Evaluation of Output Embeddings for Fine-Grained Image Classification

Zeynep Akata*, Scott Reed†, Daniel Walter†, Honglak Lee† and Bernt Schiele*

* Computer Vision and Multimodal Computing
Max Planck Institute for Informatics, Saarbrucken, Germany

† Computer Science and Engineering Division
University of Michigan, Ann Arbor

Abstract

Image classification has advanced significantly in recent years with the availability of large-scale image sets. However, fine-grained classification remains a major challenge due to the annotation cost of large numbers of fine-grained categories. This project shows that compelling classification performance can be achieved on such categories even without labeled training data. Given image and class embeddings, we learn a compatibility function such that matching embeddings are assigned a higher score than mismatching ones; zero-shot classification of an image proceeds by finding the label yielding the highest joint compatibility score. We use state-of-the-art image features and focus on different supervised attributes and unsupervised output embeddings either derived from hierarchies or learned from unlabeled text corpora. We establish a substantially improved state-of-the-art on the Animals with Attributes and Caltech-UCSD Birds datasets. Most encouragingly, we demonstrate that purely unsupervised output embeddings (learned from Wikipedia and improved with fine-grained text) achieve compelling results, even outperforming the previous supervised state-of-the-art. By combining different output embeddings, we further improve results.

1. Introduction

The image classification problem has been redefined by the emergence of large scale datasets such as ImageNet [7]. Since deep learning methods [27] dominated recent Large-Scale Visual Recognition Challenges (ILSVRC12-14), the attention of the computer vision community has been drawn to Convolutional Neural Networks (CNN) [31]. Training CNNs requires massive amounts of labeled data; but, in fine-grained image collections, where the categories are visually very similar, the data population decreases significantly. We are interested in the most extreme case of learning with a limited amount of labeled data, zero-shot learning, in which no labeled data is available for some classes.

Without labels, we need alternative sources of information that relate object classes. Attributes [15, 14, 29], which describe well-known common characteristics of objects, are an appealing source of information, and they can be easily obtained through crowd-sourcing techniques [8, 41]. However, fine-grained concepts present a special challenge: due to the high degree of similarity among categories, a large number of attributes are required to effectively model these subtle differences. This increases the cost of attribute annotation. One aim of this work is to move towards eliminating the human labeling component from zero-shot learning, e.g. by using alternative sources of information.

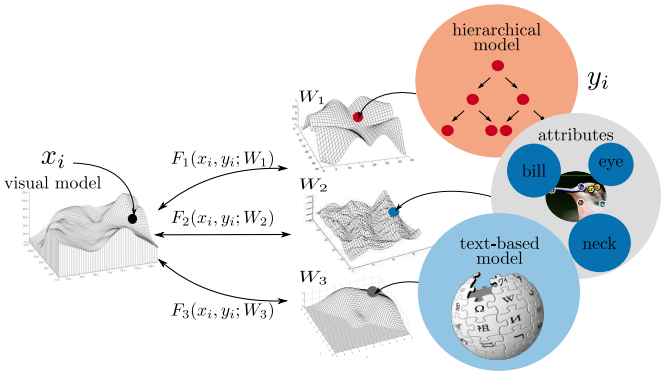

Figure 1. Structured Joint Embedding leverages images (x_i) and labels (y_i) by learning parameters W of a function $F(x_i, y_i, W)$ that measures the compatibility between input ($\theta(x_i)$) and output embeddings ($\varphi(y_i)$). It is a general framework that can be applied to any learning problem with more than one modality.

On the other hand, large-margin support vector machines (SVM) operate with labeled training images, so a lack of labels limits their use for this task. Inspired by previous work on label embedding [56, 3, 1] and structured SVMs [52, 38], we propose to use a Structured Joint Embedding (SJE) framework (Fig. 1) that relates input embeddings (i.e. image features) and output embeddings (i.e. side information) through a compatibility function, therefore taking advantage of a structure in the output space. The SJE framework separates the subspace learning problem from the specific input and output features used in a given application. As a general framework, it can be applied to any learning problem where more than one modality is provided for an object.

Our contributions are: (1) We demonstrate that unsuper-

vised class embeddings trained from large unlabeled text corpora are competitive to previously published results that use human supervision. (2) Using the most recent deep architectures as input embeddings, we significantly improve the state-of-the-art (SoA). (3) We extensively evaluate several unsupervised output embeddings for fine-grained classification in a zero-shot setting on three challenging datasets. (4) By combining different output embeddings we obtain best results, surpassing the SoA by a large margin. (5) We propose a novel weakly-supervised Word2Vec variant that improves the accuracy when combined with other output embeddings.

The rest of the paper is organized as follows. Section 2 provides a review of the relevant literature; Sec. 3 details the SJE method; Sec. 4 explains the output embeddings that we analyze; Sec. 5 presents our experimental evaluation; Sec. 6 presents the discussion and our conclusions.

2. Related Work

Learning to classify in the absence of labeled data (zero-shot learning) [58, 45, 25, 29, 2, 37, 34, 17] is a challenging problem, and achieving better-than-chance performance requires structure in the output space. Attributes [15, 13, 29] provide one such space; they relate different classes through well-known and shared characteristics of objects.

Attributes, which are often collected manually [25, 41, 12], have shown promising results in various applications, i.e. caption generation [28, 39], face recognition [48, 6], image retrieval [49, 11], action recognition [33, 57] and image classification [29, 2]. The main challenge of attribute-based zero-shot learning arises on more challenging fine-grained data collections [55, 26], in which categories may visually differ only subtly. Therefore, generic attributes fail at modeling small intra-class variance between objects. Improved performance requires a large number of specific attributes which increases the cost of data gathering.

As an alternative to manual annotation, side information can be collected automatically from text corpora. Bag-of-words [19] is an example where class embeddings correspond to histograms of vocabulary words extracted automatically from unlabeled text. Another example is using taxonomical order of classes [52] as structured output embeddings. Such a taxonomy can be built automatically from a pre-defined ontology such as WordNet [36, 46, 1] . In this case, the distance between nodes is measured using semantic similarity metrics [24, 30, 32, 44]. Finally, distributed text representations [35, 42] learned from large unsupervised text corpora can be employed as structured embeddings. We compare several representatives of these methods (and their combinations) in our evaluation.

Embedding labels in an Euclidean space is an effective tool to model latent relationships between classes [3]. These relationships can be collected separately from the data [21, 9], learned from the data [56, 20] or derived from side information [15, 16, 1, 37]. In order to collect relationships independently of data, compressed sensing [21] uses random projections whereas Error Correcting Output Codes [9] builds embeddings inspired from information theory. WSABIE [56] uses images with their corresponding labels to learn an embedding of the labels, and CCA [20] maximizes the correlation between two different data modalities. DeViSE [16] employs a ranking formulation for zero-shot learning using images and distributed text representations. The ALE [1] method employs an approximate ranking formulation for the same using images and attributes. ConSe [37] uses the probabilities of a softmax-output layer to weigh the semantic vectors of all the classes. In this work, we use the multiclass objective to learn structured output embeddings obtained from various sources.

Among the closest related work, ALE [1] uses Fisher Vectors (FV [43]) as input and binary attributes / hierarchies as output embeddings. Similarly, DeviSe [16] uses CNN [27] features as input and Word2Vec [35] representations as output embeddings. In this work, we benefit from both ideas: (1) We use SoA image features, i.e. FV and CNN, (2) among others, we also use attributes and Word2Vec as output embeddings. Our work differs from [16] w.r.t. two aspects: (1) We propose and evaluate several output embedding methods specifically built for fine-grained classification. (2) We show how some of these output embeddings complement each other for zero-shot learning on general and fine-grained datasets. The reader should be aware of [2].

3. Structured Joint Embeddings

In this work, we aim to leverage input and output embeddings in a joint framework by learning a compatibility between these embeddings. We are interested in the problem of zero-shot learning for image classification where training and test images belong to two disjoint sets of classes.

Following [1], given input/output $x_n \in \mathcal{X}$ and $y_n \in \mathcal{Y}$ from $\mathcal{S} = \{(x_n, y_n), n = 1 \dots N\}$, Structured Joint Embedding (SJE) learns $f : \mathcal{X} \rightarrow \mathcal{Y}$ by minimizing the empirical risk $\frac{1}{N} \sum_{n=1}^{N} \Delta(y_n, f(x_n))$ where $\Delta : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$ defines the cost of predicting $f(x)$ when the true label is y. Here, we use the $0/1$ loss.

3.1. Model

We define a compatibility function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$ between an input space $\mathcal{X}$ and a structured output space $\mathcal{Y}$. Given a specific input embedding, we derive a prediction by maximizing the compatibility F over SJE as follows:

$$f(x; w) = \arg\max_{y \in \mathcal{Y}} F(x, y; w).$$

The parameter vector w can be written as a $D \times E$ matrix W with D being the input embedding dimension and E being the output embedding dimension. This leads to the bi-linear form of the compatibility function:

$$F(x, y; W) = \theta(x)^\top W \varphi(y). \quad (1)$$

Here, the input embedding is denoted by $\theta(x)$ and the output embedding by $\varphi(y)$. The matrix W is learned by enforcing the correct label to be ranked higher than any of the other labels (Sec. 3.2), i.e. multiclass objective. This formulation is closely related to [1, 16, 56]. Within the label embedding framework, ALE [1] and DeViSe [16] use pairwise ranking objective, WSABIE [56] learns both $\varphi(y)$ and W through ranking, whereas we use multiclass objective. Similarly, [40, 50] use the regression objective and CCA [20] maximizes the correlation of input and output embeddings.

3.2. Parameter Learning

According to the unregularized structured SVM formulation [52], the objective is:

$$\frac{1}{N} \sum_{n=1}^{N} \max_{y \in \mathcal{Y}} \{0, \ell(x_n, y_n, y)\}. \quad (2)$$

where the misclassification loss $\ell(x_n, y_n, y)$ takes the form:

$$\Delta(y_n, y) + \theta(x_n)^\top W \varphi(y) - \theta(x_n)^\top W \varphi(y_n) \quad (3)$$

For the zero-shot learning scenario, the training and test classes are disjoint. Therefore, we fix φ to the output embeddings of training classes and learn W. For prediction, we project a test image onto the W and search for the nearest output embedding vector (using the dot product similarity) that corresponds to one of the test classes.

We use Stochastic Gradient Descent (SGD) for optimization which consists in sampling (x_n, y_n) at each step and searching for the highest ranked class y. If $\arg\max_{y \in \mathcal{Y}} \ell(x_n, y_n, y) \neq y_n$, we update W as follows:

$$W^{(t)} = W^{(t-1)} + \eta_t \theta(x_n)[\varphi(y_n) - \varphi(y)]^\top \quad (4)$$

where η_t is the learning step-size used at iteration t. We use a constant step size chosen by cross-validation and we perform regularization through early stopping.

3.3. Learning Combined Embeddings

For some classification tasks, there may be multiple output embeddings available, each capturing a different aspect of the structure of the output space. Each may also have a different signal-to-noise ratio. Since each output embedding possibly offers non-redundant information about the output space, as also shown in [45, 2], we can learn a better joint embedding by combining them together. We model the resulting compatibility score as

$$F(x, y; \{W\}_{1..K}) = \sum_k \alpha_k \theta(x)^\top W_k \varphi_k(y) \quad (5)$$
$$\text{s.t.} \sum_k \alpha_k = 1$$

where $W_1, ..., W_K$ are the joint embedding weight matrices corresponding to the K output embeddings (φ_k). In our experiments, we first train each W_k independently, then perform a grid search over α_k on a validation set. Interestingly, we found that the optimal α_k for previously-seen classes is often different from the one for unseen classes. Therefore, it is critical to cross-validate α_k on the zero-shot setting.

Note that if we take $\alpha_k = 1/K, \forall k$, Equation 5 is equivalent to simply concatenating the φ_k. This corresponds to stacking the W_k into a single matrix W and computing the standard compatibility as in Equation 1. However, such a stacking learns a large W where a high dimensional φ biases the final prediction. In contrast, α eliminates the bias, leading to better predictions. Thus, α_k can be thought of as the confidence associated with φ_k whose contribution we can control. We show in Sec. 5.2 that finding an appropriate α_k can yield improved accuracy compared to any single φ.

4. Output Embeddings

In this section, we describe three types of output embeddings: human-annotated attributes, unsupervised word embeddings learned from large text corpora, and hierarchical embeddings derived from WordNet.

4.1. Embedding by Human Annotation: Attributes

Annotating images with class labels is a laborious process when the objects represent fine-grained concepts that are not common in our daily lives. Attributes provide a means to describe such fine-grained concepts. They model shared characteristics of objects such as color and texture which are easily annotated by humans and converted to machine-readable vector format. The set of descriptive attributes may be determined by language experts [29] or by fine-grained object experts [55]. The association between an attribute and a category can be a binary value depicting the presence/absence of an attribute ($\varphi^{0,1}$ [29, 1, 45]) or a continuous value that defines the confidence level of an attribute ($\varphi^{\mathcal{A}}$ [29, 2, 47]) for each class. We write per-class attributes as:

$$\varphi(y) = [\rho_{y,1}, \ldots, \rho_{y,E}]^\top$$

where $\rho_{y,i}$ can be $\{0, 1\}$ or a real number that associates a class with an attribute, y denotes the associated class and E is the number of attributes. Potentially, $\varphi^{\mathcal{A}}$ encodes more information than $\varphi^{0,1}$. For instance, for classes *rat, monkey, whale* and the attribute *big*, $\varphi^{0,1} = [0, 0, 1]$ im-

plies that in terms of size *rat* = *monkey* < *whale*, whereas $\varphi^{\mathcal{A}} = [2, 10, 90]$ can be interpreted as *rat* < *monkey* << *whale* which is more accurate. We empirically show the benefit of $\varphi^{\mathcal{A}}$ over $\varphi^{0,1}$ in Sec. 5.2. In practice, our output embeddings use a per-class vector form, but they can vary in dimensionality (E). For the rest of the section we denote the output embeddings as φ for brevity.

4.2. Learning Label Embeddings from Text

In this section, we describe unsupervised and weakly-supervised label embeddings mined from text. With these label embeddings, we can (1) avoid dependence on costly manual annotation of attributes and (2) combine the embeddings with attributes, where available, to achieve better performance.

Word2Vec ($\varphi^{\mathcal{W}}$). In Word2Vec [35], a two-layer neural network is trained to predict a set of target words from a set of context words. Words in the vocabulary are assigned with one-shot encoding so that the first layer acts as a look-up table to retrieve the embedding for any word in the vocabulary. The second layer predicts the target word(s) via hierarchical soft-max. Word2Vec has two main formulations for the target prediction: skip-gram (SG) and continuous bag-of-words (CBOW). In SG, words within a local context window are predicted from the centering word. In CBOW, the center word of a context window is predicted from the surrounding words. Embeddings are obtained by back-propagating the prediction error gradient over a training set of context windows sampled from the text corpus.

GloVe ($\varphi^{\mathcal{G}}$). GloVe [42] incorporates co-occurrence statistics of words that frequently appear together within the document. Intuitively, the co-occurrence statistics encode meaning since semantically similar words such as "ice" and "water" occur together more frequently than semantically dissimilar words such as "ice" and "fashion." The training objective is to learn word vectors such that their dot product equals the co-occurrence probability of these two words. This approach has recently been shown to outperform Word2Vec on the word analogy prediction task [42].

Weakly-supervised Word2Vec ($\varphi^{\mathcal{W}_{ws}}$). The standard Word2Vec [35] scans the entire document using each word within a sample window as the target for prediction. However, if we know the global context, i.e. the topic of the document, we can use that topic as our target. For instance, in Wikipedia, the entire article is related to the same topic. Therefore, we can sample our context windows from any location within the article rather than searching for context windows where the topic explicitly appears in the text. We consider this method as a weak form of supervision.

We achieve the best results in our experiments using our novel variant of the CBOW formulation. Here, we pre-train the first layer weights using standard Word2Vec on Wikipedia, and fine-tune the second layer weights using a negative-sampling objective [18] only on the fine-grained text corpus. These weights correspond to the final output embedding. The negative sampling objective is formulated as follows:

$$L = \sum_{w,c \in D_+} \log \sigma(v_c^T v_w) + \sum_{w',c \in D_-} \log \sigma(-v_c^T v_{w'}) \quad (6)$$

$$v_c = \sum_{i \in \text{context(w)}} v_i / |\text{context(w)}|$$

where v_w and $v_{w'}$ are the label embeddings we seek to learn, and v_c is the average of word embeddings v_i within a context window around word w. D_+ consists of context v_c and matching targets v_w, and D_- consists of the same v_c and mismatching $v_{w'}$. To find the v_i (which are the columns of the first-layer network weights), we take them from a standard unsupervised Word2Vec model trained on Wikipedia.

During SGD, the v_i are fixed and we update each sampled v_w and $v_{w'}$ at each iteration. Intuitively, we seek to maximize the similarity between context and target vectors for matching pairs, and minimize it for mismatching pairs.

Bag-of-Words ($\varphi^{\mathcal{B}}$). BoW [19] builds a "bag" of word frequencies by counting the occurrence of each vocabulary word that appears within a document. It does not preserve the order in which words appear in a document, so it disregards the grammar. We collect Wikipedia articles that correspond to each object class and build a vocabulary of most frequently occurring words. We then build histograms of these words to vectorize our classes.

$\rho_{jcn} = 2 * IC(mscs(u,v)) - (IC(u) + IC(v))$
$\rho_{lin} = \dfrac{2 * IC(mscs(u,v))}{IC(u) + IC(v)}$
$\rho_{path} = \min_{p \in pth(u,v)} len(p)$

Table 1. Notations [5]: mscs (most specific common subsumer), pth (set of paths between two nodes), len (path length), IC (Information Content, defined as the log of the probability of finding a word in a text corpus independent of the hierarchy).

4.3. Hierarchical Embeddings

Semantic similarity measures how closely related two word senses are according to their meaning. Such a similarity can be estimated by measuring the distance between terms in an ontology. WordNet[1], a large-scale hierarchical database of over 100,000 words for English, provides us a means of building our class hierarchy. To measure similarity, we use Jiang-Conrath [24] (φ^{jcn}), Lin [32] (φ^{lin}) and path (φ^{path}) similarities formulated in Table 1. We denote our whole family of hierarchical embeddings as $\varphi^{\mathcal{H}}$. For a more detailed survey, the reader may refer to [5].

[1] http://wordnetweb.princeton.edu/

5. Experiments

While our main contribution is a detailed analysis of output embeddings, good image representations are crucial to obtain good classification performance. In Sec. 5.1 we detail datasets, input and output embeddings used in our experiments and in Sec. 5.2 we present our results.

5.1. Experimental Setting

We evaluate SJE on three datasets: Caltech UCSD Birds (CUB) [54] and Stanford Dogs (Dogs)[2] [26] are fine-grained, and Animals With Attributes (AWA) [29] is a standard attribute dataset for zero-shot classification. CUB contains 11,788 images of 200 bird species, Dogs contains 19,501 images of 113 dog breeds and AWA contains 30,475 images of 50 different animals. We use a truly zero-shot setting where the train, val, and test sets belong to mutually exclusive classes. We employ train and val, i.e. disjoint subsets of training set, for cross-validation. We report average per-class top-1 accuracy on the test set. For CUB, we use the same zero-shot split as [1] with 150 classes for the train+val set and 50 disjoint classes for the test set. AWA has a predefined split for 40 train+val and 10 test classes. For Dogs, we use approximately the same ratio of classes for train+val/test as CUB, i.e. 85 classes for train+val and 28 classes for test. This is the first attempt to perform zero-shot learning on the Dogs dataset.

Input Embeddings. We use Fisher Vectors (FV) and Deep CNN Features (CNN). FV [43] aggregates per image statistics computed from local image patches into a fixed-length local image descriptor. We extract 128-dim SIFT from regular grids at multiple scales, reduce them to 64-dim using PCA, build a visual vocabulary with 256 Gaussians [53] and finally reduce the FVs to 4,096. As an alternative, we extract features from a deep convolutional network. Features that are typically obtained from the activations of the fully connected layers have been shown to induce semantic similarities. We resize each image to 224×224 and feed into the network which was pre-trained following the model architecture of either AlexNet [27] or GoogLeNet [51, 22]. For AlexNet (denoted as CNN) we use the 4,096-dim top-layer hidden unit activations (fc7) as features, and for GoogLeNet (denoted as GOOG) we use the 1,024-dim top-layer pooling units. For both networks, we used the publicly-available BVLC implementations [23]. We do not perform any task-specific pre-processing, such as cropping foreground objects or detecting parts.

Output Embeddings. AWA classes have 85 binary and continuous attributes. CUB classes have 312 continuous attributes and the continuous values are thresholded around the mean to obtain binary attributes. The Dogs dataset does not have human-annotated attributes available.

[2]We use 113 classes that appear in the Federation Cynologique Internationale (FCI) database of dog breeds.

		AWA		CUB	
		$\varphi^{0,1}$	$\varphi^{\mathcal{A}}$	$\varphi^{0,1}$	$\varphi^{\mathcal{A}}$
Ours	FV (4K)	36.6	42.3	15.2	19.0
	CNN (4K)	45.9	61.9	30.0	40.3
	GOOG (1K)	52.0	**66.7**	37.8	**50.1**
SoA	ALE [2] (64K)	44.6	48.5	22.3	26.9

Table 2. Discrete ($\varphi^{0,1}$) and continuous ($\varphi^{\mathcal{A}}$) attributes with SJE vs SoA. For AWA (CUB) [2] achieves 49.4% (27.3%) by combining $\varphi^{\mathcal{A}}$ and binary hierarchies.

We train Word2Vec ($\varphi^{\mathcal{W}}$) and GloVe ($\varphi^{\mathcal{G}}$) on the English-language `Wikipedia` from 13.02.2014. We first pre-process it by replacing the class-names, i.e. *black-footed albatross*, with alternative unique names, i.e. scientific name, *phoebastrianigripes*. We cross-validate the skip-window size and embedding dimensions. For our proposed weakly-supervised Word2Vec ($\varphi^{\mathcal{W}_{ws}}$), we use the same embedding dimensions as the plain Word2Vec ($\varphi^{\mathcal{W}}$). For BoW, we download the `Wikipedia` articles that correspond to each class and build the vocabulary by omitting least- and most-frequently occurring words. We cross-validate the vocabulary size. $\varphi^{\mathcal{B}}$ is a histogram of the vocabulary words as they appear in the respective document.

For hierarchical embeddings ($\varphi^{\mathcal{H}}$), we use the WordNet hierarchy spanning our classes and their ancestors up to the root of the tree. We employ the widely used NLTK library[3] for building the hierarchy and measuring the similarity between nodes. Therefore, each $\varphi^{\mathcal{H}}$ vector is populated with similarity measures of the class to all other classes.

Combination of output embeddings. We explore combinations of five types of output embeddings: supervised attributes $\varphi^{\mathcal{A}}$, unsupervised Word2Vec $\varphi^{\mathcal{W}}$, GloVe $\varphi^{\mathcal{G}}$, BoW $\varphi^{\mathcal{B}}$ and WordNet-derived similarity embeddings $\varphi^{\mathcal{H}}$. We either concatenate (*cnc*) or combine (*cmb*) different embeddings. In *cnc*, for instance in AWA, 85-dim $\varphi^{\mathcal{A}}$ and 400-dim $\varphi^{\mathcal{W}}$ would be merged to 485-dim output embeddings. In this case, if we use 1,024-dim GOOG as input embeddings, we learn a single 1,024×485-dim W. In *cmb*, we first learn 1,024×85-dim $W_{\mathcal{A}}$ and 1,024×400-dim $W_{\mathcal{W}}$ and then cross-validate the α coefficients to determine the amount each embedding contributes to the final score.

5.2. Experimental Results

In this section, we evaluate several output embeddings on the CUB, AWA and Dogs datasets.

Discrete vs Continuous Attributes. Attribute representations are defined as a vector per class, or a column of the (class × attribute) matrix. These vectors (85-dim for AWA, 312-dim for CUB) can either model the presence/absence ($\varphi^{0,1}$) or the confidence level ($\varphi^{\mathcal{A}}$) of each attribute. We

[3]`http://www.nltk.org/`

supervision	source	φ	**AWA**	**CUB**	**Dogs**
unsupervised	text	$\varphi^{\mathcal{W}}$	51.2	**28.4**	19.6
	text	$\varphi^{\mathcal{G}}$	**58.8**	24.2	17.8
	text	$\varphi^{\mathcal{B}}$	44.9	22.1	**33.0**
	WordNet	$\varphi^{\mathcal{H}}$	51.2	20.6	24.3
supervised	human	$\varphi^{0,1}$	52.0	37.8	-
	human	$\varphi^{\mathcal{A}}$	**66.7**	**50.1**	-

Table 3. Summary of zero-shot learning results with SJE w.r.t. supervised and unsupervised output embeddings (Input embeddings: 1K-GOOG).

show that continuous attributes indeed encode more semantics than binary attributes by observing a substantial improvement with $\varphi^{\mathcal{A}}$ over $\varphi^{0,1}$ with deep features (Tab. 2). Overall, CNN outperforms FV, while GOOG gives the best performing results; therefore in the following, we comment only on our results obtained using GOOG.

On CUB, i.e. a fine-grained dataset, $\varphi^{0,1}$ obtains 37.8% accuracy, which is significantly above the SoA (26.9% [2]). Moreover, $\varphi^{\mathcal{A}}$ achieves an impressive 50.1% accuracy; outperforming the SoA by a large margin. We observe the same trend for AWA, which is a benchmark dataset for zero-shot learning. On AWA, $\varphi^{0,1}$ obtains 52.0% accuracy and $\varphi^{\mathcal{A}}$ improves the accuracy substantially to 66.7%, significantly outperforming the SoA (48.5% [2]). To summarize, we have shown that $\varphi^{\mathcal{A}}$ improves the performance of $\varphi^{0,1}$ using deep features, which indicates that with $\varphi^{\mathcal{A}}$, the SJE method learns a matrix W that better approximates the compatibility of images and side information than $\varphi^{0,1}$.

Learned Embeddings from Text. As the visual similarity between objects in different classes increases, e.g. in fine-grained datasets, the cost of collecting attributes also increases. Therefore, we aim to extract class similarities automatically from unlabeled online textual resources. We evaluate three methods, Word2Vec ($\varphi^{\mathcal{W}}$), GloVe ($\varphi^{\mathcal{G}}$) and the historically most commonly-used method BoW ($\varphi^{\mathcal{B}}$). We build $\varphi^{\mathcal{W}}$ and $\varphi^{\mathcal{G}}$ on the entire English Wikipedia dump. Note that the plain Word2Vec [35] was used in [2]; however, rather than using Word2Vec in an averaging mechanism, we pre-process the Wikipedia as described in Sec 4.2 so that our class names are directly present in the Word2Vec vocabulary. This leads to a significant accuracy improvement. For $\varphi^{\mathcal{B}}$ we use a subset of Wikipedia populated only with articles that correspond to our classes.

On CUB (Tab. 3), the best accuracy is observed with $\varphi^{\mathcal{W}}$ (28.4%) improving the supervised SoA (26.9% [2], Tab. 2). This is promising and impressive since $\varphi^{\mathcal{W}}$ does not use any human supervision. On AWA (Tab. 3), the best accuracy is observed with $\varphi^{\mathcal{G}}$ (58.8%) followed by $\varphi^{\mathcal{W}}$ (51.2%), improving the supervised SoA (48.5% [2]) significantly. On Dogs (Tab. 3), the best accuracy is obtained with $\varphi^{\mathcal{B}}$ (33.0%). On the other hand, using $\varphi^{\mathcal{W}}$ (19.6%) and $\varphi^{\mathcal{G}}$ (17.8%) leads to significantly lower accuracies. Unlike

	$\varphi^{\mathcal{G}}$			$\varphi^{\mathcal{W}}$			$\varphi^{\mathcal{W}}$ (W) + $\varphi^{\mathcal{W}_{ws}}$ (B)
	B	W	B+W	B	W	B+W	
FV	10.5	13.3	13.2	16.0	16.0	16.5	17.1
CNN	13.4	20.6	20.6	20.0	24.1	21.4	25.1
GOOG	13.7	24.2	**26.1**	22.5	**28.4**	27.5	**29.7**

Table 4. Comparison of Word2Vec ($\varphi^{\mathcal{W}}$) and GloVe ($\varphi^{\mathcal{G}}$) learned from a bird specific corpus (B), Wikipedia (W) and their combination (B + W), evaluated on CUB (Input embeddings: 4K-FV, 4K-CNN and 1K-GOOG).

birds, different dog breeds belong to the same species and thus they share a common scientific name. As a result, our method of cleanly pre-processing Wikipedia by replacing the occurrences of bird names with a unique scientific name was not possible for Dogs. This may lead to vectors obtained from Wikipedia for dogs that are vulnerable to variation in nomenclature. In summary, our results indicate no winner among $\varphi^{\mathcal{W}}$, $\varphi^{\mathcal{G}}$ and $\varphi^{\mathcal{B}}$. These embeddings may be task specific and complement each other. We investigate the complementarity of embeddings in the following sections.

Effect of Text Corpus. For $\varphi^{\mathcal{W}}$ and $\varphi^{\mathcal{G}}$, we analyze the effects of three text corpora (B, W, B+W) with varying size and specificity. We build our specialized bird corpus (B) by collecting bird-related information from various online resources, i.e. `audubon.org`, `birdweb.org`, `allaboutbirds.org` and BNA[4]. In combination, this corresponds to 50MB of bird-related text. We use the English-language `Wikipedia` from 13.02.2014 as our large and general corpus (W) which is 40GB of text. Finally, we combine B and W to build a large-scale text corpus enriched with bird specific text (B+W). On W and B+W, a small window size (10 for $\varphi^{\mathcal{W}}$ and 20 for $\varphi^{\mathcal{G}}$); on B, a large window size (35 for $\varphi^{\mathcal{W}}$ and 50 for $\varphi^{\mathcal{G}}$) is required. We choose parameters after a grid search. Increased specificity of the text corpus implies semantic consistency throughout the text. Therefore, large context windows capture semantics well in our bird specific (B) corpus. On the other hand, W is organized alphabetically w.r.t. the document title; hence, a large sampling window can include content from another article that is adjacent to the target word alphabetically. Here, small windows capture semantics better by looking at the text locally. We report our results in Tab. 4.

Using $\varphi^{\mathcal{G}}$, B+W (26.1%) gives the highest accuracy, followed by W (24.2%). One possible reason is that when the semantic similarity is modeled with cooccurrence statistics, output embeddings become more informative with the increasing corpus size, since the probability of cooccurrence of similar concepts increases.

Using $\varphi^{\mathcal{W}}$, the accuracy obtained with B (22.5%) is already higher than the $\varphi^{0,1}$-based SoA (22.3%), illustrating the benefit of using fine-grained text for fine-grained tasks. Another advantage of using B is that, since it is short,

[4] `http://bna.birds.cornell.edu/bna/`

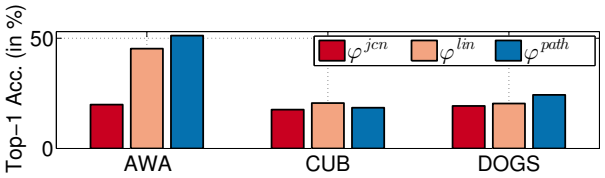

Figure 2. Comparison of WordNet similarity measures: φ^{jcn}, φ^{lin} and φ^{path}. We use $\varphi^{\mathcal{H}}$ as a general name for hierarchical output embedding. (Input embedding: 1K-GOOG).

building $\varphi^{\mathcal{W}}$ is efficient. Moreover, building $\varphi^{\mathcal{W}}$ with B does not require any annotation effort. Building $\varphi^{\mathcal{W}}$ using W (28.4%) gives the highest accuracy, followed by W + B (27.5%) which improves the supervised SoA (26.9%). We speculate that since Word2Vec is a variant of the Feedforward Neural Network Language Model (FNNLM) [4], a deep architecture, it may learn more from negative data than positives. This was also observed for CNN features learned with a large number of unlabeled surrogate classes [10].

Additionally, we propose a weakly-supervised alternative to Word2Vec framework ($\varphi^{\mathcal{W}_{ws}}$, Sec. 4.2). The weak-supervision comes from using the specialized B corpus to fine-tune the weights of the network and model the bird-related information. With $\varphi^{\mathcal{W}_{ws}}$ alone, we obtain 21.0% accuracy. However, when it is combined with $\varphi^{\mathcal{W}}$ (28.4%), the accuracy improves to 29.7%. Compared to the results in Tab. 4, 29.7% is the highest accuracy obtained using unsupervised embeddings. We regard these results as a very encouraging evidence that Word2Vec representations can indeed be made more discriminative for fine-grained zero-shot learning by integrating a fine-grained text corpus directly to the output embedding learning problem.

Hierarchical Embeddings. The hierarchical organization of concepts typically embodies a fair amount of hidden information about language, such as synonymy, semantic relations, etc. Therefore, semantic relatedness defined by hierarchical distance between classes can form numerical vectors to be used as output embeddings for zero-shot learning. We build ontological relationships between our classes using the WordNet [36] taxonomy. Due to its large size, WordNet encapsulates all of our AWA and Dog classes. For CUB, the high level bird species, i.e. albatross, appear as synsets in WordNet, but the specific bird names, i.e. black-footed albatross, are not always present. Therefore we take the hierarchy up to high level bird species as-is and we assume the specific bird classes are all at the bottom of the hierarchy located with the same distance to their immediate ancestors. The WordNet hierarchy contains 319 nodes for CUB (200 classes), 104 nodes for AWA (50 classes) and 163 nodes for Dogs (113 classes). We measure the distance between classes using the similarity measures from Sec 4.1.

While as shown in Fig. 2 different hierarchical similarity measures have very different behaviors on each dataset. The best performing $\varphi^{\mathcal{H}}$ obtains 51.2% (Tab. 3) accuracy

					AWA		CUB		Dogs	
$\varphi^{\mathcal{A}}$	$\varphi^{\mathcal{W}}$	$\varphi^{\mathcal{G}}$	$\varphi^{\mathcal{B}}$	$\varphi^{\mathcal{H}}$	*cnc*	*cmb*	*cnc*	*cmb*	*cnc*	*cmb*
	✓			✓	53.9	55.5	28.2	29.4	23.5	26.6
		✓		✓	**60.1**	59.5	28.5	**29.9**	23.5	26.7
			✓	✓	49.4	49.2	26.4	27.7	**35.1**	28.2
✓	✓			✓	71.3	73.5	45.1	51.0	-	-
✓		✓		✓	73.3	**73.9**	42.2	**51.7**	-	-
✓			✓	✓	69.4	71.1	40.9	51.5	-	-

Table 5. Attribute ensemble results for all datasets. $\varphi^{\mathcal{H}}$: lin for CUB, path for AWA and Dogs. Top part shows combination results of unsupervised embeddings and bottom part integrates supervised embeddings to the rest (Input embeddings: 1K-GOOG).

on AWA which reaches our $\varphi^{0,1}$ (52.0%) and improves $\varphi^{\mathcal{B}}$ (44.9%) significantly. On CUB, $\varphi^{\mathcal{H}}$ obtains 20.6% (Tab. 3) which remain below our $\varphi^{0,1}$ (37.8%) and approaches $\varphi^{\mathcal{B}}$ (22.1%). On the other hand, on Dogs $\varphi^{\mathcal{H}}$ obtains 24.3% (Tab. 3) which is significantly higher than the unsupervised text embeddings $\varphi^{\mathcal{W}}$ (19.6%) and $\varphi^{\mathcal{G}}$ (17.8%).

Combining Output Embeddings. In this section, we combine output embeddings obtained through human annotation ($\varphi^{\mathcal{A}}$), from text ($\varphi^{\mathcal{W},\mathcal{G},\mathcal{B}}$) and from hierarchies ($\varphi^{\mathcal{H}}$).[5] As a reference, Tab. 3 summarizes the results obtained using one output embedding at a time. Our intuition is that because the different embeddings attempt to encapsulate different information, accuracy should improve when multiple embeddings are combined. We can observe this complementarity either by simple concatenation (*cnc*) or systematically combining (*cmb*) output embeddings (Sec.3.3) also known as early/late fusion [2]. For *cnc*, we perform full SJE training and cross-validation on the concatenated output embeddings. For *cmb*, we learn joint embeddings W_k for each output separately (which is trivially parallelized), and find ensemble weights α_k via cross-validation. In contrast to the *cnc* method, no additional joint training is used, although it can improve performance in practice. We observe (Tab. 5) in almost all cases *cmb* outperforms *cnc*.

We analyze the combination of unsupervised embeddings ($\varphi^{\mathcal{W},\mathcal{G},\mathcal{B},\mathcal{H}}$). On AWA, $\varphi^{\mathcal{G}}$ (58.8%, Tab. 3) combined with $\varphi^{\mathcal{H}}$ (51.2%, Tab. 3), we achieve 60.1% (Tab. 5) which improves the SoA (48.5%, Tab. 2) by a large margin. On CUB, combining $\varphi^{\mathcal{G}}$ (24.2%, Tab. 3) with $\varphi^{\mathcal{H}}$ (20.6%, Tab. 3), we get 29.9% (Tab. 5) and improve the supervised-SoA (26.9%, Tab. 2). Supporting our initial claim, unsupervised output embeddings obtained from different sources, i.e. text vs hierarchy, seem to be complementary to each other. In some cases, *cmb* performs worse than *cnc*; e.g. 28.2% versus 35.1% when using $\varphi^{\mathcal{B}}$ with $\varphi^{\mathcal{H}}$ on Dogs. In most other cases *cmb* performs equivalent or better. Combining supervised ($\varphi^{\mathcal{A}}$) and unsupervised

[5] We empirically found that the hierarchical embeddings $\varphi^{\mathcal{H}}$ consistently improved performance when combined or concatenated with other embeddings. Therefore, we report results using $\varphi^{\mathcal{H}}$ by default.

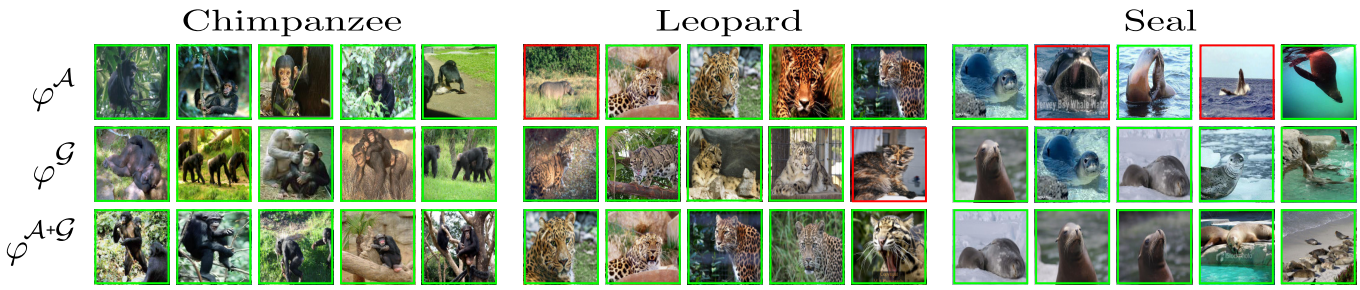

Figure 3. Highest ranked 5 images for *chimpanzee*, *leopard* and *seal* (AWA) using $\varphi^{\mathcal{A}}$, $\varphi^{\mathcal{G}}$ and $\varphi^{\mathcal{G}+\mathcal{A}}$. For *chimpanzee*, $\varphi^{\mathcal{A}}$ ranks chimpanzees on trees at the top, whereas $\varphi^{\mathcal{G}}$ models the social nature of the animal ranking a group of chimpanzees highest, $\varphi^{\mathcal{G}+\mathcal{A}}$ synthesizes both aspects. For *leopard* $\varphi^{\mathcal{A}}$ puts an emphasis on the head, $\varphi^{\mathcal{G}}$ seems to place the animal in the wild. In case of *seal*, $\varphi^{\mathcal{A}}$ retrieves images related to *water*, whereas $\varphi^{\mathcal{G}}$ adds more context by placing seals in the icy natural environment and $\varphi^{\mathcal{G}+\mathcal{A}}$ combines both.

supervision	method	**AWA**	**CUB**	**Dogs**
unsupervised	SJE (best from Tab. 5)	60.1	29.9	**35.1**
supervised	SJE (best from Tab. 5)	**73.9**	**51.7**	–
	AHLE [2]	49.4	27.3	–

Table 6. Summary of best zero-shot learning results with SJE with or without supervision along with SoA.

embeddings ($\varphi^{\mathcal{W},\mathcal{G},\mathcal{B},\mathcal{H}}$) shows a similar trend. On AWA, combining $\varphi^{\mathcal{A}}$ (66.7%, Tab. 3) with $\varphi^{\mathcal{G}}$ and $\varphi^{\mathcal{H}}$ leads to 73.9% (Tab. 5) which significantly exceeds the SoA (48.5%, Tab. 2). On CUB, combining $\varphi^{\mathcal{A}}$ with $\varphi^{\mathcal{G}}$ and $\varphi^{\mathcal{H}}$ leads to 51.7% (Tab. 5), improving both the results we obtained with $\varphi^{\mathcal{A}}$ (50.1%, Tab. 3) and the supervised-SoA (26.9%, Tab. 2). We have shown with these experiments that output embeddings obtained through human annotation can also be complemented with unsupervised output embeddings using the SJE framework.

Qualitative Results. Fig. 3 shows top-5 highest ranked images for classes *chimpanzee*, *leopard* and *seal* that are selected from 10 test classes of AWA. We use GOOG as input embeddings and as output embeddings we use supervised $\varphi^{\mathcal{A}}$, the best performing unsupervised embedding on AWA ($\varphi^{\mathcal{G}}$), and the combination of the two ($\varphi^{\mathcal{G}+\mathcal{A}}$). For the class *chimpanzee*, $\varphi^{\mathcal{A}}$ emphasizes that chimpanzees live on trees, which is among the list of attributes. On the other hand, $\varphi^{\mathcal{G}}$ models the social nature of the animal, ranking a group of chimpanzees interacting with each other at the highest. Indeed this information can easily be retrieved from Wikipedia. $\varphi^{\mathcal{G}+\mathcal{A}}$ synthesizes both aspects. Similarly, for *leopard* $\varphi^{\mathcal{A}}$ puts an emphasis on the head where we can observe several of the attributes, i.e. color, spotted, whereas $\varphi^{\mathcal{G}}$ seems to place the animal in the wild. $\varphi^{\mathcal{G}+\mathcal{A}}$ combines both aspects. In case of class *seal*, $\varphi^{\mathcal{A}}$ retrieves images related to *water* and ranks whales and seals highest, whereas $\varphi^{\mathcal{G}}$ adds more context by placing seals in the icy natural environment and within groups. Finally, $\varphi^{\mathcal{G}+\mathcal{A}}$ ranks seal-shaped animals on ice, close to water and within groups the highest. We find these qualitative results interesting as they depict how (1) unsupervised embeddings capture nameable semantics about objects and (2) different output embeddings are semantically complementary for zero-shot learning.

6. Conclusion

We evaluated the Structured Joint Embedding (SJE) framework on supervised attributes and unsupervised output embeddings obtained from hierarchies and unlabeled text corpora. We proposed a novel weakly-supervised label embedding technique. By combining multiple output embeddings (*cmb*), we established a new SoA on AWA (73.9%, Tab. 6) and CUB (51.7%, Tab. 6). Moreover, we showed that unsupervised zero-shot learning with SJE improves the SoA, to 60.1% on AWA and 29.9% on CUB, and obtains 35.1% on Dogs (Tab. 6).

We emphasize the following take-home points: (1) Unsupervised label embeddings learned from text corpora yield compelling zero-shot results, outperforming previous supervised SoA on AWA and CUB (Tab. 2 and 3). (2) Integrating specialized text corpora helps due to incorporating more fine-grained information to output embeddings (Tab. 4). (3) Combining unsupervised output embeddings improve the zero-shot performance, suggesting that they provide complementary information (Tab. 5). (4) There is still a large gap between the performance of unsupervised output embeddings and human-annotated attributes on AWA and CUB, suggesting that better methods are needed for learning discriminative output embeddings from text. (5) Finally, supporting [2, 47], encoding continuous nature of attributes significantly improve upon binary attributes for zero-shot classification (Tab. 2).

As future work, we plan to investigate other methods to combine multiple output embeddings and to improve the discriminative power of unsupervised and weakly-supervised label embeddings for fine-grained classification.

Acknowledgments

This work was supported in part by ONR N00014-13-1-0762, NSF CMMI-1266184, Google Faculty Research Award, and NSF Graduate Fellowship.

References

[1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label embedding for attribute-based classification. In *CVPR*, 2013. 1, 2, 3, 5

[2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for image classification. *arXiv:1503.08677*, 2015. 2, 3, 5, 6, 7, 8

[3] S. Bengio, J. Weston, and D. Grangier. Label embedding trees for large multi-class tasks. In *NIPS*, 2010. 1, 2

[4] Y. Bengio, R. Ducharme, P. Vincent, and C. Janvin. A neural probabilistic language model. *JMLR*, 2003. 7

[5] E. Blanchard, M. Harzallah, H. Bri, P. Kuntz, and R. C. Pauc. A typology of ontology-based semantic measures. In *EMOI-INTEROP'05 workshop, at CAiSE'05*, 2005. 4

[6] H. Chen, A. Gallagher, and B. Girod. What's in a name? first names as facial attributes. In *CVPR*, 2013. 2

[7] 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

[8] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, June 2013. 1

[9] T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *JAIR*, 1995. 2

[10] A. Dosovitskiy, J. T. Springenberg, M. Riedmiller, and T. Brox. Discriminative unsupervised feature learning with convolutional neural networks. *arXiv:1406.6909*, 2014. 7

[11] M. Douze, A. Ramisa, and C. Schmid. Combining attributes and Fisher vectors for efficient image retrieval. In *CVPR*, 2011. 2

[12] K. Duan, D. Parikh, D. J. Crandall, and K. Grauman. Discovering localized attributes for fine-grained recognition. In *CVPR*, 2012. 2

[13] A. Farhadi, I. Endres, and D. Hoiem. Attribute-centric recognition for cross-category generalization. In *CVPR*, 2010. 2

[14] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. *CVPR*, 2009. 1

[15] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 1, 2

[16] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, 2013. 2, 3

[17] 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

[18] Y. Goldberg and O. Levy. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv:1402.3722*, 2014. 4

[19] Z. Harris. Distributional structure. *Word*, 10(23), 1954. 2, 4

[20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning (2nd Ed.)*. Springer Series in Statistics. Springer, 2008. 2, 3

[21] D. Hsu, S. Kakade, J. Langford, and T. Zhang. Multi-label prediction via compressed sensing. In *NIPS*, 2009. 2

[22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015. 5

[23] 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 preprint arXiv:1408.5093*, 2014. 5

[24] J. J. Jiang and D. W. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *CoRR*, 1997. 2, 4

[25] P. Kankuekul, A. Kawewong, S. Tangruamsub, and O. Hasegawa. Online incremental attribute-based zero-shot learning. In *CVPR*, 2012. 2

[26] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Stanford dogs dataset. http://vision.stanford.edu/aditya86/ImageNetDogs/. 2, 5

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*. 2012. 1, 2, 5

[28] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. choi, A. Berg, and T. Berg. Baby talk: understanding and generating simple image descriptions. In *CVPR*, 2011. 2

[29] C. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. In *TPAMI*, 2013. 1, 2, 3, 5

[30] C. Leacock and M. Chodorow. Filling in a sparse training space for word sense identification, 1994. 2

[31] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. In *Proc. of the IEEE*, 1998. 1

[32] D. Lin. An information-theoretic definition of similarity. In *ICML*, 1998. 2, 4

[33] J. Liu, B. Kuipers, and S. Savarese. Recognizing human actions by attributes. In *CVPR*, 2011. 2

[34] T. E. J. Mensink, E. Gavves, and C. G. M. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014. 2

[35] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013. 2, 4, 6

[36] G. A. Miller. Wordnet: a lexical database for english. *CACM*, 38:39–41, 1995. 2, 7

[37] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *CoRR*, 2013. 2

[38] S. Nowozin and C. Lampert. *Structured Learning and Prediction in Computer Vision*. Foundations and Trends in Computer Graphics and Vision, 2011. 1

[39] V. Ordonez, G. Kulkarni, and T. Berg. Im2Text: Describing images using 1 million captioned photographs. In *NIPS*, 2011. 2

[40] M. Palatucci, D. Pomerleau, G. Hinton, and T. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009. 3

[41] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 1, 2

[42] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *EMNLP*, 2014. 2, 4

[43] F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *CVPR*, 2007. 2, 5

[44] P. Resnik. Using information content to evaluate semantic similarity in a taxonomy. In *IJCAI*, 1995. 2

[45] M. Rohrbach, M. Stark, and B.Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *CVPR*, 2011. 2, 3

[46] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps here – and why? Semantic relatedness for knowledge transfer. In *CVPR*, 2010. 2

[47] M. Rohrbach, M. Stark, G. Szarvas, and B. Schiele. Combining language sources and robust semantic relatedness for attribute-based knowledge transfer. *Trends and Topics in Computer Vision*, 2012. 3, 8

[48] W. J. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *CVPR*, 2012. 2

[49] B. Siddiquie, R. Feris, and L. Davis. Image ranking and retrieval based on multi-attribute queries. In *CVPR*, 2011. 2

[50] R. Socher, M. Ganjoo, H. Sridhar, O. Bastani, C. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, 2013. 3

[51] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 5

[52] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *JMLR*, 2005. 1, 2, 3

[53] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms, 2008. 5

[54] C. Wah, S. Branson, P. Perona, and S. Belongie. Multiclass recognition and part localization with humans in the loop. In *ICCV*, 2011. 5

[55] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, Caltech, 2010. 2, 3

[56] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: Learning to rank with joint word-image embeddings. *ECML*, 2010. 1, 2, 3

[57] B. Yao, X. Jiang, A. Khosla, A. L. Lin, L. J. Guibas, and F.-F. Li. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 2

[58] X. Yu and Y. Aloimonos. Attribute-based transfer learning for object categorization with zero or one training example. In *ECCV*, 2010. 2