

Describing Unseen Classes by Exemplars: Zero-shot Learning Using Grouped Simile Ensemble

Yang Long
The University of Sheffield
ylong2@sheffield.ac.uk

Ling Shao
The University of East Anglia
ling.shao@ieee.org

Abstract

Learning visual attributes is an effective approach for zero-shot recognition. However, existing methods are restricted to learning explicitly nameable attributes and cannot tell which attributes are more important to the recognition task. In this paper, we propose a unified framework named Grouped Simile Ensemble (GSE). We claim our contributions as follows. 1) We propose to substitute explicit attribute annotation by similes, which are more natural expressions that can describe complex unseen classes. Similes do not involve extra concepts of attributes, i.e. only exemplars of seen classes are needed. We provide an efficient scenario to annotate similes for two benchmark datasets, AwA and aPY. 2) We propose a graph-cut-based class clustering algorithm to effectively discover implicit attributes from the similes. 3) Our GSE can automatically find the most effective simile groups to make the prediction. On both datasets, extensive experimental results manifest that our approach can significantly improve the performance over the state-of-the-art methods.

1. Introduction

Zero-shot recognition is an attractive new task that has recently aroused increasing attentions [21, 28, 5, 33, 30]. It has made it possible to recognise a new category without acquiring training examples beforehand. Compared to traditional methods, zero-shot techniques leverage intermediate semantic models that are shareable to both seen and unseen classes. Such a technique can have wide real-world applications. First, we can now recognise many novel categories for which the visual instances are difficult to be obtained. For example, one may wish to recognise rare animals using only textual descriptions in the book. Second, in the big-data era, the number of required target categories can be enormous. Zero-shot learning (ZSL) can effectively alleviate the burden of collecting training data. Third, for many traditional methods, it is inevitable to retrain the whole

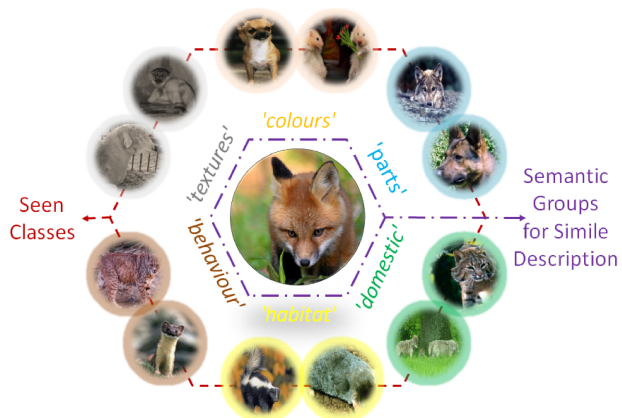


Figure 1. A new class can be described by similes of seen classes without extra attribute concepts involved. We use semantic grouping to make the similes more discriminative. Similes are more natural to describe complex concepts, e.g. *behaviour* or *domestic*.

model again when we need to add new categories. In zero-shot approaches, the trained model can be shareable for any newly added categories so as to avoid re-training.

One of the fundamental premises for existing ZSL frameworks is the effectiveness of the semantic models. Previous methods [21, 17, 33, 8] widely adopt human nameable attributes as the semantic representations and demonstrate promising results. However, using human nameable attributes can also suffer from several problems. Firstly, deciding an attribute list for ZSL is an ambiguous task. It is easy to consider some visual semantic groups, such as *colours*, *textures* and *parts*. However, more complex attributes, e.g. some intangible visual effects, can be hardly described by specific words. Secondly, the designed attribute list is not guaranteed to be discriminative for ZSL. For one thing, semantic attributes may not be visually describable, e.g. *domestic* and *carnivorous* in the AwA dataset. Consequently, we can hardly find a converged model for such attributes due to the large variety of visual patterns. Another common issue is known as the *correlation problem* [18]. Namely, different attributes can be highly correlated to each other and are always present or absent together among

the whole training set. It then becomes impossible to differentiate these attributes from each other since they share the same positive and negative samples.

Simile is a figure of speech that directly compares two exemplars. In this paper, we propose to use similes instead of explicit attributes. Our idea is motivated by [20] that makes use of similes to describe human faces, *e.g.* the glasses on the query face looks like Harry potter’s. However, only similes are not competent for ZSL tasks due to the number of *seen* classes is limited compared to faces. Therefore, we go one step further: we propose a novel graph-cut algorithm that can discover the shared attributes possessed by the similes of exemplars without explicit names. We call such attributes *Implicit Attributes*. Furthermore, to achieve more discriminative semantic models for ZSL tasks, our similes are under different semantic groups, *i.e.* from various aspects such as *colours*, *shapes* and *parts*. We propose a unified framework named Grouped Simile Ensemble (GSE) that can recognise unseen objects by an ensemble model of simile groups. Our method aims to automatically balance the weights between different simile groups, just like we humans can easily find more important attributes to distinguish things. For example, it is easier to differentiate a *panda* from a *bear* by *colours* rather than *shapes*.

Our framework can be briefly summarised as follows. Firstly, we manually annotate both seen and unseen categories by similes under different groups, from which we can discover the implicit attributes by our graph-cut algorithm. We then train our GSE model using training images and the discovered implicit attributes. During the test, our GSE model can find the most important attributes to make predictions for unseen classes. We claim four desired properties of the proposed framework:

- Similes do not involve many additional concepts like explicit attributes. Only the names of seen classes are used. Also, a simile is visually representable by exemplars. It is natural to describe complex visual appearances by the similarities to training exemplars.
- Our graph-cut algorithm is aware of how many implicit attributes exist in the similes. Each attribute is trained by non-overlapped exemplars to prevent the correlation problem.
- Our GSE model can automatically weigh the significance of different simile groups during the test. On two benchmark datasets, our method achieves state-of-the-art ZSL recognition performance.

The remaining paper is arranged as follows: in Section 2, we review related zero-shot methods; in Section 3, we illustrate our framework and derive the formulations of our ensemble model; we provide extensive evaluations in Section 4; finally, we conclude our findings in Section 5.

2. Related Work

Zero-shot learning frameworks The key technique of ZSL is to find an intermediate clue that can generalise to unseen classes. Larochelle *et al.* [22] propose a template-based framework that can depict new classes by manually defined templates. Recently, learning visual attributes [12, 29] gains popularity. In [21], attribute classification is utilised as a mid-level task. During the test, the posterior probability of each attribute is estimated separately by pre-trained classifiers; and the final prediction is made by Maximum a Posteriori (MAP) criteria. Since attribute classifiers are trained separately, such frameworks suffer from the correlation problem [18] and unreliable annotations [17]. In [2], Akata *et al.* propose an embedding-based framework that regards all of the defined attributes as a whole representation. Many recent approaches adopt such an embedding manner and achieve promising results [13, 4, 33, 15, 7, 19, 39, 8, 23]. Besides, similarity-based frameworks also adopt the embedding approach [24, 40, 41, 34, 8, 25]. But the semantic space aims to associate unseen to seen classes. Although these methods have empirically shown improved performance, their embeddings are not human-understandable like the attribute-based methods, *e.g.* they cannot tell which attribute makes the recognition failure like [11]. In comparison to existing methods, our method adopts the advantages of using embedding approaches that can effectively map visual features to the semantic spaces. Furthermore, our embeddings are also interpretable since each simile group has an explicit meaning.

Variations of Semantic information ZSL recognition relies on how to represent unseen classes by prior human knowledge accurately. The representation must be **1)** generalisable, *i.e.* the trained model on seen classes is also effective on unseen classes; **2)** visual-related, the gap between the semantic and visual spaces should be small enough to train a stable model. According to these requirements, learning visual attributes has gain most popularity [21, 29, 38, 27, 14, 16, 8]. However, attribute annotations are very expensive, especially for image-level tasks. Also, the involved attributes in the list require careful design. Different datasets often cannot share the learnt attribute models. Such issues make using attributes impractical. As a low-cost solution, text-based semantic features is proposed [32, 10, 37, 26]. However, the textual description from the Internet can be noisy and not directly related to the visual appearance. Another mainstream of semantic representations is similarity-based. Class-wise similarities can be obtained by either human annotators [38, 20] or based on the textual descriptions [40, 41]. Our simile description also shares the idea of similarity comparison. However, none of the existing methods make use of grouping so that the similes can be precisely interpreted. Furthermore, we require the annotators try to make similes based on the vi-

sual appearance rather than the semantics so that the visual-semantic gap can be mitigated.

3. Approach

We first introduce how to annotate classes by similes. Then we formalise the whole framework. The first step of our approach is to discover the implicit attributes from the similes a graph-cut algorithm. Our second step is to train a robust GSE model. Finally, we show how to make predictions using the GSE model during the ZSL test.

3.1. Simile Annotation

We aim to annotate both *seen* and *unseen* classes by similes of *seen* exemplars. We illustrate the annotation process in Fig. 2. For each target class under annotating, we ask the annotator first meditate its visual appearance from a semantic aspect for ten seconds, e.g. *colour*, *parts*, or, *shape*. Afterwards, our program starts to flash random exemplars from different *seen* classes, ten images per time. The annotator is asked to choose the most similar exemplars. We accumulate the choices and find the top k most similar classes. Such a process is repeated for all classes under different simile groups. In average, we present ten exemplars from each *seen* class. Key statistics of our simile annotation is summarised in Table 1.

Table 1. Statistics of simile annotation on AwA and aPY datasets.

items	AwA	aPY
Number of Classes	50	32
Number of Simile Groups	9	5
Number of Images per Flash	10	10
Average Annotating Time	2.5 hours	1 hour

3.2. Preliminary

Problem: The training set is in pairs of samples and labels: $(x_1, y_1), \dots, (x_N, y_N) \subseteq \mathcal{X} \times \mathcal{Y}$, where \mathcal{X} is an arbitrary feature space and $y_n \in \{1, \dots, C\}$ consists of C discrete categories. In the test domain, only names of L *unseen* classes are provided without any instances, i.e. $\mathcal{Z} = \{z_1, \dots, z_L\}$. The goal is to learn a classifier, $f : \mathcal{X} \rightarrow \mathcal{Z}$. It is noticeable that $\mathcal{Z} \cap \mathcal{Y} = \emptyset$. Such a problem is known as the Zero-shot classification.

Discovering implicit attributes from similes: After simile annotation in Section 3.1, any class $j \in \mathcal{Y} \cup \mathcal{Z}$ can be interpreted by a set of similarity-based exemplars from the training set, i.e. $\mathcal{NN}_j \in \mathcal{Y}$, which can form an undirected graph. Using graph-cut, we can discover what are the implicit attributes that make the classes similar to each other. This is conducted under G different simile groups. For each group: $f_1^{(g)} : \mathcal{NN}^{(g)} \rightarrow \mathcal{A}^{(g)}$. As a result, each category gains an attribute signature in each simile group:

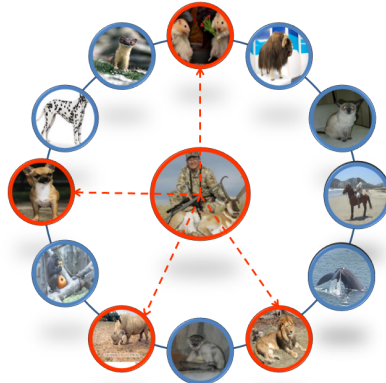


Figure 2. An example of simile annotation process: whose *colour* is similar to *antelope*. We randomly show exemplars from *seen* classes to the annotator. The annotator is asked to choose a number of most similar exemplars. We achieve averaged similarities by repeating such a process several times using different random exemplars.

$\mathcal{A}_j^{(g)} = (a_1, \dots, a_{m_g}) \in \mathbb{R}^{m_g}$, where $j \in \mathcal{Y} \cup \mathcal{Z}$, and m_g is the total number of discovered implicit attributes.

Base feature extraction and GSE: Low-level features are extracted and concatenated to form a base visual space. We train ensemble models for different simile groups. Each model aims to embed the visual features from *seen* classes to their corresponding implicit attribute space: $f_2^{(g)} : \mathcal{X} \rightarrow \mathcal{A}^{(g)}$.

Zero-shot classification: Given a query instance, it is firstly represented by GSE using f_2 . Our final ensemble mechanism aims to make predictions for instances from both *seen* and *unseen*: $f_3 : (\mathcal{A}^{(1)}, \dots, \mathcal{A}^{(G)}) \rightarrow \mathcal{Y} \cup \mathcal{Z}$.

3.3. Implicit Attributes Discovery

Implicit attributes are shared attributes of a group of exemplars without explicit names. Our implicit attributes are under different semantic groups, such as colour, shape, and texture. For instance, one attribute could be a mixture of colours that is possessed by *zebra*, *panda*, and *dalmatian*. Furthermore, some implicit attributes are even intangible but can be only expressed by similes. The number of such implicit attributes can be arbitrary. Our motivation of using graph-cut aims to scope the various implicit attributes by several clusters. Within each cluster, the simile of exemplars can have very close visual attributes so that we can train stable models for them.

The simile annotation introduced in Section 3.1 naturally satisfies a class-level undirected k -nearest neighbour graph. In the graph, each vertex v_c corresponds to a class from $\mathcal{Y} \cup \mathcal{Z}$. Fig. 3 illustrates such a problem intuitively. v_{c_1} and v_{c_2} are connected if and only if v_{c_2} is a member of similes

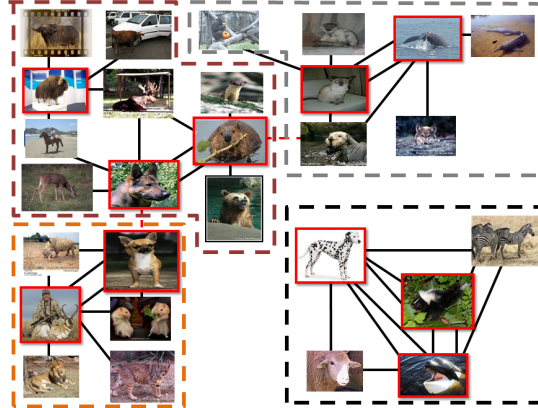
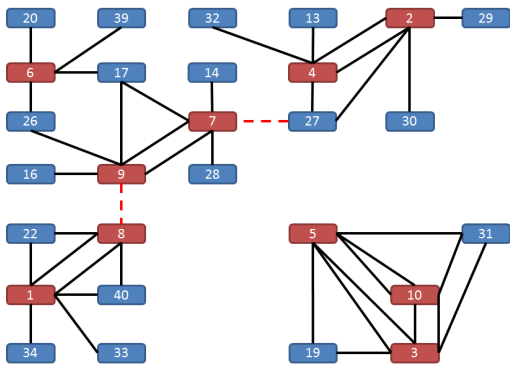


Figure 3. Implicit attribute discovery. Under each simile group, the associated exemplars of each class satisfy a k -nn graph (left). Red vertices indicate *unseen* classes. Our algorithm can cut the weakest edges and cluster the classes with similar implicit attributes (right).

\mathcal{NN}_{c_1} of class c_1 . In this way, if v_{c_1} and v_{c_2} are mutually nearest neighbours, the weight of the edge in between is 2. Similarly, if v_{c_1} and v_{c_2} are not mutually nearest neighbours but connected, the weight of the edge in between is 1. Since $\mathcal{NN} \in \mathcal{Y}$, the achieved graph has the same dimension as the number of *seen* classes: $W \in \{0, 1, 2\}^{C \times C}$. Cutting such a graph clusters the seen classes. Each cluster possesses a visually similar implicit attribute.

According to [36], graph cut can be approximated through the spectral clustering approach in order to improve the efficiency. The unnormalised graph Laplacian matrix is defined as:

$$L = D - W, \quad (1)$$

where D is a degree matrix with d_1, \dots, d_C on the diagonal, and each d_c is defined as:

$$d_c = \sum_{c_i=1}^C W_{cc_i}. \quad (2)$$

The number of 0s in the eigenvalues of L indicates how many subsets are disconnected. However, in practice, we can decide whether it is necessary to cut those weak connections further by visualising the distribution of remaining non-zero eigenvalues. In Fig. 4, we can clearly see that the distribution of the eigenvalues from 40 *seen* classes can be roughly divided into four more groups. Adding on the zero eigenvalue, the optimal number of clusters is 5. Finally, classes are clustered by the k -means algorithm on the first m eigenvectors, where m equals the optimal number of implicit attributes ($m = 5$ in this case).

After graph-cut, each class $c \in \mathcal{Y} \cup \mathcal{Z}$ can be soft-assigned to the discovered implicit attributes according to the original similes \mathcal{NN}_c , *i.e.* $\mathcal{A}_c = (a_1, \dots, a_m) \in \mathbb{R}^m$. Each dimension indicates the prior probability of each implicit attribute presenting in the class. We repeat such processes for G simile groups.

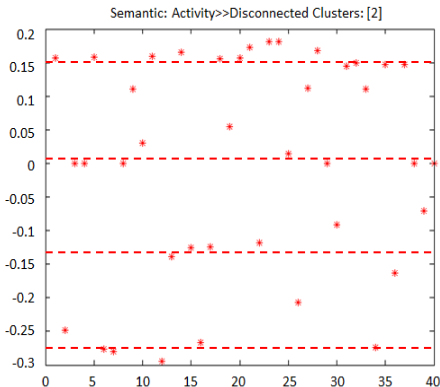


Figure 4. Visualisation of eigenvalues. We demonstrate the example from the simile group of *activity* in the AwA dataset. The k -NN graph of similes has two disconnected subsets (one zero eigenvalue). However, we could find roughly four more layers, which indicates that the optimal value for m is 5.

3.4. Grouped Simile Ensemble

The primary purpose of using grouped simile ensemble is to find the most effective attributes for different tasks. Our main idea is to observe the visual data from various semantic aspects. Specifically, we first extract various low-level visual features from the images and concatenate them as base features. We then train embedding functions to map the base features to different simile groups. Such a framework satisfies the spirit of ensemble model [9] that a single input can be interpreted with various aspects, *i.e.* simile groups. There are three potential advantages of using ensemble models. 1) The limited training examples now can be utilised multiple times for different simile groups. 2) The difficulty of attribute classification task is lower since the number of implicit attributes in each cluster is much smaller than that of the whole attribute list. Moreover, our pre-process of graph-cut makes the boundaries between im-

pllicit attributes more discriminable. 3) the ensemble of base features provides rich representations which make it easier to find discriminative dimensions to satisfy the hypothesis.

The whole ensemble learning task can be defined as a Bayesian probabilistic setting. For each simile group g , we use the discovered implicit attributes as labels to train a hypothesis for supervised multi-label classification. Each hypothesis $h^{(g)}$ embeds the input base visual feature in \mathcal{X} into an implicit attribute space $\mathcal{A}^{(g)}$ satisfy a conditional probability distribution:

$$h^{(g)}(\mathcal{X}) = p\left(f_2^{(g)}(\mathcal{X}) = \mathcal{A}^{(g)} | \mathcal{X}, h^{(g)}\right). \quad (3)$$

The whole GSE model consists of all of the hypotheses in \mathcal{H} , where $\mathcal{H} = \{h^{(1)}, \dots, h^{(G)}\}$, in which each multi-class classifier in each group $h^{(g)}(x)$ possesses a basis. Given a test sample \hat{x} and the training set \mathcal{X} , the problem of predicting the overall implicit attributes of all simile groups can be expressed as weighted sum over all hypotheses:

$$\begin{aligned} p(f_2(\hat{x}) = \hat{\mathcal{A}} | \hat{x}, \mathcal{X}) &= \prod_{g=1}^G h^{(g)}(\hat{x}) p\left(h^{(g)} | \mathcal{X}\right) \\ &\propto \frac{1}{G} \sum_{g=1}^G \log h^{(g)}(\hat{x}) p\left(h^{(g)} | \mathcal{X}\right), \end{aligned} \quad (4)$$

where $\hat{\mathcal{A}} = (\hat{\mathcal{A}}^{(1)}, \dots, \hat{\mathcal{A}}^{(G)})$ is the overall implicit attributes of \hat{x} by concatenating all of the simile groups. During training, \mathcal{A} and \mathcal{X} are in pairs. Because there is no prior knowledge about which simile group will work better during the test, we assume the simile groups are i.i.d.. taking the Bayes rule we get:

$$p(h^{(g)} | \mathcal{X}) \propto p(\mathcal{X} | h^{(g)}) p(h^{(g)}), \quad (5)$$

where $p(h^{(g)})$ is assumed equal to one, the performance of each classier $p(\mathcal{X} | h^{(g)})$ can be estimated during training. However, for ZSL tasks, \hat{x} is from unknown classes. The prior training score of $p(\mathcal{X} | h^{(g)})$ may not hold during the test. For an intuitive instance, the *colours* simile group may work better on the training set to distinguish *panda* from *bear*. However, to test with unseen instances *zebra* and *dalmatian*, the *shapes* group is more discriminative. In this paper, we employ the maximum-a-posteriori criteria to make an approximate estimation that can automatically find the most effective simile group for unseen classes.

Specifically, we employ LDA [6] to estimate $p(\mathcal{X} | h^{(g)})$ on the training set so that visual features possessing the same implicit attributes can be projected into a more compact space. Each LDA model $h^{(g)}$ is trained with the g^{th} group of implicit attributes $\mathcal{A}^{(g)}$. We empirically show the advantages of using such embedding in our later experiments. During the test, an unseen instance can be mapped

to the embedding hypotheses space by taking the log probability of the maximum likelihood decision rule:

$$\begin{aligned} \hat{\mathcal{A}} &= \arg \max_{\mathcal{A}} \sum_{g=1}^G \log h^{(g)}(\hat{x}) p(\mathcal{X} | h^{(g)}) p(h^{(g)}) \\ &\approx \arg \min_{\mathcal{A}} \sum_{g=1}^G \|h^{(g)}(\hat{x}) - NN_{\mathcal{A}^{(g)}}(h^{(g)}(\hat{x}))\|_F^2, \end{aligned} \quad (6)$$

where $NN(\cdot)$ is a nearest neighbour searching from the embedding hypothesis space $\mathcal{A}^{(g)}$ of the g^{th} group, and $\log p(\mathcal{A} | \hat{a}) \propto \sum_{g=1}^G \|h^{(g)}(\hat{x}) - NN_{\mathcal{A}^{(g)}}(h^{(g)}(\hat{x}))\|_F^2$. Intuitively, weights of different simile groups are automatically determined by the Frobenius Norm distances. As a result, the maximum likelihood decision can find the optimal ensemble of implicit attributes of the test instance under each simile group.

3.5. Zero-shot Classification

After predicting the implicit attributes $\hat{\mathcal{A}}$, we can make classify a test instance \hat{x} by comparing $\hat{\mathcal{A}}$ to the reference attributes that we have achieved through the graph-cut. As introduced in Section 3.3, we have obtained a unique attribute signature \mathcal{A}_j for both *seen* and *unseen* classes, *i.e.* $j \in \mathcal{Y} \cup \mathcal{Z}$. Because $\hat{\mathcal{A}}$ is i.i.d. given its class, the bias towards the *seen* classes can be eliminated. Therefore, we can extend the previous ZSL setting that restricts to test by *unseen* instances. In this paper, our method can classify both *seen* and *unseen* instances at the same time. In order to show the power of our GSE model and the advantages of using implicit attributes, we simply adopt the most straightforward NN classifier:

$$\hat{C} = \arg \min_j \|\hat{\mathcal{A}} - NN(\mathcal{A}_j)\|^2, \quad (7)$$

where $\hat{C}, j \in \mathcal{Y} \cup \mathcal{Z}$. Again, if some implicit attributes are incorrectly predicted or annotated, the Frobenius Norm distances can suppress such noises to some extends.

4. Experiments and Results

Datasets We evaluate our method on two ZSL benchmark datasets, Animals with Attributes (AwA) [21], and aPascal&aYahoo (aPY) [11]. AwA contains 30,475 images of 50 wild animal classes. In aPY, there are totally 15339 images from more various categories than AwA, including humans, artificial objects, buildings, as well as animals, which makes the recognition task more challenging.

Visual Features In order to compare to as many existing methods as possible, we adopt both low-level features that are provided by the datasets and deep features that are published by [40]. The low-level features include both local and global descriptors, such as SIFT, PHOG, Colour histogram,

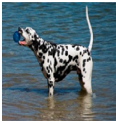

Dalmatian	Colour	Texture	Shape	Part	Activity
	Panda	Giraffe	Bobcat	Zebra	German Shepherd
	Killer Whale	Cow	Leopard	Wolf	Persian Cat
	Zebra	Leopard	Wolf	Lion	Siamese Cat
	Skunk	Skunk	Zebra	Leopard	Collie
	Sheep	Deer	Deer	German Shepherd	Horse
Bicycle	Shape	Texture	Plant	Part	Material
	Motorbike	Motorbike	Motorbike	Motorbike	Motorbike
	Chair	Train	Bus	Person	Boat
	Sofa	TV Monitor	Car	Building	Bus

Figure 5. Examples of images annotated by similes under different groups in AwA (upper) and aPY (lower).

textual and edge descriptors. Local features are coded by Bag-of-words. We concatenate such low-level features as our base features, on which we perform PCA that results in 9751-dimensional representations. The deep features are extracted by VGG-19 that results in 4096-dimensional representations.

Attributes and Semantic Groups Our GSE does not use the provided explicit attributes in AwA and aPY. On AwA, we adopt the same semantic groups as suggested by [21, 18] for fair comparison. There are nine semantic groups, which are: *colour, texture, shape, part, activity, behaviour, nutrition, and habitat*. For aPY, [18] report that the provided 64 attributes are significantly repeated and redundant. They manually choose 25 of them in their experiments. Such a suggestion also supports the necessity of our idea that using semantic groups. There are five groups: *shape, texture, plant, part, and materials* which are shown in Fig. 5. It is noticeable that the *plant* group is unusual and only possessed by the class that is also named *plant*. In the later experiments, we show such an unusual group can be accurately classified.

Simile Annotations We invite three labellers to give annotations for the two data through the process introduced in Section 3.1. We accumulate their choices of similes to each target class. We then empirically choose k similes of each target classes, where $k = 5$ and 3 for AwA and aPY respectively. We demonstrate two examples of classes annotated by grouped similes in Fig. 5.

4.1. Implicit Attribute Discovery

Fig. 6 shows some examples of our graph-cut results. We demonstrate the simile groups of *shape* and *part* that shared by the two datasets. Two trends can be seen from the results. Firstly, the clustering tends to agree with the animal taxonomy. For example, in the term of *part*, dogs and wolves



Figure 6. Partial results of graph-cut class-clustering. Images with in the same colour of frames are from the same cluster.

are clustered due to our human visual perception is not isolated from knowledge. The semantic meaning can also affect how we perceive the visual information. The second trend is that we can easily tell many implicit attributes from the cluster of images. For instance, it can be seen that the bulls and goats are clustered. We assume that the implicit attribute is ‘with horns’, although their horns have different styles. In contrast, the aPY dataset is far more challenging. The attributes of natural things, e.g. dog, are barely associated with artificial things, e.g. bikes. Therefore the clusters tend to be more isolated from each other. Consequently, the average size of a cluster tends to be larger than that of AwA. For example, in Fig. 6, seven classes are clustered together in the *shape* group of the aPY dataset, whereas for the AwA dataset, the average cluster size is only 3.57. It is also noticeable that, in the *bicycle* example that is shown in Fig. 5, all of the first simile is *motorbike* since this is the only relevant class in the training set. Since the number of classes is small in aPY, such situation does not severely degrade the performance. However, for a large number of unseen classes, we might require the training sources to be more abundant.

4.2. Compared to State-of-the-art methods

Settings Due to the large variations of published settings that are different in terms of adopted visual features, types of semantics, seen/unseen splits, etc., it is impractical to

Table 2. Compared to the state-of-the-arts using deep features.

Methods	Deep Feature	AwA	aPY
DAP [21]	V	57.23	38.16
SJE [3]	A	61.90	-
ESZSL [33]	V	75.32	24.22
SSE [40]	V	76.33	46.23
JLSE [41]	V	79.12	50.35
Ours	V	78.42	56.38

V: VGG; A: AlexNet; - indicate the published result is missing.

compare with every possible setting. Therefore, adopt the most common setting, on which the highest published results are reported. Methods under different settings, *e.g.* transductive settings [31, 13, 19], or aided by various semantic informations [1] are not compared. Specifically, the seen/unseen splits is 40/10 for AwA, and 20/12 for aPY. The adopted visual features are extracted by deep models. Our method and most of state-of-the-art methods adopt the VGG-19 features [35] whereas [3] use AlexNet instead. We summarise our comparison in Table 2.

Discussion Our method can outperform most of the state-of-the-art methods and the overall recognition rate is only 0.7 % lower than that of [41] on AwA. However, our method achieves significant improvement of 6.03% over [41] on the aPY dataset. We ascribe such performance difference to that the variation of unseen classes of the two datasets is different. For instance, as shown in Fig. 5, an *unseen* class of AwA is similar to several *seen* classes, whereas the unseen classes in aPY are often related to only one class. In other words, the boundaries between the implicit attributes in aPY are more discriminative than that of AwA. In contrast, 5 adopts explicit attributes which are noisy and therefore cannot share such a priority.

4.3. Detailed Analysis

4.3.1 Various baseline methods

In order to understand the contribution of each component of our method, we compare to extensive baseline methods and related work using low-level features rather than deep features. For published results, we compare to DAP [21], DSV [18], ZSRwUA [17], ESZSL [33], and DCLA [38]. We also substitute or remove components in our GSE model so as to show their contributions to the overall performance. Our experiments are summarised in Table. 3, using which we can discuss following questions.

Advantages of implicit attributes For the first baseline EA+GSE, we use the same learning framework as our GSE. We only substitute the implicit attributes into conventional explicit attributes. From the comparison between using EA and IA, the performance gains are 4% and 5% on the two datasets, which indicates implicit attributes can adequately fill the visual-semantic gap than explicit attributes. DCLA

Table 3. Compared to baseline methods using low-level features.

Baselines	Attribute	Mapping	AwA	aPY
DAP [21]	A	P	40.5	18.12
DSVA[18]	A+G	E	30.6	19.43
ZSRwUA[17]	A	P	43.0	26.02
ESZSL[33]	A	E	49.3	27.27
DCLA [38]	DA	P	48.3	-
EA + GSE	A+G	E	46.5	25.12
IA + LDA + NN	IA	E	27.4	17.20
IA + Grouping + NN	IA+G	P	44.2	22.82
Ours: IA + GSE	IA+G	E	50.1	30.25

A: Explicit Attributes; G: Attribute Grouping; DA: Data-driven Attributes; IA: Implicit Attributes; P: Prediction based; E: Embedding based.

is data-driven attributes based on visual data that is 8% than DAP, but the performance is 2% lower than ours. More importantly, our implicit has specific semantic meaning, *i.e.* we know which of *seen* classes possess the attributes, whereas DA in DCLA is completely not human-understandable.

Effect of Grouping For the second baseline IA+LDA+NN, we show the effect of using grouped simile. The statistics of all groups are summed up. We then perform graph-cut using non-grouped similes to achieve non-grouped implicit attributes. The model is simply LDA+NN without ensemble. As a result, we observe dramatical performance degradation, 23% on AwA, and 13% on aPY, respectively. The reason is that implicit attributes are only discriminative to class clusters. The classes within the cluster cannot be distinguished, which results in the worst performance.

Visual-semantic mapping approaches Most previous methods adopt the DAP framework that predict each attribute separately. Recent methods are shown improved performance using embedding based framework in [2] that learns all attributes jointly as a whole representation. Our embedding is slightly different from their approach due to the implicit attributes are separated by graph cut. Our purpose is to project the visual data with the same attribute into a compact space rather than multi-label embedding as ESZSL [33]. For the baseline method IA+Grouping+NN, the visual feature is directly mapped to training samples and use the attributes of the nearest neighbour for prediction like IAP[21]. Again, our method significantly outperforms all of the aforementioned baselines.

Efficiency The entire framework is very efficient. Even though the off-line training time is usually not that important, it can determine whether or not the method can be utilised in practical applications. Our work is conducted in Matlab 2014a environment that is installed on a 12-core Linux system with 400G memory. For PCA, it takes 123 seconds and 109 seconds on AwA and aPY datasets, respectively. For LDA, each semantic group requires up to

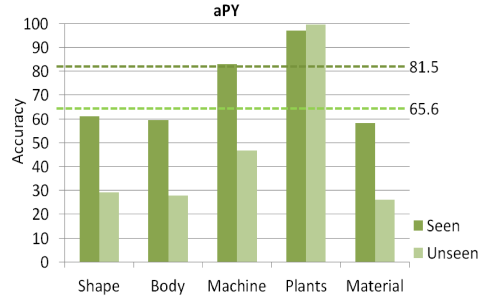
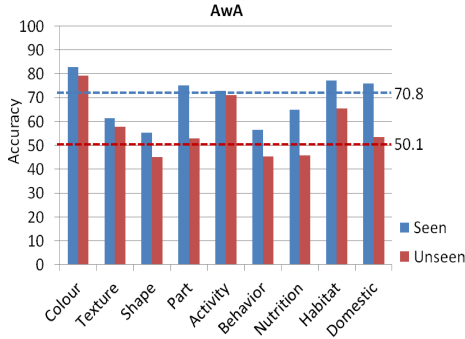


Figure 7. Implicit Attribute Prediction Precision on AwA and aPY. Results are shown by different simile groups.

20 seconds to train each model. Besides these two main training steps, the other procedures are completed within a few seconds. We ascribe the high efficiency to the grouping strategy and the highly compact implicit attributes. Because the learning task is decomposed into grouped subtasks, the computation cost is reduced exponentially.

4.3.2 Implicit attribute prediction

The success of our GSE relies on the premise that the implicit attributes can be reliably predicted. Since our graph-cut algorithm assigns each class to one implicit attribute, during the test, we examine whether the images are mapped to the correct implicit attributes. We test on both *seen* and *unseen* classes to show the performance drop from training to test. From Fig. 7, we can see the average performance drop is roughly 20% on both datasets. However, in aPY, only one class use the highest group *plants*. The remaining training-test performance drop is significantly large, which explain the overall ZSL recognition rate is only 30.25% in Fig 3. Interestingly, the recognition rate on AwA is the same to the average precision of implicit attribute prediction. Such results manifest our embedding mechanism can reliably make ZSL prediction based on given implicit attributes. The attribute-to-label gap is zero in this case. We assume the visual-semantic error is corrected by our ensemble mechanism to some extents.

4.3.3 GSE under different Scenarios

Lastly, we evaluate our GSE under different settings. We mainly concern how is the performance when testing by both *seen* and *unseen* classes. We randomly choose half of the images in each *seen* class for training (denoted by \mathcal{X}_{train}) and the other half for testing (\mathcal{X}_{test}). Firstly, we perform ZSL recognition on the reduced training set. The overall accuracies do not drop down (50.1 to 49.7 and 30.25 to 30.16). The second setting is conventional classification task, \mathcal{X}_{test} is also from *seen* classes. We observe significant improvements over the ZSL results. In the last experiment, the test images are from mixture classes of \mathcal{Y} and \mathcal{Z} .

Table 4. Evaluating GSE on different settings.

Settings	AwA		aPY	
Methods	DAP	Ours	DAP	Ours
$\mathcal{X}_{train} \rightarrow \mathcal{Z}$	50.2	49.7	18.42	30.16
$\mathcal{X}_{train} \rightarrow \mathcal{Y}$	39.8	70.4	49.96	64.32
$\mathcal{X}_{train} \rightarrow \mathcal{Y} + \mathcal{Z}$	12.9	42.5	13.84	24.22

The performance loss is not severe, *i.e.* only 7% and 6% recognition rate drop for the two datasets. Such results indicate our method can withstand the training-bias problem in most existing approaches, such as DAP [21].

5. Conclusion

In this paper, we proposed a unified framework for ZSL including simile annotating, implicit attribute discovery, and the GSE model for ZSL classification. Our method achieved state-of-the-art results on AwA and significantly outperformed existing methods on aPY. We conclude our work as follows. Firstly, similes are effective to describe complex visual appearance. Grouping makes simile more meaningful and discriminative for ZSL tasks. Secondly, our graph-cut algorithm can reliably capture the implicit attributes from similes and do not suffer from the correlation and training bias problems. Thirdly, our ensemble mechanism can find the most relevant simile groups during the test. As a result, the loss of accuracy from attribute prediction to ZSL recognition is small.

For future work, it is necessary to extend our method on large-scale datasets so as to achieve more class exemplars for similes. Another interesting direction for future investigation is the cross-domain ability of the implicit attributes. Since most of the similes are visual-based general terms, we do not need to change the attribute list to adapt to different datasets. One could train rich implicit attribute models on a large-scale dataset that can be generalised widely. In this way, the cost of designing attribute list is significantly mitigated.

References

- [1] Z. Akata, M. Malinowski, M. Fritz, and B. Schiele. Multi-cue zero-shot learning with strong supervision. In *CVPR*, 2016.
- [2] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.
- [3] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of output embeddings for fine-grained image classification. In *CVPR*, 2015.
- [4] Z. Al-Halah, T. Gehrig, and R. Stiefelwagen. Learning semantic attributes via a common latent space. In *VISAPP*, 2014.
- [5] Z. Al-Halah and R. Stiefelwagen. How to transfer? zero-shot object recognition via hierarchical transfer of semantic attributes. In *WACV*, 2015.
- [6] P. N. Belhumeur, J. P. Hespanha, and D. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(7):711–720, 1997.
- [7] Z. Cai, L. Liu, M. Yu, and L. Shao. Latent structure preserving hashing. In *BMVC*, 2015.
- [8] S. Changpinyo, W.-L. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *CVPR*, 2016.
- [9] T. G. Dietterich. Ensemble methods in machine learning. In *Multiple classifier systems*, pages 1–15. Springer, 2000.
- [10] M. Elhoseiny, B. Saleh, and A. Elgammal. Write a classifier: Zero-shot learning using purely textual descriptions. In *CVPR*, 2013.
- [11] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *CVPR*, 2009.
- [12] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007.
- [13] 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.
- [14] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Learning multimodal latent attributes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):303–316, 2014.
- [15] Y. Fu, T. M. Hospedales, T. Xiang, and S. Gong. Transductive multi-view zero-shot learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 37(11):2332–2345, 2015.
- [16] S. Huang, M. Elhoseiny, A. Elgammal, and D. Yang. Learning hypergraph-regularized attribute predictors. In *CVPR*, 2015.
- [17] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *NIPS*, 2014.
- [18] D. Jayaraman, F. Sha, and K. Grauman. Decorrelating semantic visual attributes by resisting the urge to share. In *CVPR*, 2014.
- [19] E. Kodirov, T. Xiang, Z. Fu, and S. Gong. Unsupervised domain adaptation for zero-shot learning. In *ICCV*, 2015.
- [20] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Attribute and simile classifiers for face verification. In *ICCV*, 2009.
- [21] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.
- [22] H. Larochelle, D. Erhan, and Y. Bengio. Zero-data learning of new tasks. In *AAAI*, 2008.
- [23] Y. Long, L. Liu, and L. Shao. Attribute embedding with visual-semantic ambiguity removal for zero-shot learning. In *BMVC*, 2016.
- [24] Y. Long, L. Liu, and L. Shao. Towards fine-grained open zero-shot learning: Inferring unseen visual features from attributes. In *WACV*, 2017.
- [25] Y. Long, F. Zhu, and L. Shao. Recognising occluded multi-view actions using local nearest neighbour embedding. *Computer Vision and Image Understanding*, 144:36–45, 2016.
- [26] T. Mensink, E. Gavves, and C. Snoek. Costa: Co-occurrence statistics for zero-shot classification. In *CVPR*, 2014.
- [27] T. Mensink, J. Verbeek, F. Perronnin, and G. Csurka. Distance-based image classification: Generalizing to new classes at near-zero cost. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 35(11):2624–2637, 2013.
- [28] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *NIPS*, 2009.
- [29] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011.
- [30] J. Qin, Y. Wang, L. Liu, J. Chen, and L. Shao. Beyond semantic attributes: Discrete latent attributes learning for zero-shot recognition. *IEEE Signal Processing Letters*, 23(11):1667–1671, 2016.
- [31] M. Rohrbach, S. Ebert, and B. Schiele. Transfer learning in a transductive setting. In *NIPS*, 2013.
- [32] M. Rohrbach, M. Stark, G. Szarvas, I. Gurevych, and B. Schiele. What helps where—and why? semantic relatedness for knowledge transfer. In *CVPR*, 2010.
- [33] B. Romera-Paredes and P. Torr. An embarrassingly simple approach to zero-shot learning. In *ICML*, 2015.
- [34] L. Shao, L. Liu, and M. Yu. Kernelized multiview projection for robust action recognition. *International Journal of Computer Vision*, 118(2):115–129, 2016.
- [35] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint:1409.1556*, 2014.
- [36] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.
- [37] N. D. Yann, T. Gokhan, H.-T. Dilek, and L. Heck. Zero-shot learning for semantic utterance classification. In *ICLR*, 2014.
- [38] F. Yu, L. Cao, R. Feris, J. Smith, and S.-F. Chang. Designing category-level attributes for discriminative visual recognition. In *CVPR*, 2013.
- [39] M. Yu, L. Liu, and L. Shao. Structure-preserving binary representations for rgb-d action recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 38(8):1651–1664, 2016.
- [40] Z. Zhang and V. Saligrama. Zero-shot learning via semantic similarity embedding. In *ICCV*, 2015.
- [41] Z. Zhang and V. Saligrama. Zero-shot learning via joint latent similarity embedding. In *CVPR*, 2016.