

Zero-shot Image Tagging by Hierarchical Semantic Embedding

Xirong Li, Shuai Liao, Weiyu Lan, Xiaoyong Du, Gang Yang^{*}

Key Lab of Data Engineering and Knowledge Engineering, Renmin University of China, 100872, China
Multimedia Computing Lab, School of Information, Renmin University of China, 100872, China
{xirong,leoshine,weiyu,duyong,yanggang}@ruc.edu.cn

ABSTRACT

Given the difficulty of acquiring labeled examples for many fine-grained visual classes, there is an increasing interest in zero-shot image tagging, aiming to tag images with novel labels that have no training examples present. Using a semantic space trained by a neural language model, the current state-of-the-art embeds both images and labels into the space, wherein cross-media similarity is computed. However, for labels of relatively low occurrence, its similarity to images and other labels can be unreliable. This paper proposes **Hierarchical Semantic Embedding** (HierSE), a simple model that exploits the WordNet hierarchy to improve label embedding and consequently image embedding. Moreover, we identify two good tricks, namely training the neural language model using Flickr tags instead of web documents, and using partial match instead of full match for vectorizing a WordNet node. All this lets us outperform the state-of-the-art. On a test set of over 1,500 visual object classes and 1.3 million images, the proposed model beats the current best results (18.3% versus 9.4% in hit@1).

Categories and Subject Descriptors

I.4.8 [Image Processing and Computer Vision]: Scene Analysis—*Object recognition*

Keywords

Image tagging, zero shot learning, semantic embedding

1. INTRODUCTION

People use tags to find specific images in social media. As user-contributed tags tend to be overly personalized and incomplete, automated image tagging that can predict labels relevant with respect to the visual content is crucial for image search. The state-of-the-art approach to image tagging employs a deep convolutional neural network [6], trained on

millions of manually labeled examples. However, such a supervised approach cannot handle novel labels that have no training examples available.

Since acquiring training examples by expert labeling is costly, one might consider to automatically harvest training examples from many user-tagged images online. Indeed, this line of research has produced encouraging results for generic visual classes such as ‘beach’, ‘animal’, and ‘car’ [5, 11]. A prerequisite for their success is that there need to be sufficient, say thousands of, candidate examples for a specific class. However, for many classes, in particular for those fine-grained subclasses, even the candidate examples are scarce, as tagging images with these classes is nontrivial for users without domain-specific knowledge.

To bypass training example acquisition, zero shot learning has been introduced to handle novel labels [1, 3, 7, 9]. Instead of finding a direct mapping between images and target labels, the key idea of zero shot learning is to introduce an intermediate layer between images and labels such that a novel label can also be represented in this layer, even when no example of this label is supplied. In the seminal work by Lampert *et al.* [7] for animal object recognition, this layer is implemented as a set of human-specified high-level attributes, e.g., shape and color, of animals. By exploiting the relative strength of association between attributes and classes, the authors extrapolate attribute-level prediction to novel classes. In a follow-up work [1], the intermediate layer is constructed by embedding image feature vectors and labels into a common Euclidean space such that cross-media relevance can be computed. The attributes are used as side information for label embedding. Although the use of attributes makes the intermediate layer more interpretable, specifying a proper set of attributes suitable for interpreting thousands of visual classes is challenging.

We are interested in zero-shot image tagging that works with no need of manually specifying attributes or other knowledge about the novel labels. In a related work [3], a deep learning based semantic embedding method is proposed. In particular, by training a neural language model [8] on millions of Wikipedia documents, the authors first construct a semantic space where semantically close words are mapped to similar vector representations. Consequently, label embedding is achieved by a table lookup in the model vocabulary. Images are mapped into the same space, by linearly transforming the corresponding visual feature vectors. The transformation is optimized on a set of training labels such that the product similarity between the embedding vectors of an image and its correct label is higher than between the

^{*}Corresponding author.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

SIGIR’15, August 09 - 13, 2015, Santiago, Chile

Copyright is held by the owner/author(s). Publication rights licensed to ACM.

ACM 978-1-4503-3621-5/15/08 ...\$15.00.

DOI: <http://dx.doi.org/10.1145/2766462.2767773>.

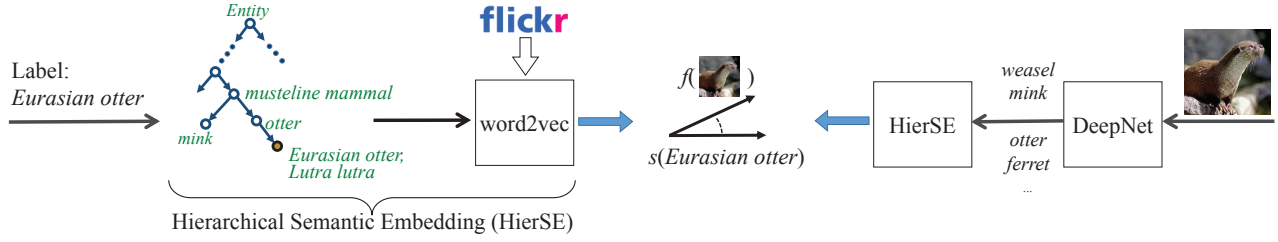


Figure 1: Zero-shot image tagging by hierarchical semantic embedding. Cross-media relevance between an unlabeled image and a test label is computed by cosine similarity between their embedding vectors.

image and other randomly chosen words. More recently, Norouzi *et al.* propose a simpler yet more effective solution for image embedding [9]. Given an unlabeled image, they use an existing m -way image classification system (whose labels have no overlap with target labels) to predict a few most relevant labels, and build the embedding vector of the given image by convex combination of the label embedding vectors. With this solution, they achieve the state-of-the-art results on the ImageNet zero-shot learning task.

We argue that several issues are overlooked in the semantic embedding method [9]. First, since the neural language model essentially exploits word co-occurrence in a text corpus, for a label of relatively low occurrence, its embedding vector could be unreliable for computing its similarity to images and other labels. On the other hand, it is this kind of label that we want to tackle via zero shot learning (otherwise we could choose to harvest training examples from the Internet). Second, how to deal with labels that are out of the language model vocabulary is unclear. Moreover, while Wikipedia articles have been the default source for deducing the semantic space [3, 9], Flickr might be a better source, as its tag co-occurrence statistics better reflect a label’s visual context. By addressing the above issues, this paper improves on the semantic embedding method, and eventually contributes a new model, as illustrated in Fig. 1, that outperforms the state-the-art for zero-shot image tagging.

2. ZERO-SHOT IMAGE TAGGING

2.1 Problem Statement

Given an unlabeled image, the goal of zero-shot image tagging is to automatically tag the image with labels that have no training examples available. This is approached by embedding both the image and the novel labels into a common semantic space such that their relevance can be estimated in terms of the distance between the corresponding vectors in the space. More formally, let x be an image, y be a label, and $p(y|x)$ be a classifier which estimates the relevance of the label y with respect to the image x . Given a set of m_0 training labels \mathcal{Y}_0 , we have access to n training examples $\mathcal{D}_0 = \{(x_i, y_i)\}_{i=1}^n$, where $y_i \in \mathcal{Y}_0$. Consequently, let $p_0(y|x)$ be a m_0 -way classifier learned from \mathcal{D}_0 . We use \mathcal{Y}_1 to denote a set of m_1 test labels, which have no training examples in the zero-shot learning setting, i.e., $\mathcal{Y}_0 \cap \mathcal{Y}_1 = \emptyset$. With the help of \mathcal{D}_0 and some semantic knowledge, we aim to build a classifier $p_1(y|x)$ that can perform reasonably well for \mathcal{Y}_1 .

Our work improves on the semantic embedding model [9], so we first describe it in Section 2.2, and subsequently introduce our model in Section 2.3.

2.2 The Semantic Embedding Model

For the ease of consistent description, we borrow some notation from [9]. The semantic embedding model assumes each label $y \in \mathcal{Y}_0 \cup \mathcal{Y}_1$ is associated with a semantic embedding vector $s(y) \in \mathcal{S}$, where \mathcal{S} is a real coordinate space of q dimensions. The semantic space is formed such that two labels are similar if and only if their corresponding vectors are close. In [9], \mathcal{S} is instantiated by training the skip-gram model [8] using Wikipedia documents. Each $s(y)$ is obtained by matching the label with words in the skip-gram model.

By projecting images into \mathcal{S} , cross-media relevance can be computed. To do so, the model leverages the existing classifier $p_0(y|x)$, and create the semantic embedding vector of x as a convex combination of semantic vectors of the most relevant training labels. In particular, let $y(x, t)$ be the t -th most likely training label for x according to $p_0(y|x)$. Then, the semantic embedding vector of x , denoted by $f(x) \in \mathcal{S}$, is defined as

$$f(x) := \frac{1}{Z} \sum_{t=1}^T p_0(y(x, t)|x) \cdot s(y(x, t)), \quad (1)$$

where T is the maximum number of training labels to be employed, and $Z = \sum_{t=1}^T p_0(y(x, t)|x)$ is a normalization factor. The classifier for the novel label set \mathcal{Y}_1 is defined as

$$p_1(y|x) := \cos(f(x), s(y)), \quad (2)$$

where \cos is the cosine similarity.

Due to potential mismatch between the vocabulary of the skip-gram model and the labels, $s(y)$ could be null. For some successfully matched labels, their relatively low occurrence in the training corpus makes the similarity measurement unreliable. Next, we present a remedy to these issues.

2.3 Hierarchical Semantic Embedding

The proposed method is to construct label embedding and consequently image embedding by exploiting the hierarchical structure defined in the WordNet. We presume that each label has a corresponding node in the WordNet. The WordNet hierarchy allows us to trace from a specific label back to the root, obtaining all its ancestors, denoted as $super(y)$. In contrast to [9] that uses y alone, we utilize both y and $super(y)$. Intuitively, nodes that are more close to y shall contribute more. Putting all this together, we define a hierarchically embedded vector $s_{hi}(y)$ as

$$s_{hi}(y) = \frac{1}{Z_{hi}} \sum_{y' \in \{y\} \cup super(y)} w(y'|y) \cdot s(y'), \quad (3)$$

where $w(y'|y)$ is a weight subject to exponential decay with respect to the minimal path length from y to y' , and Z_{hi} is a

normalization factor given by $Z_{hi} = \sum_{y' \in \{y\} \cup \text{super}(y)} w(y'|y)$. As shown in Eq. (3), the use of $\text{super}(y)$ always allows y to be mapped into \mathcal{S} , while the convex combination makes the similar measure more reliable for rare labels. Moreover, for a label of multiple senses, e.g., ‘mouse’, it will have distinct embedding vectors depending on its given sense. In contrast, in previous embedding models the label will always be represented by the same vector regardless of its senses.

For some WordNet nodes, they consist of multiple phrases, e.g., ‘book jacket’ and ‘dust cover’. To represent a specific node in \mathcal{S} , previous work tries to find matches in the skip-gram model for every phrase, and average the corresponding vectors [9]. Since we are interested in learning the skip-gram model from Flickr tags, which are however mostly single words instead of phrases, we consider a less strict rule that makes matches in terms of single words. As opposed to the previous full match, we term this rule as partial match.

Putting $s_{hi}(y)$ into Eq. (1), we obtain $f_{hi}(x)$ as the hierarchically embedded semantic vector of an image. Accordingly, we get a new classifier as $\cos(f_{hi}(x), s_{hi}(y))$. The hyper-parameter T is set to be 10, according to [9]. An overview of the proposed model is visualized in Fig. 1.

3. EMPIRICAL EVALUATION

This section presents an evaluation to verify our proposal, compared with the baseline model [9]. For reproducibility, the evaluation is based on public data.

3.1 Setup

Training Label Set \mathcal{Y}_0 . Following [9], we use the ImageNet 1K label set as \mathcal{Y}_0 , including 1,000 visual object classes defined in the Large Scale Visual Recognition Challenge 2012 [10]. A pre-trained DeepNet model provided by Caffe [4] is used as $p_0(y|x)$. It is a stacked convolutional neural network [6], consisting of five convolutional layers followed by two fully connected layers and a softmax output layer. The model is learned from over 1m labeled examples.

Test Label Set \mathcal{Y}_1 . Following [9], the test label set consist of labels within 2 tree hops of \mathcal{Y}_0 , namely their direct parent and child nodes, resulting in 1,548 novel labels in total. The ground-truthed test images are from ImageNet [2], with the number of relevant images per label ranges from 1 to 2,330, with an average number of 846. The total number of test images is 1.3m.

Semantic Space \mathcal{S} . Similar to [9], \mathcal{S} in this work is also trained by a skip-gram model, using the word2vec software [8]. We train a 500-D model with 2.2m words using the latest Wikipedia dump. Another 500-D model with 382k words is learned from user tags of 4 million Flickr images. Additionally, we experiment with a pre-trained model with 3m words [8], trained on a Google News dataset.

Performance Metric. We report $\text{hit}@k$, which is the percentage of test images for which the true label is among the top- k predicted tags.

3.2 Experiments

In order to study 1) which resources to use for constructing the semantic space, 2) how to convert a WordNet node to a semantic embedding vector, and 3) if adding the hierarchy in the embedding process is helpful, we conduct the following three experiments. For the ease of comparison, we abbreviate the proposed model as HierSE, and name the baseline model [9] as FlatSE.

Table 1: Performance of different models. Numbers marked with star (*) are taken from [9]. Bold font indicates the top performer in different settings.

Model	Resource for word2vec	hit@k (%)			
		1	2	5	10
FlatSE [9]	Wikipedia	*9.4	*15.1	*24.7	*32.7
<i>Full match:</i>					
FlatSE	Wikipedia	7.5	12.1	19.4	25.3
FlatSE	Google	7.5	11.7	18.8	24.9
FlatSE	Flickr	9.3	14.7	22.6	29.1
HierSE	Wikipedia	14.5	23.1	36.4	45.1
HierSE	Google	15.6	24.4	38.3	48.2
HierSE	Flickr	16.2	25.0	40.7	52.0
<i>Partial match:</i>					
FlatSE	Wikipedia	13.5	20.8	32.7	40.6
FlatSE	Google	14.3	22.1	33.6	41.5
FlatSE	Flickr	15.7	24.7	39.4	49.9
HierSE	Wikipedia	17.3	26.4	39.8	48.9
HierSE	Google	16.9	26.2	40.2	49.9
HierSE	Flickr	18.3	27.9	42.9	54.1





Table 2: Nearest terms to ‘grass’ and ‘hair’, measured using cosine distance in the semantic spaces that are learned from Google News (by [8]) and Flickr tags (by this work), respectively.

Tag	Resource for word2vec	The 10 nearest terms
grass	Google	bermuda_grass, rye_grass, lawns, fescue, zoysia_grass, kikuyu, kikuyu_grass, tall_fescue_grass, buffalograss, zoysia
	Flickr	greengrass, tree, leaf, hedge, fence, pasture, lawn, meadow, graze, field
hair	Google	curly_hair, tresses, mane, hairdo, blonde_hair, blond_hair, gray_hair, wavy_hair, hairstyle, blonde_locks
	Flickr	hairstyle, hairdo, curl, haircare, blackhair, curlyhair, forehead, facial, hairspray, updo

Experiment 1. Which resource for training the semantic space? As shown in Table 1, the Flickr-based word2vec model consistently outperforms its Wikipedia and Google counterparts under varied settings. For a more intuitive understanding, Table 2 shows tags closest, in terms of cosine similarity, to ‘grass’ and ‘hair’ in the Google based and Flickr based semantic spaces, respectively. While subspecies of grass is retrieved in the Google based space, tags depicting the visual context of grass, e.g., ‘tree’, ‘hedge’, and ‘pasture’, are deemed to be more relevant to ‘grass’ in the Flickr based space. Hence, among the three resources compared, we conclude that Flickr is most suited for learning word2vec models for zero-shot image tagging.

Experiment 2. Partial Match or Full Match? We see from Table 1 that given the same embedding model and the same word2vec model, partial match results in a higher hit rate than full match. While full match is more precise, the number of labels that cannot be matched is much larger than its counterpart in partial match. Consider the Flickr based semantic space for instance. The number of test labels that cannot be matched in FlatSE is 749, which is reduced to only 54 when using partial match instead. Also notice that with

Table 3: Top-5 predicted tags, with the correct prediction marked by *italic* (blue) font. Test images are hand-picked to illustrate the cases that the HierSE performs well, and a failure case.

Test Image	Ground truth	$p_0(y x)$ by Caffe	$p_1(y x)$ by FlatSE	$p_1(y x)$ by HierSE
	entellus	gibbon langur Saluki patas spider monkey	crab-eating macaque vervet tamarin guereza striped hyena	<i>entellus</i> vervet crab-eating macaque Barbary ape rhesus
	hamburger	cheeseburger bagel potpie French loaf burrito guacamole meat loaf	pepperoni pizza sausage pizza anchovy pizza cheese pizza <i>hamburger</i> sandwich spaghetti sauce	<i>hamburger</i> sandwich sausage pizza pepperoni pizza anchovy pizza cheese pizza quesadilla
	weight	dumbbell barbell sunglass lab coat sunglasses	shoulder holster toy box spiral ratchet screwdriver carpenter's hammer flat tip screwdriver	<i>weight</i> lifting device airfoil gymnastic apparatus fastener
	dune buggy	thresher go-kart harvester racer lawn mower	car farm machine refrigerator car bicycle machine	farm machine car machine textile machine bulldozer

partial match, our implementation of FlatSE outperforms its original implementation (13.5% versus 9.4% in hit@1).

Experiment 3. HierSE or FlatSE? As Table 1 shows, HierSE beats FlatSE. In particular, when used in combination with the Flickr based semantic space, HierSE almost achieves a doubled accuracy when compared to the number reported in [9] (18.3% versus 9.4% in hit@1). Notice that our implementation of FlatSE scores lower than the original paper. One possible reason is that we use an off-the-shelf DeepNet, which might be less effective than its counterpart in [9]. Some tagging results are provided in Table 3.

4. CONCLUSIONS

This paper contributes to zero-shot image tagging by introducing the WordNet hierarchy into a deep learning based semantic embedding framework. The proposed hierarchical semantic embedding model is found to be effective. In addition, we identify two good practices which further improve the tagging accuracy. That is, using Flickr tags to train the semantic space, and using partial match to embed a WordNet node into the space. Code is available at <https://github.com/li-xirong/hierse>

Acknowledgements. This work was supported by NSFC (No. 61303184), the Fundamental Research Funds for the Central Universities and the Research Funds of Renmin University of China (No. 14XNLQ01).

5. REFERENCES

- [1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-embedding for attribute-based classification. In *CVPR*, 2013.
- [2] 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.
- [3] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. DeViSE: A deep visual-semantic embedding model. In *NIPS*, 2013.
- [4] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACMMM*, 2014.
- [5] S. Kordumova, X. Li, and C. Snoek. Best practices for learning video concept detectors from social media examples. *MTAP*, 74(4):1291–1315, 2015.
- [6] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012.
- [7] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009.
- [8] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. In *ICLR*, 2013.
- [9] 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 embedding. In *ICLR*, 2014.
- [10] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. Berg, and L. Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *arXiv:1409.0575*, 2014.
- [11] S. Zhu, Y.-G. Jiang, and C.-W. Ngo. Sampling and ontologically pooling web images for visual concept learning. *TMM*, 14(4):1068–1078, 2012.