embeddings of our class ($v(\mathcal{C})$) and attribute

($v(\mathcal{A})$) entities. This gives our model the ability to generalize well to unseen classes and take advantage of the initial learned similarities among the attributes. Secondly, in our case of zero-shot classification, the novel class name is not available during training and we have no information about how this unseen class is related with the visual attributes. Consequently, we treat the set of categories as an *open* set and fix their embedding $v(\mathcal{C})$ to the one learned in Sec. 3.1. On the other hand, visual attributes \mathcal{A} are usually restricted to entities which we have seen before, and for which we have training images and learned models. This allows us to propagate gradients to $v(\mathcal{A})$ during training and optimize the attributes embeddings which yields improved performance (see model analysis in Sec. 4.2).

Limited training data Learning \mathbf{R} directly from training data is not favorable since the number of class-attribute associations available for training are usually small. For example, a typical data set consisting of 40 categories and 80 attributes yields around 1500 positive associations compared to tens or even hundreds of thousands of parameters in \mathbf{R} . Hence, in order to avoid overfitting we build on the ideas of [20] and reduce the number of parameters that are required to be learned, by representing the relation operator \mathbf{R}_j as a combination of L latent factors:

$$\mathbf{R}_j = \sum_{l=1}^L \alpha_l^j \Theta_l, \quad \alpha^j \in \mathbb{R}^L \text{ and } \Theta_l \in \mathbb{R}^{d \times d}, \quad (3)$$

where α^j is a sparse vector used to weight the contributions of the rank one latent factors Θ . Both α and Θ are learned while minimizing Eq. 2 and constraining $\|\alpha^j\|_1 \leq \lambda$. The parameter λ controls the sparsity of α , and hence the extent to which latent factors are shared across relations. Modeling \mathbf{R} with latent factors has the benefit of allowing the learned relations to interact and exchange information through Θ and hence improves the ability of the model to generalize.

Type of relations In order to train our model, we need to define the relations that link classes with the respective attributes. Usually these relations are harvested through the process of collecting and annotating attributes (e.g. `what color is a bear?` `what shape is a bus?`). We refer to this type of relations as *semantic relations*. However, while some data sets do provide such relation annotations [15, 41] others do not [24]. An alternative approach to manual annotation is to automatically discover relations by utilizing the word embedding space. As described earlier in Sec. 3.1, embeddings of semantically related entities tend to be close to each other (see Fig. 2b). Hence, one can simply group attributes into several relations by clustering their embeddings (i.e. N = number of clusters). We refer to this type of relations as *data driven relations*.

3.3. Predicting binary associations

Given an unseen class z , we predict its associations with the attribute set \mathcal{A} :

$$r_j(z, a_m) = \begin{cases} 1 & \text{if } p(r_j(z, a_m)) > t_+ \\ 0 & \text{if } p(r_j(z, a_m)) < t_- \\ \emptyset & \text{otherwise} \end{cases} \quad \forall m, \quad (4)$$

where thresholds t_+ and t_- are learned to help select the most confident positive and negative associations while at the same time provide enough discriminative attributes to predict a novel class. Assignment to \emptyset discards the attribute for ZSL since we are not confident about the type (positive or negative) of the association. We learn these thresholds using leave-K-class-out cross-validation so as to maximize zero-shot classification accuracy of the held out classes.

Zero-shot learning The score for unseen class z on image x is estimated based on the predicted attribute associations ($\mathcal{A}^z = \{a_m^z\} \subseteq \mathcal{A}$) using the Direct Attribute Prediction (DAP) [23] method:

$$s(z|x) = \prod_{a_m \in \mathcal{A}^z} p(a_m = a_m^z | x) / p(a_m), \quad (5)$$

where $p(a_m|x)$ is the posterior probability of observing attribute a_m in image x . We assume identical class and attribute priors.

4. Experiments

In this section, we evaluate our model at: (1) predicting class-attribute associations and (2) unsupervised zero-shot classification. Furthermore, we demonstrate the ability of our model to (3) transfer attributes across data sets without the cost of additional annotations. Finally, (4) we show that the model is generic and can learn different types of relations and not only attribute-based ones. In the following, we refer to our Class-Attribute Association Prediction model as CAAP.

Data setup We use two publicly available data sets.

- (i) Animals with Attributes (AwA) [23]: consists of 50 animal classes that are described with 85 attributes. The classes are split into 40 seen and 10 unseen classes for ZSL.
- (ii) aPascal/aYahoo (aPaY) [15]: contains 32 classes of artifacts, people and animals; and they are described with 64 attributes. 20 of these classes (aPascal) come from the Pascal challenge [13] and are used for training, while the rest 12 (aYahoo) are considered unseen and used for ZSL.

4.1. Predicting class-attribute association

We consider two types of relations for training our CAAP model:

Semantic relations (SR) For aPaY, we use the 3 predefined relations (*has_material*, *has_shape* and *has_part*). As

Model	AwA	aPaY
Co-Occurrence [26, 34]		
Bing	41.8 (57.4)	20.9 (69.4)
Yahoo-Img ¹	50.9 (62.5)	-
Flickr	48.7 (63.4)	28.1 (82.3)
Word Embedding		
C → A (Top Q)	41.3 (53.7)	34.2 (74.0)
C → A (Similarity)	41.3 (43.1)	34.2 (77.5)
Ours		
CAAP (SR)	79.1 (78.2)	76.1 (89.8)
CAAP (DR)	79.7 (78.9)	75.7 (89.6)

Table 1: Performance of class-attribute association predictions for unseen classes, presented in mAP (accuracy).

for AwA, a cursory look at the set of attributes shows us that they can be easily grouped into 9 sets of relationships like *has_color*, *lives_in*, *food_type*, etc.²

Data-driven relations (DR) For both data sets, we perform hierarchical agglomerative clustering on the word embeddings of the attributes and by analyzing the respective dendrogram the clustering is stopped at 10 groups of attributes.

We generate both positive and negative training triplets using the attribute annotations of the training set (e.g. *has_part*(horse, tail) = 1, *lives_in*(dolphin, desert) = 0). We estimate the number of latent factors L and λ using 5-folds cross validation. We report the performance to predict all attribute associations for the unseen classes, hence we set $t_- = t_+ = 0.5$. For words embedding, we train a skip-gram model on the Wikipedia corpus and obtain a $d = 300$ dimensional representation.

We compare our method of predicting class-attribute associations via word vector representations and learned semantic relationships against the state-of-the-art (SOTA) Co-occurrence approach and two other baselines based on Word embedding space.

Co-occurrence As in the state-of-the-art methods [26, 34], we use the Microsoft Bing Search API [1], the Flickr API [2] and Yahoo Image to obtain hit counts H_{c_k} for classes (e.g. “chimpanzee”); H_{a_m} for attributes (e.g. “stripes”), and H_{c_k, a_m} jointly for class-attribute pairs (e.g. “chimpanzee stripes”). We use the Dice score metric [26] to obtain a hit-count based class-attribute association score:

$$s_{c_k, a_m}^H = \frac{H_{c_k, a_m}}{H_{c_k} + H_{a_m}}, \quad (6)$$

¹We use Yahoo Image association scores provided by [34] for AwA.

²More details can be found in the supplementary material.

where s^H is the co-occurrence similarity matrix of classes and attributes.

Word embedding space These methods directly use the word vector representations (Sec. 3.1) to predict class-attribute associations. We present two approaches using the word embeddings:

(i) $C \rightarrow A$ (Top Q): Consider the average number of attributes that are associated with every class in the training set to be Q. For each unseen class, we consider a positive association with the Q nearest attributes (in terms of Euclidean distance) using the vector space embedding.

(ii) $C \rightarrow A$ (Similarity): Similar to the co-occurrence method, we construct a similarity matrix between class and attribute labels as:

$$s_{c_k, a_m}^W = \exp(-\|v(c_k) - v(a_m)\|_2) \quad \forall c_k, a_m. \quad (7)$$

For s^H and s^W , binary associations are obtained by choosing the best threshold over the class-attribute similarity matrix which maximizes the ZSL performance.

Results Table 1 presents the mean average precision (mAP) and accuracy for predicting class-attribute associations. Note that among co-occurrence methods Flickr and Yahoo Image search perform better than Bing web search. This can be related to the fact that the search results are grounded from visual information. As demonstrated earlier, the word embedding space is not suitable to directly model the relations and it fails to reliably predict class-attribute associations (see Fig. 2c).

Our method of modeling relations outperforms state-of-the-art by a significant amount (19% on AwA, 42% on aPaY). Table 2 presents examples of the top 5 confident positive and negative associations. In general, we observe that our model ranks the most distinctive attributes of a category higher (e.g. leopard ↔ fast, chimpanzee ↔ walk, hippopotamus ↔ strong). Fig. 4 provides a deeper insight on the performance of each semantic relation presented by the precision-recall curve.

Moreover, both SR and DR models perform at the same level with no substantial difference. Hence, the data-driven approach is a very good alternative for the semantic relations thus even removing the need to provide extra relation annotations for CAAP. In the rest of the experiments we adopt the DR approach.

4.2. Unsupervised zero-shot learning

We now present unsupervised zero-shot classification performance comparing against methods of the previous section which also use predicted class-attribute associations. For all attribute-based approaches, we use the DAP model as described in Sec. 3.3.

Word embedding space In addition to the previous attribute-based baselines, we also examine two category-based options:

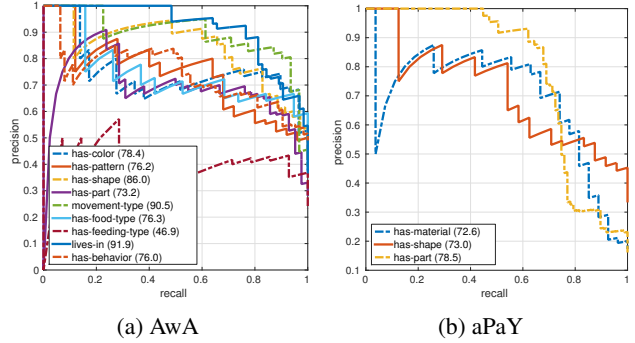


Figure 4: Prediction performance of individual relations learned by CAAP given by precision-recall curves along with mAP scores (see legend).

Unseen Category	Top Associations	
	Positive	Negative
hippopotamus	strong, <i>group</i> , walks, ground	big, claws, flies, red, nocturnal, weak
leopard	fast, lean, oldworld, active, tail	tusks, water, arctic, plankton, weak
humpback_whale	fast, ocean, water, group, fish	red, weak, tunnels, nocturnal, plains
seal	fast, <i>meat</i> teeth, <i>bulbous</i> , big, toughskin	grazer, tunnels, longleg, hooves, longneck
chimpanzee	walks, group, fast, chewteeth, active	arctic, flippers, red, plankton, straintooth

Table 2: Examples of predicted class-attribute associations for unseen classes. Wrong associations are highlighted in gray and italic.

(i) $C \rightarrow C$ (Top 1): As we see from Fig. 2a, similar classes do appear together in the word embedding space. Hence, for each unseen class we use the category classifier of the training set class which appears closest to it in the vector space.

(ii) $C \rightarrow C$ (Weighted K): This takes into consideration the similarity of the novel class for all known classes [6]. We build a classifier as a weighted linear combination of all training classes where the weights are based on distances between their vector representations:

$$s_{wc}(z|x) = \sum_{k=1}^K w_k^z s(c_k|x), \quad (8)$$

where $w_k^z = \exp(-\|v(z) - v(c_k)\|_2)$ and $s(c_k|x)$ is the score obtained by the classifier for category c_k on image x .

Furthermore, most previous works assume that the attribute labels are provided by a human operator for the unseen class. While in this paper we circumvent this additional overhead, we present the results of supervised DAP [24] as a reference.

Image features and classifiers We use the output of the last hidden layer of the public BVLC implementation [21] of GoogLeNet [39] as our 1024 dimensional image features. The deep representation is then used to train linear SVMs with regularized logistic regression [14] for the attribute and

Model	AwA	aPaY
Supervised ZSL		
DAP [24]	59.5	37.1
Unsupervised ZSL		
Co-Occurrence [34, 26]		
Bing	11.8	13.1
Yahoo-Img	39.8	-
Flickr	44.2	13.8
Word Embedding		
C → A (Top Q)	10.2	14.3
C → A (Similarity)	26.4	20.4
C → C (Top 1)	48.6	15.0
C → C (Weighted K)	40.6	22.5
CAAP (ours)	67.5	37.0

Table 3: Zero-shot classification performance presented in mean per-class accuracy.

category classifiers. The SVM parameter C is estimated using 5-folds cross validation. The same classifiers are used for the various baselines and our model. For our model we estimate the number of latent factors L and λ and additionally learn the thresholds (t_-, t_+) by 5-folds cross validation.

Results Table 3 presents the mean per class accuracy for the test classes used in zero-shot classification.

We see again that image-based hit-count information obtained by Yahoo images or Flickr outperforms general web-based search (Bing). However, they are all far from supervised ZSL performance with ground truth association.

The word embedding methods based on attributes ($C \rightarrow A$) show poor performance. In comparison, using the classifier of the nearest class ($C \rightarrow C$ (Top 1)) performs well for AwA (48%) while poorly on aPaY (15%). An explanation for this is that unseen classes on AwA are visually close to the train set while aPaY has higher diversity in class types (animals and man-made objects). Building a classifier by weighting all other classes ($C \rightarrow C$ (Weighted K)) shows moderate performance on both data sets.

Our method outperforms all baselines with an accuracy of 67.5% for AwA and 37.0% for aPaY. In fact, for aPaY CAAP performs at the same level of supervised DAP while for AwA we see impressive performance surpassing the performance of supervised ZSL with ground truth attribute associations. We show in Sec. 4.3 that our method allows to conveniently transfer attributes across data sets with no additional effort. Using automatic transfer we can improve performance even more on both data sets.

Model analysis We study the effect of the different aspects of our model on the final unsupervised ZSL performance.

(1) Single relation: In the previous experiments, we used a small set of relations that group similar attributes together.

Source (<i>seen</i>)	Target (<i>unseen</i>)		
	AwA	aPaY	AwA+aPaY
AwA	67.5	39.5	37.1
aPaY	10.4	37.0	6.2
AwA+aPaY	68.6	49.0	46.8

Table 4: Zero-shot classification accuracy when attributes are transferred across data sets using CAAP. A source AwA and Target aPaY means classifying the unseen classes of aPaY based on their predicted associations with the attributes of AwA.

Here, we group all attributes in a single abstract relation ($N = 1$) called *has_attribute* and try to model the class-attribute associations accordingly. We observe that in this setting, the absolute drop in accuracy is 5% on AwA while on aPaY we see a reduction by 12%.

(2) Fixed attribute embedding: Similar to the category embeddings, we fix the representation of the attributes during learning. Here the performance on both data sets drops by 2% on AwA and 9% on aPaY.

(3) Threshold@0.5: Rather than learning the thresholds (t_-, t_+) we set them both to $t_- = t_+ = 0.5$. In this case, the accuracy drops by 2% on AwA while the performance on aPaY goes down by 12%.

We conclude that improving the attribute representation during learning is beneficial. We notice that attribute pairs like (*big, small*) and (*weak, strong*) which get initialized with similar embeddings are pushed apart by our model to facilitate the learning of the relations. It is also good to learn multiple relations that account for the discrepancies in the attributes rather than an abstract mapping that groups all of them together in one inhomogeneous group. Our model learns proper confidence scores on the associations, and ranks most distinctive attributes higher leading to better ZSL performance when considering the most confident associations. In general, aPaY is more sensitive to changes which can be related to the large variance in both classes and attributes, since they describe not only animals (like in AwA) but also vehicles and other man made objects.

4.3. Attribute transfer across data sets

A major advantage of our approach is the ability to automatically transfer the set of attributes from one data set to another at no additional annotation cost. For example, we can use the 85 attributes of the AwA data set to describe categories from aPaY and vice-versa. Most importantly, we do *not* need to manually associate the classes of one data set with the attributes of the other. These associations are automatically obtained through our CAAP model.

In particular, we learn the relations of AwA and aPaY jointly without providing any additional associations. Then for a novel class (from AwA or aPaY), we predict its associations to the attribute set $\mathcal{A} = \mathcal{A}^{AwA} \cup \mathcal{A}^{aPaY}$. We see in Table 4 (3rd row), that CAAP results in a significant im-

provement surpassing the performance of the manually defined associations on each data set. Especially on aPaY, we see a dramatic improvement of 12% in performance which can be attributed to the fact that roughly half the classes of aPaY test set are animals, which benefit strongly from the rich attributes transferred from AwA. This demonstrates the effectiveness of CAAP in integrating knowledge from different sources without the need for any additional effort.

In Table 4, we provide additional evaluation of the attribute transfer by changing the source and target sets. Comparing the two sources AwA and aPaY, it is clear that AwA encompasses a richer diversity as it results in good performance for both test sets, while transferring attributes from aPaY→AwA results in performance on par with a random classifier. Taking a closer look at the assigned attributes, we notice the following:

(1) AwA→aPaY, not only the animal classes but even some man made classes get associated with reasonable attributes. For example, the class “jetski” is positively associated with attributes “water” and “fast”; and class “carriage” with “grazer” and “muscle”.

(2) aPaY→AwA, the attributes assigned to the animal classes are in general correct. However, aPaY doesn’t have enough animal-related attributes to distinguish the fine grained categories on AwA. Most of the test classes in AwA are assigned to attributes like “eye”, “head” and “leg”.

(3) AwA+aPaY→AwA+aPaY, even on this harder setting where we test on 22 unseen classes (*i.e.* random performance drops to 4.5% as compared to 10% on AwA and 8.3% on aPaY). Our model generalizes gracefully with 46.8% accuracy.

4.4. CAAP versus state-of-the-art

In Table 5, the performance of our approach is compared against state-of-the-art in unsupervised ZSL. Both [17] and [30] use the same word embedding as ours while [3] uses GloVe [31] and Word2Vec. Additionally, all methods in Table 5 use image embedding from GoogLeNet. CAAP outperforms approaches based only on class names [17, 30] with more than 20% on both data sets. Approaches like Text2Visual [11, 8], SJE [3] and HAT [4] make use of additional source of information like Wikipedia articles or WordNet. While theoretically this information can be obtained automatically, practically, a manual intervention is often necessary to resolve ambiguities between class names and article titles or to find the proper synset of a class in WordNet. Nonetheless, CAAP outperform state-of-the-art by 8.5% and 18.8% on AwA and aPaY respectively, while only needing the name of the unseen class.

4.5. Beyond attributes

Various approaches in the literature have reported the advantage of incorporating hierarchical information for ZSL (*e.g.* [4, 3, 33]). Our model can also learn hierarchical rela-

Model	ZSL Information	AwA	aPaY
DeViSE [17]	C	44.5	25.5
Text2Visual [11, 8]	C + Text ^{Wiki}	55.3	30.2
ConSE [30]	C	46.1	22.0
SJE [3]	C + H ^{WordNet}	60.1	-
HAT [4] ³	C + H ^{WordNet}	59.7	31.1
CAAP (ours)	C	68.6	49.0

Table 5: Unsupervised zero-shot learning accuracy of state-of-the-art versus CAAP. The second column shows the type of information leveraged by each model for the unseen classes.

tions, for example to predict the ancestors of a category. To test this, we query WordNet [28] with the AwA categories and extract the respective graph relevant to the hypernym links. We then learn the *has_ancestor* relation by generating triplets of the form *has_ancestor*(horse, equine) = 1 using the information from the extracted graph. The evaluation on AwA test set reveals that we can predict the ancestor relation of an unseen class with a mAP of 89.8%. Interestingly, learning such a hierarchy-based relation can aid the learning of some attribute-based relations. The model allows the various relations to interact and exchange information at the level of the shared latent factors. Among the improved attribute-based relations, is *has_pattern* (+2.5%), and *feeding_type* (+2.1%). These relations correlate well with the hierarchical information of the classes (*e.g.* carnivores tend to have similar pattern and feeding type). Predicting such a hierarchical relation alleviates the need of a complete hierarchy or manual synonym matching since this is automatically handled by the word embedding and CAAP model. This keeps user intervention to the minimal requirement of providing class names. We expect that modeling more relations among the classes jointly with class-attribute relations can result in better performance.

5. Conclusion

Attribute-based ZSL suffers from a major drawback of needing class-attribute associations to be defined manually. To counter this, we present an automatic approach to predict the associations between attributes and unseen classes. We model the associations using a set of relationships linking categories and their respective attributes in an embedding space. Our approach effectively predicts the associations of novel categories and outperforms state-of-the-art in two tasks; namely association prediction and unsupervised ZSL. Moreover, we demonstrate the ability of our model to transfer attributes between data sets at no cost. The transferred attributes enlarge the size of the description vocabulary, which results in more discriminative classifiers for ZSL yielding an additional boost in performance.

³Results are from the updated arXiv version of [4]: [1604.00326v1](https://arxiv.org/abs/1604.00326v1)

References

- [1] Bing Search API. <https://datamarket.azure.com/dataset/bing/search>. 5
- [2] Flickr API. <https://www.flickr.com/services/api/flickr.photos.search.html>. 5
- [3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. In *CVPR*, 2015. 2, 8
- [4] Z. Al-Halah and R. Stiefelhagen. How to Transfer? Zero-Shot Object Recognition via Hierarchical Transfer of Semantic Attributes. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2015. 1, 2, 8
- [5] J. L. Ba, K. Swersky, S. Fidler, and R. Salakhutdinov. Predicting Deep Zero-Shot Convolutional Neural Networks using Textual Descriptions. In *ICCV*, 2015. 2
- [6] E. Bart and S. Ullman. Single-example learning of novel classes using representation by similarity. In *BMVC*, 2005. 1, 6
- [7] I. Biederman. Recognition by components: A theory of human image understanding. *Psychological Review*, 1987. 1
- [8] L. Bo and C. Sminchisescu. Twin gaussian processes for structured prediction. *IJCV*, 2010. 8
- [9] Q. Chen, J. Huang, R. Feris, L. M. Brown, J. Dong, and S. Yan. Deep Domain Adaptation for Describing People Based on Fine-Grained Clothing Attributes. In *CVPR*, 2015. 2
- [10] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *CVPR*, 2009. 1
- [11] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a Classifier: Zero-Shot Learning Using Purely Textual Descriptions. In *ICCV*, 2013. 2, 8
- [12] V. Escorcia, J. C. Niebles, and B. Ghanem. On the Relationship between Visual Attributes and Convolutional Networks. In *CVPR*, 2015. 2
- [13] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2008 (VOC2008) Results, 2008. 5
- [14] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008. 6
- [15] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing Objects by their Attributes. In *CVPR*, 2009. 2, 4, 5
- [16] V. Ferrari and A. Zisserman. Learning Visual Attributes. In *NIPS*, 2008. 2
- [17] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. DeViSE: A Deep Visual-Semantic Embedding Model. In *NIPS*, 2013. 2, 8
- [18] Y. Fu, T. M. Hospedales, T. Xiang, Z. Fu, and S. Gong. Transductive Multi-view Embedding for Zero-Shot Recognition and Annotation. In *ECCV*, 2014. 2
- [19] E. H. Huang, R. Socher, C. D. Manning, and A. Y. Ng. Improving word representations via global context and multiple word prototypes. In *Association for Computational Linguistics (ACL)*, 2012. 2
- [20] R. Jenatton, A. Bordes, N. L. Roux, and G. Obozinski. A Latent Factor Model for Highly Multi-relational Data. In *NIPS*, 2012. 4
- [21] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv:1408.5093*, 2014. 6
- [22] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, 2012. 1
- [23] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 1, 2, 5
- [24] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *T-PAMI*, 2013. 4, 6, 7
- [25] J. Liu, B. Kuipers, and S. Savarese. Recognizing Human Actions by Attributes. In *CVPR*, 2011. 1, 2
- [26] T. Mensink, E. Gavves, and C. G. M. Snoek. COSTA: Co-Occurrence Statistics for Zero-Shot Classification. In *CVPR*, 2014. 2, 5, 7
- [27] T. Mikolov, G. Corrado, K. Chen, and J. Dean. Efficient Estimation of Word Representations in Vector Space. In *International Conference on Learning Representations (ICLR)*, 2013. 2, 3
- [28] G. A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM Vol. 38, No. 11: 39-41.*, 1995. 2, 8
- [29] M. Nickel, V. Tresp, and H.-P. Kriegel. Factorizing yago: Scalable machine learning for linked data. In *International Conference on World Wide Web (WWW)*, 2012. 4
- [30] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-Shot Learning by Convex Combination of Semantic Embeddings. In *ICLR*, 2014. 2, 8
- [31] J. Pennington, R. Socher, and C. D. Manning. GloVe: Global Vectors for Word Representation. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014. 8
- [32] M. Rohrbach, S. Ebert, and B. Schiele. Transfer learning in a transductive setting. In *NIPS*, 2013. 2
- [33] M. Rohrbach, M. Stark, and B. Schiele. Evaluating Knowledge Transfer and Zero-Shot Learning in a Large-Scale Setting. In *CVPR*, 2011. 1, 2, 8
- [34] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What Helps Where – And Why? Semantic Relatedness for Knowledge Transfer. In *CVPR*, 2010. 2, 5, 7
- [35] A. Sadovnik, A. Gallagher, and T. Chen. Its Not Polite To Point: Describing People With Uncertain Attributes. In *CVPR*, 2013. 2
- [36] B. Saleh, A. Farhadi, and A. Elgammal. Object-Centric Anomaly Detection by Attribute-Based Reasoning. In *CVPR*, 2013. 2
- [37] R. Socher, M. Ganjoo, C. D. Manning, and A. Y. Ng. Zero-Shot Learning Through Cross-Modal Transfer. In *NIPS*, 2013. 2

- [38] I. Sutskever, J. B. Tenenbaum, and R. R. Salakhutdinov. Modelling Relational Data using Bayesian Clustered Tensor Factorization. In *NIPS*, 2009. [4](#)
- [39] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going Deeper with Convolutions. *arXiv:1409.4842v1*, 2014. [1](#), [6](#)
- [40] L. van der Maaten and G. Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008. [3](#)
- [41] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011. [2](#), [4](#)
- [42] F. X. Yu, L. Cao, R. S. Feris, J. R. Smith, and S.-F. Chang. Designing Category-Level Attributes for Discriminative Visual Recognition . In *CVPR*, 2013. [2](#)