# Coherent Image Annotation by Learning Semantic Distance

Tao Mei †, Yong Wang ‡, Xian-Sheng Hua †, Shaogang Gong ‡, Shipeng Li †
† Microsoft Research Asia
‡ Department of Computer Science, Queen Mary, University of London
{tmei,xshua,spli}@microsoft.com; {ywang,sgg}@dcs.qmul.ac.uk

## Abstract

*Conventional approaches to automatic image annotation usually suffer from two problems: (1) They cannot guarantee a good semantic coherence of the annotated words for each image, as they treat each word independently without considering the inherent semantic coherence among the words; (2) They heavily rely on visual similarity for judging semantic similarity. To address the above issues, we propose a novel approach to image annotation which simultaneously learns a semantic distance by capturing the prior annotation knowledge and propagates the annotation of an image as a whole entity. Specifically, a semantic distance function (SDF) is learned for each semantic cluster to measure the semantic similarity based on relative comparison relations of prior annotations. To annotate a new image, the training images in each cluster are ranked according to their SDF values with respect to this image and their corresponding annotations are then propagated to this image as a whole entity to ensure semantic coherence. We evaluate the innovative SDF-based approach on Corel images compared with Support Vector Machine-based approach. The experiments show that SDF-based approach outperforms in terms of semantic coherence, especially when each training image is associated with multiple words.*

## 1. Introduction

With the prevalence of digital imaging devices such as webcams, phone cameras and digital cameras, image data are now explosively increased. An emerging issue is how to browse and retrieve this daunting volume of images. A possible way is to annotate images and then retrieve these images by their associated words [4]. If all the images are annotated, image retrieval can be solved effectively and efficiently by the well-developed techniques in text retrieval. Automatic image annotation aims to automatically generate words to describe the content of a given image.

Conventional approaches to automatic image annotation can be categorized along the following two dimensions.

(a) peach (b) peach (c) sun

Figure 1. Semantic similarity $\neq$ visual similarity. (a) and (b) have the same semantic but different appearances, while (b) and (c) have similar appearance but different semantics.

- *Learning*-based approaches which formalize image annotation as a learning problem. Usually, a statistical model is learned either for each word [9] [18] or for the joint distribution of words and visual tokens in each image [1] [5] [6].

- *Search*-based approaches which leverage search techniques to find similar images and then directly mine the annotation from the words associated with these images [10] [13] [16].

However, both of these two categories suffer from two problems. (1) It is difficult for most of existing approaches to guarantee a good semantic coherence of the annotated words for an image, as they treat each word independently without considering the inherent semantic coherence of the words. In other words, few of them take the annotation as a coherent semantic entity. For example, *indoor* and *sky* are unlikely to appear together in a real situation. However, if these two words are labeled individually, both of them may be annotated to the same image. In another case, if an *tiger* image is labeled as *cat*, it is better than to be labeled as *garden* although *cat* is not an exact match. (2) They heavily rely on visual similarity for judging semantic similarity. In fact, it is well-known that semantic similarity does not equal to visual similarity [4] (see Figure 1 for an example). As a result, the noises introduced by visual similarity can propagate to learning and search steps, and therefore degrade the

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

overall performance. For instance, if Figure 1(a) is input to search-based approach, then *sun* may be a false alarm.

To address the above problems in these two categories, we propose a novel approach to image annotation which is characterized by simultaneously learning semantic distance on the basis of prior annotation knowledge and propagating the annotation of an image as a whole entity. Specifically, we partition the training set into a number of semantic clusters and learn a semantic distance function (SDF) for each cluster based on the relative comparison relations of prior annotations. The learned SDF is used for measuring semantic similarity between images. To annotate a new image, the training images in each cluster are ranked according to their SDF values with respect to this image, and their corresponding annotations are then propagated among different clusters to this image. The proposed SDF-based approach have the following two distinctive advantages compared with conventional approaches.

(1) The annotation of a training image is always propagated as a whole entity rather than separate words, which enables our approach to generate annotations with semantically coherent words, as well as to be more flexible in the size of word vocabulary.

(2) The SDF is learned based on relative comparison relations (e.g., B is closer to A than C is to A) rather than absolute pairwise distances, which leads to our approach being easily extended to weakly labeled training data. It is well-known that such relative comparison is more consistent with human's perception of similarity. Note that this is the first work on learning distance by directly exploiting the prior semantic knowledge in annotations.

We will next review related work on image annotation in Section 2, and then present the proposed SDF-based approach in Section 3. Section 4 gives the experiments and evaluations, followed by the conclusion in Section 5.

## 2. Related Work

We provide in this section a review on conventional approaches to image annotation including learning-based and search-based approaches.

### 2.1. Learning-Based Annotation

In learning-based approaches, image annotation is posed as a learning problem which learned a statistical model either for each word or for the joint distribution of words and visual tokens in each image. One approach treats image annotation as an image classification problem [9] [18]. Specifically, each word is viewed as a unique class. A binary classifier for each class or a multi-class classifier is trained independently to predict the annotations of new images. Various learning algorithms have been adopted for this purpose, such as Support Vector Machines (SVM) [18], Hidden Markov Models (HMM) [9], and so on. Another approach represents the words and visual tokens in each image as features in different modalities. Image annotation is then formalized by modeling the joint distribution of visual and textual features on the training data and predicting the missing textual features for a new image. The works for modeling this joint distribution include translation language model [1], cross-media relevance model (CMRM) [6], multiple Bernoulli relevance model (MBRM) [5], and so on.

### 2.2. Searching-Based Annotation

One of the most well-known approaches in this category is AnnoSearch system [16]. This system employs a two-step process of searching semantically similar images followed by mining annotations from them. Specifically, given a query image and initial keywords, the search process is to discover visually and semantically similar images on the Web, while the mining process is designed to discover salient words from textual descriptions of the search results. Li *et al.* relaxed the query of both image and initial words to image only [10]. The idea of searching-based annotation was further applied to Web image annotation by mining the surrounding text in the same web page [13].

## 3. Approach

Given a set of training images with annotations, the visual feature of the $i$-th training image is represented as a vector $\mathbf{x}_i$. Thus, the whole training set with $n$ images is denoted as $\mathcal{T} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$. The associated annotations are represented as $\{\mathbf{t}_1, \mathbf{t}_2, \ldots, \mathbf{t}_n\}$, where $\mathbf{t}_i$ is the annotation of $i$-th image. $\mathbf{t}_i(j) = 1$ if $j$-th word in the vocabulary is annotated to $i$-th image; otherwise, $\mathbf{t}_i(j) = 0$. A test image is represented as $\mathbf{x}$, and its soft annotation is represented as $\mathbf{w}$, where $\mathbf{w}(j) \in [0, 1]$ indicates the probability associating word $j$ to $\mathbf{x}$. Figure 2 illustrates the proposed SDF-based approach to image annotation consisting of the following components.

1) **Semantic clustering of training images**. In this step, the training images are clustered according to the semantics indicated by their ground truth annotations. The clustering is implemented through a pairwise clustering algorithm, where the pairwise similarity between two images is measured by the word similarity via WordNet. As a result, the semantic space is partitioned into several subspaces or clusters where the images in each cluster are semantically similar enough.

2) **SDF learning**. Given the partition of the semantic space, a SDF is learned for each semantic cluster. SDF

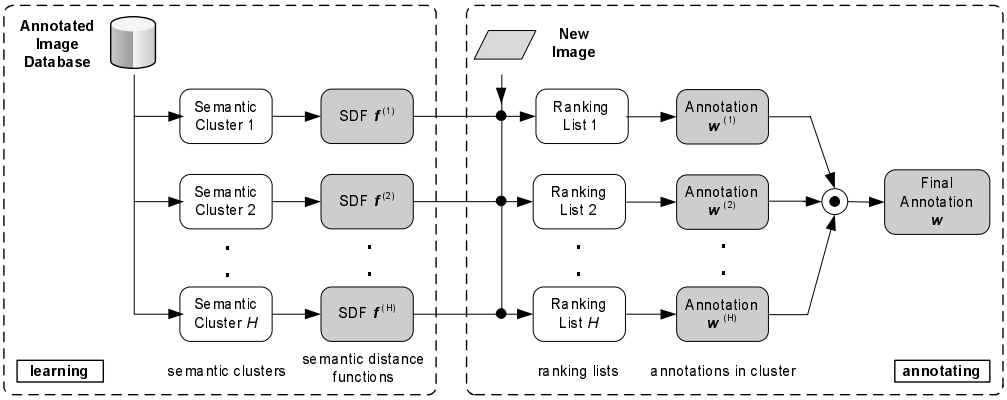



Figure 2. Proposed approach to image annotation by semantic distance learning.

is a function mapping the visual features of two images to their semantic distance. The learning of SDF is based on the relative comparison relations rather than the absolute pairwise distances.

3) **SDF-based image ranking**. Given a new image, the probability of this image associated with each semantic cluster is estimated. For each cluster, the semantic distance between the new image and each training image in this cluster is computed by the learned SDF function. We then rank the images in this cluster according to their SDF distances and propagate their annotations to the new image, which yields a probabilistic annotation for this new image from each cluster and the probability associating this image to each cluster.

4) **Annotations propagation**. The annotations obtained among all semantic clusters are then probabilistically propagated to the new image. We adopt a linear weighted fusion of the annotation from each cluster.

### 3.1. Semantic Clustering of Images

The first step is semantic clustering of the training images, i.e., partition of semantic space. We are aware that images with different semantics have different measures of semantic similarity—they focus on different aspects of visual properties. This phenomenon can be explained by Figure 3, in which two words (*motorbike* and *sky*) are given as examples. It is obvious that for *motorbike*, shape is more informative than color and texture, while for *sky*, color and texture are more informative than shape. Therefore, it is not applicable to learn a single semantic similarity for all the images. Instead, we partition the semantic space into a number of subspaces by semantically clustering the training images. A semantic distance function (SDF) is learned for each cluster with the assumption that images with similar semantic share the same SDF.

However, semantic clustering of images is very different from traditional image clustering since we do not have

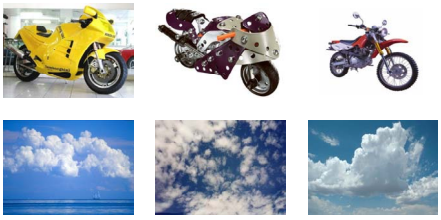


Figure 3. Images with different semantics have different measures of semantic similarity. It is obvious that for *motorbike* (first row), shape is more informative than color and texture, while color and texture are more informative than shape for *sky* (second row).

vectorized features to represent image semantics. We assume that given the manual annotation of images, the semantics of images can be represented by the annotations rather than low-level visual features. This is because the textual words are at a much higher language level than visual features. Given this kind of features, i.e., each sample is a small set of words, we can partition the data using the pairwise proximity clustering method. Partitioning proximity data is considered to be a more difficult problem than partitioning vectorized data. The proximity is not a metric because the triangle inequality rule does not hold. Therefore, traditional clustering methods such as $k$-means are not suitable. Instead, we have taken a two step approach which finds the embedded vectors of the original pairwise proximity data followed by using $x$-means algorithm [11] to cluster the embedded vectors.

Given two sets of words $\mathbf{a} = \{a_1, a_2, .., a_{n_1}\}$ and $\mathbf{b} = \{b_1, b_2, .., b_{n_2}\}$, where $n_1$ and $n_2$ are the number of words in these two sets, respectively, the semantic distance $SD(\mathbf{a}, \mathbf{b})$ between the two sets is computed by looking for the closest word in one set with respect to a particular word in another.

$$SD(\mathbf{a}, \mathbf{b}) = \frac{1}{2n_1}\sum_{i=1}^{n_1} \min_j JCN(a_i, b_j) + \frac{1}{2n_2}\sum_{j=1}^{n_2} \min_i JCN(a_i, b_j) \quad (1)$$

where $JCN(a_i, b_j)$ represents the semantic distance between two words, given by the JCN word similarity via

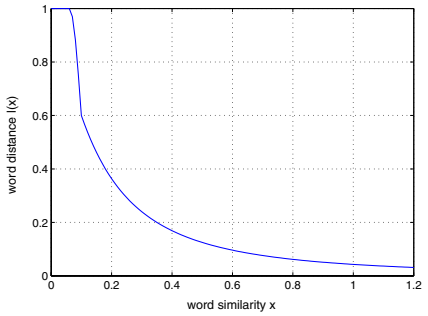



Figure 4. The function $\ell(x)$ maps word similarity to word distance.

WordNet. The original JCN similarity measure ranges from 0 to $\infty$. It can be further transformed into a distance measure ranging from 0 to 1 by the following function [8].

$$\ell(x) = \begin{cases} 1 & x \leq 0.06 \\ 0.6 - 0.4\sin(\frac{25\pi}{2}x + \frac{3}{4}\pi) & 0.06 \leq x \leq 0.1 \\ 0.6 - 0.6\sin\left(\frac{\pi}{2}(1 - \frac{1}{3.471x+0.653})\right) & x \geq 0.1 \end{cases} \quad (2)$$

where $x$ denotes the JCN similarity, and $\ell(x)$ is motivated by an empirical investigation of the JCN similarity measures between different pairs of words, shown in Figure 4. Specifically, words with a JCN similarity less than 0.06 are found to be rarely related, e.g., apple/bath (0.051) and earth/lighthouse (0.059). Thus, $f(x)$ is set to the largest distance 1.0 when $x \leq 0.06$. When $x = 0.1$, $f(x)$ is intentionally set to 0.6, e.g., elephant/lion (0.092) and glacier/rock (0.099). The third segment of $f(x)$ is obtained by fitting a sin function considering the continuousness.

Given the annotations of the training images, we compute the semantic distance between each pair of training images according to Eq. (1). Since the pairwise semantic distances violate the metric, it is impossible to find a natural embedding of the original data into a vector space. However, a natural embedding is not really necessary if the purpose is only data clustering instead of preserving the absolute distances. Roth *et al.* [12] found that any clustering method such as $k$-means, which is invariant under additive shifts of the pairwise proximities, can be reformulated as a grouping problem in an Euclidean space. Based on this observation, they proposed a *constant shift embedding* framework for non-metric pairwise proximity, which can completely preserve the cluster structure. Embedding the pairwise data into an Euclidean space is helpful for data visualization, as well as there are many existing clustering algorithms in such space. We used $x$-means [12] which is a variant of $k$-means to cluster the embedded vectors and automatically chose the optimal number of clusters.

### 3.2. Learning Semantic Distance Functions

The second step is to learn a SDF for each semantic cluster which maps visual features to a semantic distance. In general, image annotation depends on a good distance function for measuring the semantic or perceptual similarity between images. Usually, the visual similarity in the Euclidean space is taken as semantic similarity [4], which does not hold in many real cases as we have mentioned in Figure 1. In a more elaborated approach, geodesic distances are used for image annotation by finding a non-linear manifold in feature space. However, learning a manifold for general image still remains an open problem. Research on semantic similarity is further enriched by learning some distance metrics [17] [19]. These works have focused on statistical analysis of feature distribution and distance functions [19] or contextual information [17], given a set of prior distances. However, they do not directly deal with semantics. In other words, the learned distances still heavily rely on visual features. To relax these strict assumptions, we propose a data-driven approach to learning the semantic similarity directly from the annotations of training images.

The SDF maps the visual features to semantic distance between two images. Since this procedure is identical in each cluster, we describe it in a general case. Given the training images $\mathcal{T}$ and their pairwise distances $SD(\mathbf{x}_i, \mathbf{x}_j)$, we need to learn a SDF $f(\mathbf{x}, \mathbf{y})$ which is consistent with the ground truth pairwise distances ($\mathbf{x}$ and $\mathbf{y}$ denotes a training or test sample). An intuitive approach to learning such a SDF is using the following least square regression

$$f = \arg\min_q \sum_{i=1}^{n}\sum_{j=1}^{n}\left(SD(\mathbf{x}_i, \mathbf{x}_j) - q(\mathbf{x}_i, \mathbf{x}_j)\right)^2 \quad (3)$$

A disadvantage of this setting is that we are solving a problem with very hard constraints. With limited training data and the high dimensional visual features, learning such a SDF tends to be over-fitting. Thus, we relax the pairwise distance constraints to the relative comparison constraints in the following form

$$\{\mathbf{x}_j \text{ is closer to } \mathbf{x}_i \text{ than } \mathbf{x}_k \text{ is to } \mathbf{x}_i\} \quad (4)$$

The idea of keeping the rank order instead of the absolute distances has been presented in some techniques for Multidimensional Scaling for non-metric data [3]. Especially in image annotation, we are more interested in the rank order of images rather than their absolute pairwise distances. Moreover, learning by relaxed constraints allows us to incorporate other data labeled by relative comparison relations since it is much easier for human to give such relative constraints (e.g., the distance between A and B is smaller than that between A and C) than quantitative constraints (e.g., the distance between A and B is 0.05 while the distance between A and C is 0.08).

#### 3.2.1 Learning by Relative Comparison

The learning of SDF based on relative comparison is derived from the work in [14]. Given the training images $\mathcal{T}$

and a set of relative comparison constraints $\mathcal{S}$

$$\mathcal{S} = \{(x_a, x_b, x_c);\ x_b \text{ is closer to } x_a \text{ than } x_c \text{ is to } x_a\} \quad (5)$$

A distance metric $f(\mathbf{x}, \mathbf{y})$ between vector $\mathbf{x}$ and $\mathbf{y}$ is parameterized by two matrices $A$ and $W$

$$f(\mathbf{x}, \mathbf{y}) = \sqrt{(\mathbf{x}-\mathbf{y})^T AWA^T(\mathbf{x}-\mathbf{y})} \quad (6)$$

where $W$ is a diagonal matrix with non-negative entries and $A$ is a real matrix transforming the original data. $f(\mathbf{x}, \mathbf{y})$ is equivalent to the weighted Euclidean distance on the linear transformed data points $A^T\mathbf{x}$ and $A^T\mathbf{y}$. Especially, if $AWA^T$ is an identity matrix, $f(\mathbf{x}, \mathbf{y})$ will be equal to the Euclidean distance.

Since linear transformation has its limitation in function complexity, we employ the kernel trick to obtain a nonlinear transformation. Suppose we have a mapping function $\phi(\mathbf{x})$ which maps $\mathbf{x}$ to a very high dimensional vector and define the transformation matrix $A = [\phi(\mathbf{x_1})\ \phi(\mathbf{x_2})\ \ldots\ \phi(\mathbf{x_n})]$. Then, $f(\mathbf{x}, \mathbf{y})$ can be kernelized as

$$f(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{i=1}^{n} W_{ii}\Big(K(\mathbf{x}, \mathbf{x}_i) - K(\mathbf{y}, \mathbf{x}_i)\Big)^2} \quad (7)$$

where the use of kernel $K(\mathbf{x}, \mathbf{y}) = \phi(\mathbf{x})\phi(\mathbf{y})$ suggests a particular choice of $A$, the parameters to be learned are the diagonal elements of $W$. Given this parameterized distance function and the relative comparison constraints, the learning problem is summarized as

$$\begin{aligned} solve \quad & W_{ii} \\ s.t. \quad & \forall(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) \in \mathcal{S}, f(\mathbf{x}_a, \mathbf{x}_c) - f(\mathbf{x}_a, \mathbf{x}_b) > 0 \\ & W_{ii} \geq 0 \end{aligned} \quad (8)$$

Similar to the optimization problem in SVM [2], we transform the hard constraints in Eq. (8) into soft ones by adding slack variables to each relative comparison constraints. For the convenience of computation, we also convert the constraints on $f$ to $f^2$. This leads to

$$\begin{aligned} min \quad & \textstyle\sum_\ell \xi_\ell \\ s.t. \quad & \forall(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) \in \mathcal{S}, f^2(\mathbf{x}_a, \mathbf{x}_c) - f^2(\mathbf{x}_a, \mathbf{x}_b) \geq 1 - \xi_\ell \\ & W_{ii} \geq 0 \\ & \xi_\ell \geq 0 \end{aligned} \quad (9)$$

where $\ell$ is the index of the relative comparison constraints.

If the set of relative comparison constraints are feasible and there exists one $W$ fulfilling all the constraints, there will be infinite number of solutions for $W$ since a scalar transformation on a feasible solution $W$ is always a feasible solution. To make the solution unique, we add an additional constraint that the learned distance function $f$ is close to the un-weighted Euclidean distances as much as possible. This constraint can be reformulated as minimizing the norm of the eigenvalues of $AWA^T$, which is equal to $\|AWA^T\|_F^2$. Thus we rewrite the optimization problem, i.e., the first row in Eq. (9), with this additional constraint as

$$min \quad \tfrac{1}{2}\|AWA^T\|_F^2 + C\textstyle\sum_\ell \xi_\ell \quad (10)$$

where $\frac{1}{2}$ is added for the convenience of formulating it as a standard quadratic programming problem, and $C > 0$ is a balance parameter. The bigger $C$ is, the more constraints are to be satisfied; the smaller $C$ is, the closer $f$ is to the unweighted distance function. Through some mathematical derivation, the above optimization problem in Eq. (9) and (10) can be further reformulated as the following standard quadratic programming problem

$$\begin{aligned} min \quad & \tfrac{1}{2}\omega^T L\omega + C\textstyle\sum_\ell \xi_\ell \\ s.t. \quad & \forall(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c) \in \mathcal{S}, \omega^T\Big(g(\mathbf{x}_a, \mathbf{x}_c) - g(\mathbf{x}_a, \mathbf{x}_b)\Big) \geq 1 - \xi_\ell \\ & \omega_i \geq 0 \\ & \xi_\ell \geq 0 \end{aligned} \quad (11)$$

where

$$\begin{aligned} L &= (A^TA) * (A^TA) \\ g(\mathbf{x}, \mathbf{y}) &= (A^T\mathbf{x} - A^T\mathbf{y}) * (A^T\mathbf{x} - A^T\mathbf{y}) \end{aligned} \quad (12)$$

$\omega$ is the diagonal elements of $W$ and $*$ denotes the elementwise product between vectors. For the kernel version, $\mathbf{x}$ is replaced by $\phi(\mathbf{x})$ and both $L$ and function $g(\mathbf{x}, \mathbf{y})$ can be written in the function of the kernel function $K(\ \cdot\ )$.

### 3.2.2 Learning a SDF for a Semantic Cluster

Given the learning algorithm in the above section, this section discuss how to learn a SDF for a specific semantic cluster. Let $\mathcal{T}^{(k)} = \{x_i^{(k)}\}_{i=1}^{n_k}$ denote $k$-th semantic cluster, where $x_i^{(k)}$ is $i$-th image and $n_k$ is the number of images in $k$-th cluster, $\mathcal{P} = \mathcal{T}^{(k)}$ denote the set of training samples in this cluster, and $\mathcal{N} = \cup_{i \neq k}\mathcal{T}^{(i)}$ the set of training samples not in this cluster, the major issue in learning a SDF for this cluster is to generate the relative comparisons constraints from the ground truth semantic distances.

A basic assumption of our approach is that images with similar semantics share the same SDF. In other words, $f(\mathbf{x}, \mathbf{y})$ is only valid if $\mathbf{x} \in \mathcal{P}$ or $\mathbf{y} \in \mathcal{P}$. Thus our relative comparison constraints for a particular cluster are only a subset of all the triples. This subset can be represented as

$$\mathcal{R} = \{(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)\} \quad (13)$$

where $\mathbf{x}_a \in \mathcal{P}$ and $(\mathbf{x}_a, \mathbf{x}_b, \mathbf{x}_c)$ satisfy either of the following two conditions

$$\begin{aligned} &(a) \quad SD(\mathbf{x}_a, \mathbf{x}_c) > SD(\mathbf{x}_a, \mathbf{x}_b) \\ &(b) \quad SD(\mathbf{x}_a, \mathbf{x}_c) = SD(\mathbf{x}_a, \mathbf{x}_b), \quad \text{but } \|\mathbf{x}_a - \mathbf{x}_c\| > \|\mathbf{x}_a - \mathbf{x}_b\| \end{aligned} \quad (14)$$

Condition (b) indicates that if two pairs of images have the same semantic distances, then the difference in the feature space is taken into account. Even by this selection policy, the number of constraints will be overwhelming which

makes the optimization problem in Eq. (11) complex. Following the divide-and-conquer philosophy, we randomly sample $m$ subsets of the relative comparison constraints from $\mathcal{R}$, each represented as $\mathcal{R}_i$ $(i = 1, \ldots, m)$. We train $m$ SDFs, denoted as $\{f_1, f_2, \ldots, f_m\}$ in which each $f_i$ is trained by constraints $\mathcal{R}_i$. The final SDF $f$ is an average of these sub-SDF's, i.e., $f = \frac{1}{m}\sum_i f_i$.

### 3.3. Annotating a New Image

Based on the learned SDF , a new image is annotated by finding a ranking list of images from the training set according to their semantic distances to this image. However, there still remain two concerns. (a) Each semantic cluster in the training set has its own SDF, which indicates that the SDF of a given semantic cluster is only valid if it is used to measure the distance between a test image to a training image in this semantic cluster. (b) The SDFs obtained on different clusters are not comparable. They are learned separately with different objective functions and constraints. Thus, we propose a two step approach to annotating a new image. In the first step, for each semantic cluster, we rank the images in this cluster according to their distances to the test image based on the learned SDF for this cluster, and assign a probability that the test image belongs to this cluster. In the second step, we propagate the annotations of the images in each semantic cluster to the test images.

#### 3.3.1 Cluster Association Probability

To propagate the annotation from the $k$-th semantic cluster to a test image, we take the distance of a sample to all the samples in $\mathcal{P}$ as a feature vector and train a logistic regression classifier on the positive and negative set $\mathcal{P}$ and $\mathcal{N}$. The detailed process is described as follows.

1) For each sample in $\mathcal{P}$, compute its semantic distances to all the positive samples in $\mathcal{P}$. This leads to a distance matrix $V^p \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{P}|}$, where $V^p(i,j) = f(\mathbf{x}_i, \mathbf{x}_j)$, $\mathbf{x}_i, \mathbf{x}_j \in \mathcal{P}$ and $|\mathcal{P}|$ is the number of positive samples.

2) For each sample in $\mathcal{N}$, compute its semantic distances to all the samples in $\mathcal{P}$, which results in a distance matrix $V^n \in \mathbb{R}^{|\mathcal{P}|\times|\mathcal{N}|}$.

3) Take the column vectors in $V^p$ and $V^n$ as positive and negative training samples, respectively, and then train a logistic regression classifier.

4) Take $[f(\mathbf{x}, \mathbf{x_1}), f(\mathbf{x}, \mathbf{x_2}), \ldots, f(\mathbf{x}, \mathbf{x}_{|\mathcal{P}|})]^T$ as the feature vector of a test sample and assign a cluster association probability using the trained logistic regression.

#### 3.3.2 Propagation of Annotations

Suppose the corresponding manual annotations of the ranking list from the $k$-th cluster are $\{\mathbf{t}_1^{(k)}, \mathbf{t}_2^{(k)}, \ldots, \mathbf{t}_{n_k}^{(k)}\}$, and their distances to the test images are $\{d_1^{(k)}, d_2^{(k)}, \ldots, d_{n_k}^{(k)}\}$, we intentionally propagate $\mathbf{t}_1^{(k)}$ with weight $1.0$ and the annotation of $\mathbf{t}_5^{(k)}$ with weight $0.5$, so that the annotation propagated from the $k$-th cluster $w^{(k)}$ is

$$\mathbf{w}^{(k)} = \frac{1}{n_k}\sum_i \frac{d_1^{(k)} - \alpha^{(k)}}{d_i^{(k)} - \alpha^{(k)}} * \mathbf{t}_i^{(k)} \tag{15}$$

where $\alpha^{(k)}$ is set so that $\frac{d_1^{(k)}-\alpha^{(k)}}{d_5^{(k)}-\alpha^{(k)}} = 0.5$. $\mathbf{w}^{(k)}$ is normalized so that its $L_1$ norm is 1. The final annotation is

$$\mathbf{w} = \sum_k^H p(k) * \mathbf{w}^{(k)} \tag{16}$$

where $p(k)$ is the association probability of the test image to the $k$-th semantic cluster obtained in Section 3.3.1.

## 4. Experiments

### 4.1. Dataset

We evaluate the proposed approach over two data sets. The first set contains 3100 Corel images, which is a subset of the 5000 images used in [1] [6]. The second set includes 2360 Corel images from Li [7]. As a result, we have 5460 Corel images in total. All the images have been manually annotated with $1 \sim 5$ words in Corel database. The number of unique annotation words in these two sets is 393.

### 4.2. Performance Measures for Image Annotation

In the previous work, there are mainly two performance measures used to compare different algorithms, namely, *precision* and *recall*. They are defined respectively wrt. a particular word $w$ as follows,

$$\begin{aligned} precision(w) &= \frac{\text{\# of images correctly annotated with} w}{\text{\# of images automatically annotated with } w} \\ recall(w) &= \frac{\text{\# of images correctly annotated with } w}{\text{\# of images manually annotated with } w} \end{aligned}$$

A disadvantage of these two measures is that they consider an annotation word as correct only if there is an exact match. However, even if an automatically annotated word does not have an exact match to the ground truth annotation, it is acceptable in some cases. An example is given in Table 1, where the first row is the manual annotation, the second and third rows are the automatic annotations generated by two different algorithms. If we evaluate the performances of these two algorithms in terms of $precision$ and $recall$, Algorithm 1 outperforms Algorithm 2 because there is at least one word (*mountain*) correctly annotated while none

of the annotation words generated by Algorithm 2 has an exact match with the ground truth. In fact, Algorithm 2 performs better than Algorithm 1 because three of the four predicted words have similar semantics with the ground truth (i.e., *tree/trunk, water/waterfall, sun/sunrise*).

Table 1. Comparison of two automatic annotations

| Manual | trunk, waterfall, mountain, sunrise |
|---|---|
| Algorithm 1 | mountain, clouds, street, garden |
| Algorithm 2 | tree, water, counds, sun |

We hereby propose a new performance measure, named semantic Relevance Comparative Score ($RCC$), which takes the semantic relevance between annotated words into account. The intuition of $RCC$ is that if a predicted word does not have an exact match, it is expected to represent the semantics of the image as close as possible. We believe that $RCC$ is more reasonable than the commonly used $precision$ and $recall$ for measure the annotation quality. For example, if a *waterfall* image is labeled as *water*, it is better than to be labeled as *clouds* if *clouds* are not present in this image. $RCC$ is computed from two annotation results on the same dataset. Suppose the number of test images is $n$, the annotations of the ground truth, annotation generated by Algorithm 1, and annotation generated by Algorithm 2 are $\mathbf{T}^0 = \{\mathbf{t}_1^0, \ldots, \mathbf{t}_n^0\}$, $\mathbf{T}^1 = \{\mathbf{t}_1^1, \ldots, \mathbf{t}_n^1\}$, and $\mathbf{T}^2 = \{\mathbf{t}_1^2, \ldots, \mathbf{t}_n^2\}$, respectively. The $RCC$ between Algorithm 1 and 2 is computed by

$$RCC = \frac{\#\text{ of images where } SD(\mathbf{t}_i^0, \mathbf{t}_i^1) < SD(\mathbf{t}_i^0, \mathbf{t}_i^2)}{n} \quad (17)$$

where $SD$ is the distance function defined in Eq. (1). If $RCC > 0.5$, Algorithm 1 performs better in terms of semantic relevance and vice versa.

### 4.3. Evaluation

We used 225 dimensional block-wise color moment as the global features which have proved to be effective for image annotation [15]. Since our approach is based on the semantic distance learning algorithm on the global visual feature, it is not suitable to compare its performance with other approach using other form of features such as blob features. Instead, we have focused on comparing the proposed SDF-based approach with SVM-based approach using the same features, as it is the most widely accepted approach to image annotation in the first research dimension described in Section 2.1. Note that we did not compare our approach with the second dimension as it has utilized a large-scale of Web images, which makes the comparison not fair.

The SVM-based approach takes each word as a unique class. A binary classifier is trained for each class. Given a test image, these classifiers are applied one by one to give a probability of each annotation word. The words with the five largest probabilities are taken as the final annotation.

Table 2. Results of experiment I and II

| Experiment | Method | *Precision* | *Recall* | *RCC* |
|---|---|---|---|---|
| I | SVM | 0.38 | 0.53 | 0.37 |
| | SDF | 0.36 | 0.51 | 0.63 |
| II | SVM | 0.32 | 0.46 | 0.24 |
| | SDF | 0.37 | 0.53 | 0.76 |

In our implementation, the parameter $C$ in Eq. (11) is adjusted automatically, i.e., for a given set of relative comparison constraints, we randomly sample $80\%$ of the constraints and train a SDF. The trained SDF is then tested on the rest $20\%$ constraints. After finding the best $C$, we train the SDF again using all of the constraints. The RBF kernel is adopted as the kernel function in Eq. (11). It is worth noticing that the partition of training and validation is on the pairwise relative comparison instead of the image set itself. We compare the performance of these two approaches in terms of three measures: $precision$, $recall$ and $RCC$. The $precision$ and $recall$ are the average precision and recall of the top 50 words for each approach. This is a widely used comparing methodology in image annotation [1] [6].

**Experiment I:** The whole data set is partitioned into training and testing set. The ratio of the size of training to testing set is 9:1. This is consistent to the setting in [1] [6]. The annotation results show that the SVM-based approach performs a little better in terms of $precision$ and $recall$. However, SVM does not show any advantage in terms of $RCC$, as listed in Table 2.

**Experiment II:** We select the images with more than three annotation words and partition this subset of images into a training and testing set by the ratio of 9:1, which results in 3192 training and 355 testing images. This experiment has shown SDF-based approach performs better on the images with multiple annotation words, as we have taken the semantic relevance between annotated words into account. If an image has a single annotation word, the advantage of cross-word semantic relevance can not be demonstrated. The results are shown in Table 2, from which we can observe that when the images have multiple words, SDF-based approach outperforms SVM in terms of all the three measures. The $RCC$ is improved from $0.63$ to $0.76$. The $precision$ and $recall$ of SVM-based approach has dropped down in this setting. It is because with multiple words, the class boundary between different semantic concept becomes more ambiguous.

Some test images with the annotations generated by the two approaches are shown in Figure 5. Taking the first image (row 1, col 1) as an example, although *horse* is relatively easy to be classified, *foal* and *mare* are not so that SVM-based approach failed to annotate these two words. In contrast, SDF-based approach annotates *foal* and *mare* because of the contextual prior knowledge from manual annotations. For the image of African woman (row 2, col 2),

**GT:** field, foals, horses, mare

**SVM:** grass, horse, animal, tree, plant

**SDF:** horse, foal, mare, fence, tree

**GT:** cliff, sky, water

**SVM:** people, ocean, flower, beach, snow

**SDF:** sky, mountain, water, valley, coast

**GT:** beach, buildings, sand, water

**SVM**: landscape, mountain, snow, people, ocean

**SDF**: sand, beach, water, people, sky

**GT:** field, foals, horses, mare

**SVM**: grass, horse, animal, tree, plant

**SDF**: horse, foal, mare, fence, tree

**GT:** indian, pots, woman

**SVM:** rock, animal, buildings, snow, winter

**SDF:** people, indian, hats, costume, african

**GT:** cow, elk, forest

**SVM:** buildings, waterfall, landscape, ocean, plant

**SDF:** field, grass, cow, deer, tree

**GT:** boats, harbor, sky, water

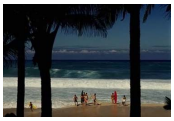
**SVM:** sky, cloth, fashion, decoration, snow

**SDF:** water, sea, boat, harbor, sky

**GT:** ceremony, church, garden, people

**SVM:** buildings, people, fish, snow, rock

**SDF:** people, church, mosque, indian, buildings



**GT:** beach, oahu, people, water

**SVM:** snow, mountain, animal, rock, tree

**SDF:** beach, bay, sea, tree, sunset

Figure 5. Sample images and annotations. GT: ground truth.

SDF-based approach annotates it with *people*. The ground truth annotation does not contain *people* but *woman*. Since *people* and *woman* have very strong semantic relevance we do not think *people* is a junk annotation.

## 5. Discussion

In this paper, we have proposed a novel approach to image annotation based on semantic distance learning. Different from conventional approaches to image annotation which are lack of a good semantic coherence of annotated words and good semantic distances, the proposed approach is able to simultaneously learn a semantic distance by capturing the prior annotation knowledge and propagate the annotation of an image as a whole entity. To the best of our knowledge, this is the first work on learning image distance by directly exploiting the prior knowledge in annotations. The major bottleneck of the proposed approach is that current semantic distance learning cannot handle with the overwhelming number of relative comparison constraints very efficiently, as well as the heavy reliance of WordNet for word similarity. Therefore, our future work will focus on designing more scalable algorithm from ranking data and experimenting on a large-scale of real-world image data.

## References


[1] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003. 1, 2, 6, 7

[2] C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. 5

[3] T. Cox and M. Cox. *Multidimensional Scaling*. Chapman & Hall, London, 1994. 4

[4] R. Datta, D. Joshi, J. Li, and J. Z. Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys*, 40(65), 2008. 1, 4

[5] S. L. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of CVPR*, pages 1002–1009, June 2004. 1, 2

[6] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of SIGIR*, pages 119–126, 2003. 1, 2, 6, 7

[7] J. Li. http://www.stat.psu.edu/~jiali/index.download.html. 6

[8] J. Li. A mutual semantic endorsement approach to image retrieval and context provision. In *Proceedings of Multimedia Information Retrieval*, pages 173–182, 2005. 4

[9] J. Li and J. Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1075–1088, 2003. 1, 2

[10] X. Li, L. Chen, L. Zhang, F. Lin, and W.-Y. Ma. Image annotation by large-scale content-based image retrieval. In *Proceedings of ACM Multimedia*, pages 607–610, 2006. 1, 2

[11] D. Pelleg and A. Moore. $X$-means: Extending $K$-means with efficient estimation of the number of clusters. In *Proceedings of ICML*, pages 727–734, 2000. 3

[12] V. Roth, J. Laub, M. Kawanabe, and J. M. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(12):1540–1551, 2003. 4

[13] X. Rui, N. Yu, T. Wang, and M. Li. A search-based web image annotation method. In *Proceedings of ICME*, 2007. 1, 2

[14] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems*, 2004. 4

[15] M. Wang, X.-S. Hua, Y. Song, X. Yuan, S. Li, and H.-J. Zhang. Automatic video annotation by semisupervised learning with kernel density estimation. In *Proceedings of ACM Multimedia*, pages 967–976, 2006. 7

[16] X.-J. Wang, L. Zhang, F. Jing, and W.-Y. Ma. Annosearch: Image auto-annotation by search. In *Proceedings of CVPR*, pages 1483–1490, 2006. 1, 2

[17] G. Wu, E. Y. Chang, and N. Panda. Formulating context-dependent similarity functions. In *Proceedings of ACM Multimedia*, pages 725–734, 2005. 4

[18] C. Yang, M. Dong, and J. Hua. Region-based image annotation using asymmetrical support vector machine-based multiple-instance learning. In *Proceedings of CVPR*, pages 2057–2063, 2006. 1, 2

[19] J. Yu, J. Amores, N. Sebe, and Q. Tian. Toward robust distance metric analysis for similarity estimation. In *Proceedings of CVPR*, pages 316–322, 2006. 4