

Fast Zero-Shot Image Tagging

Yang Zhang, Boqing Gong, and Mubarak Shah

Center for Research in Computer Vision, University of Central Florida, Orlando, FL 32816

yangzhang@knights.ucf.edu, bgong@crcv.ucf.edu, shah@crcv.ucf.edu

Abstract

The well-known word analogy experiments show that the recent word vectors capture fine-grained linguistic regularities in words by linear vector offsets, but it is unclear how well the simple vector offsets can encode visual regularities over words. We study a particular image-word relevance relation in this paper. Our results show that the word vectors of relevant tags for a given image rank ahead of the irrelevant tags, along a principal direction in the word vector space. Inspired by this observation, we propose to solve image tagging by estimating the principal direction for an image. Particularly, we exploit linear mappings and nonlinear deep neural networks to approximate the principal direction from an input image. We arrive at a quite versatile tagging model. It runs fast given a test image, in constant time w.r.t. the training set size. It not only gives superior performance for the conventional tagging task on the NUS-WIDE dataset, but also outperforms competitive baselines on annotating images with previously **unseen** tags.

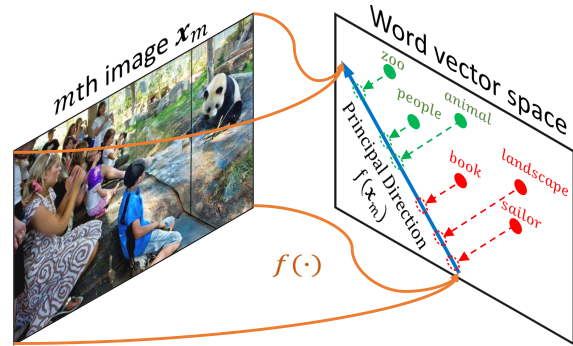


Figure 1: Given an image, its relevant tags’ word vectors rank ahead of the irrelevant tags’ along some direction in the word vector space. We call that direction the **principal direction** for the image. To solve the problem of image tagging, we thus learn a function $f(\cdot)$ to approximate the principal direction from an image. This function takes as the input an image x_m and outputs a vector $f(x_m)$ for defining the principal direction in the word vector space.

1. Introduction

Recent advances in the vector-space representations of words [37, 38, 45] have benefited both NLP [49, 60, 52] and computer vision tasks such as zeros-shot learning [50, 15, 1] and image captioning [31, 26, 27]. The use of word vectors in NLP is grounded on the fact that the fine-grained **linguistic** regularities over words are captured by linear word vector offsets—a key observation from the well-known word analogy experiments [39, 45], such as the syntactic relation $dance - dancing \approx fly - flying$ and semantic relation $king - man \approx queen - woman$. However, it is unclear whether the **visual** regularities over words, which are implicitly used in the aforementioned computer vision problems, can still be encoded by the simple vector offsets.

In this paper, we are interested in the problem of image tagging, where an image (e.g., of a zoo in Figure 1) calls for a partition of a vocabulary of words into two disjoint sets according to the image-word relevance (e.g., relevant tags $Y = \{people, animal, zoo\}$ and irrelevant

ones $\bar{Y} = \{sailor, book, landscape\}$). This partitioning of words, (Y, \bar{Y}) , is essentially different from the fine-grained syntactic (e.g., dance to dancing) or semantic (e.g., king to man) relation tested in the word analogy experiments. Instead, it is about the relationship between two sets of words due to a visual image. Such a relation in words is semantic and descriptive, and focuses on **visual association**, albeit relatively coarser. In this case, do the word vectors still offer the nice property, that the simple linear vector offsets can depict the visual (image) association relations in words? For the example of the zoo, while humans are capable of easily answering that the words in Y are more related to the zoo than those in \bar{Y} , can such zoo-association relation in words be expressed by the 9 pairwise word vector offsets $\{people - sailor, people - book, \dots, zoo - landscape\}$ between the relevant Y and irrelevant \bar{Y} tags’ vectors?

One of the main contributions of this paper is to empirically examine the above two questions (cf. Section 3). Every image introduces a visual association rule (Y, \bar{Y}) over words. Thanks to the large number of images in benchmark

datasets for image tagging, we are able to examine many distinct *visual association regulations* in words and the corresponding *vector offsets* in the word vector space. Our results reveal a somehow surprising connection between the two: the offsets between the vectors of the relevant tags Y and those of the irrelevant \bar{Y} are along about the same direction, which we call the **principal direction**. See Figure 2 for the visualization of some vector offsets. In other words, there exists at least one vector (direction) w in the word vector space, such that its inner products with the vector offsets between Y and \bar{Y} are greater than 0, i.e., $\forall p \in Y, \forall n \in \bar{Y}$,

$$\langle w, p - n \rangle > 0 \text{ equivalently, } \langle w, p \rangle > \langle w, n \rangle, \quad (1)$$

where the latter reads that the vector w *ranks* all relevant words Y (e.g., for the zoo image) ahead of the irrelevant ones \bar{Y} . For brevity, we overload the notations Y and \bar{Y} to respectively denote the vectors of the words in them.

The visual association relations in words thus represent themselves by the (linear) rank-abilities of the corresponding word vectors. This result reinforces the conclusion from the word analogy experiments that, for a single word multiple relations are embedded in the high dimensional space [39, 45]. Furthermore, those relations can be expressed by simple linear vector arithmetic.

Inspired by the above observation, we propose to solve the image tagging problem by estimating the principal direction, along which the relevant tags rank ahead of the irrelevant ones in the word vector space. Particularly, we exploit linear mappings and deep neural networks to approximate the principal direction from each input image. This is a grand new point of view to image tagging and results in a quite versatile tagging model. It operates fast given a test image, in constant time with respect to the training set size. It not only gives superior performance for the conventional tagging task, but is also capable of assigning novel tags from an open vocabulary, which are unseen at the training stage. We do not assume any *a priori* knowledge about these unseen tags as long as they are in the same vector space as the seen tags for training. To this end, we name our approach fast zero-shot image tagging (Fast0Tag) to recognize that it possesses the advantages of both FastTag [7] and zero-shot learning [29, 17, 18].

In sharp contrast to our approach, previous image tagging methods can only annotate test images with the tags seen at training except [18], to the best of our knowledge. Limited by the static and usually small number of seen tags in the training data, these models are frequently challenged in practice. For instance, there are about 53M tags on Flickr and the number is rapidly growing. The work of [18] is perhaps the first attempt to generalize an image tagging model to unseen tags. Compared to the proposed method, it depends on two extra assumptions. One is that the unseen tags are known *a priori* in order to tune the model towards their

combinations. The other is that the test images are known *a priori*, to regularize the model. Furthermore, the generalization of [18] is limited to a very small number, U , of unseen tags, as it has to consider all the 2^U possible combinations.

To summarize, our first main contribution is on the analyses of the visual association relations in words due to images, and how they are captured by word vector offsets. We hypothesize and empirically verify that, for each visual association rule (Y, \bar{Y}) , in the word vector space there exists a principal direction, along which the relevant words' vectors rank ahead of the others'. Built upon this finding, the second contribution is a novel image tagging model, Fast0Tag, which is fast and generalizes to open-vocabulary unseen tags. Last but not least, we explore three different image tagging scenarios: **conventional** tagging which assigns seen tags to images, **zero-shot** tagging which annotates images by (a large number of) unseen tags, and **seen/unseen** tagging which tags images with both seen and unseen tags. In contrast, the existing work tackles either conventional tagging, or zero-shot tagging with very few unseen tags. Our Fast0Tag gives superior results over competitive baselines under all the three testing scenarios.

2. Related work

Image tagging. Image tagging aims to assign relevant tags to an image or to return a ranking list of tags. In the literature this problem has been mainly approached from the tag ranking perspective. In the generative methods, which involve topic models [3, 40, 58, 42] and mixture models [30, 23, 51, 14, 6, 12], the candidate tags are naturally ranked according to their probabilities conditioned on the test image. For the non-parametric nearest neighbor based methods [35, 36, 33, 25, 20, 32, 59], the tags for the test image are often ranked by the votes from some training images. The nearest neighbor based algorithms, in general, outperform those depending on generative models [25, 34], but suffer from high computation costs in both training and testing. The recent FastTag algorithm [7] is magnitude faster and achieves comparable results with the nearest neighbor based methods. Our Fast0Tag shares the same level of low complexity as FastTag. The embedding method [55] assigns ranking scores to the tags by a cross-modality mapping between images and tags. This idea is further exploited using deep neural networks [19]. Interestingly, none of these methods learn their models explicitly for the ranking purpose except [55, 19], although they all rank the candidate tags for the test images. Thus, there exists a mismatch between the models learned and the actual usage of the models, violating the principle of Occam's razor. We use a ranking loss in the same spirit as [55, 19].

In contrast to our Fast0Tag, which can rank both seen and an arbitrary number of unseen tags for test images, the aforementioned approaches only assign tags to images from

a closed vocabulary seen at the training stage. An exception is by Fu et al. [17], where the authors consider pre-fixed U unseen tags and learn a multi-label model to account for all the 2^U possible combinations of them. This method is limited to a small number U of unseen tags.

Word embedding. Instead of representing words using the traditional one-hot vectors, word embedding maps each word to a continuous-valued vector, by learning from primarily the statistics of word co-occurrences. Although there are earlier works on word embedding [46, 11], we point out that our work focuses on the most recent GloVe [45] and word2vec vectors [39, 38, 37]. As shown in the well-known word analogy experiments [39, 45], both types of word vectors are able to capture fine-grained semantic and syntactic regularities using vector offsets. In this paper, we further show that the simple linear offsets also depict the relatively coarser visual association relations in words.

Zero-shot learning. Zero-shot learning is often used exchangeably with zero-shot classification, whereas the latter is a special case of the former. Unlike weakly-supervised learning [41, 16] which learn new concepts by mining noisy new samples, zero-shot classification learns classifiers from seen classes and aims to classify the objects of unseen classes [44, 43, 29, 1, 17, 22, 43, 44, 50]. Attributes [28, 13] and word vectors are two of the main semantic sources making zero-shot classification feasible.

Our Fast0Tag along with [18] enriches the family of zero-shot learning by zero-shot multi-label classification [53]. Fu et al. [18] reduce the problem to zero-shot classification by treating every combination of the multiple labels as a class. We instead directly model the labels and are able to assign/rank many unseen tags for an image.

eat

3. The linear rank-ability of word vectors

Our Fast0Tag approach benefits from the finding that the visual association relation in words, i.e., the partition of a vocabulary of words according to their relevances to an image, expresses itself in the word vector space as the existence of a principal direction, along which the words/tags relevant to the image rank ahead of the irrelevant ones. This section details the finding.

3.1. The regulation over words due to image tagging

We use \mathcal{S} to denote the seen tags available for training image tagging models and \mathcal{U} the tags unseen at the training stage. The training data are in the form of $\{(\mathbf{x}_m, Y_m); m = 1, 2, \dots, M\}$, where $\mathbf{x}_m \in \mathbb{R}^D$ is the feature representation of image m and $Y_m \subset \mathcal{S}$ are the seen tags relevant to that image. For brevity, we overload the notation Y_m to also denote the collection of the corresponding word/tag vectors.

The **conventional** image tagging aims to assign seen tags in \mathcal{S} to the test images. The **zero-shot** tagging, formalized in [18], tries to annotate test images using a pre-fixed set of unseen tags \mathcal{U} . In addition to those two scenarios, this paper considers **seen/unseen** image tagging, which finds both relevant seen tags from \mathcal{S} and relevant unseen tags from \mathcal{U} for the test images. Furthermore, the set of unseen tags \mathcal{U} could be open and dynamically growing.

Denote by $\bar{Y}_m := \mathcal{S} \setminus Y_m$ the irrelevant seen tags. An image m introduces a visual association regulation to words—the partition (Y_m, \bar{Y}_m) of the seen tags to two disjoint sets. Noting that many fine-grained syntactic and semantic regulations over words can be expressed by linear word vector offsets, we next examine what properties the vector offsets could offer for this new visual association rule.

3.2. Principal direction and cluster structure

Figure 2 visualizes the vector offsets $(\mathbf{p} - \mathbf{n})$, $\forall \mathbf{p} \in Y_m, \forall \mathbf{n} \in \bar{Y}_m$ using t-SNE [54] and PCA for two visual association rules over words. One is imposed by an image with 5 relevant tags and the other is with 15 relevant tags. We observe two main structures from the vector offsets:

Principal direction. Mostly, the vector offsets point to about the same direction (relative to the origin), which we call the principal direction, for a given visual association rule (Y_m, \bar{Y}_m) in words for image m . This implies that the relevant tags Y_m rank ahead of the irrelevant ones \bar{Y}_m along the principal direction (cf. eq. (1)).

Cluster structure. There exist cluster structures in the vector offsets for each visual association regulation over the words. Moreover, all the offsets pointing to the same relevant tag in Y_m fall into the same cluster. We differentiate the offsets pointing to different relevant tags by colors in Figure 2.

Can the above two observations generalize? Namely, do they still hold in the high-dimensional word vector space for more visual association rules imposed by other images? To answer the questions, we next design an experiment to verify the existence of the principal directions in word vector spaces, or equivalently the linear rank-ability of word vectors. We leave the cluster structure for future research.

3.3. Testing the linear rank-ability hypothesis

Our experiments in this section are conducted on the validation set (26,844 images, 925 seen tags \mathcal{S} , and 81 unseen tags \mathcal{U}) of NUS-WIDE [8]. The number of relevant seen/unseen tags associated with an image ranges from 1 to 20/117 and on average is 1.7/4.9. See Section 5 for details.

Our objective is to investigate, for any visual association rule (Y_m, \bar{Y}_m) in words by image m , the existence of the principal direction along which the relevant tags Y_m rank ahead of the irrelevant tags \bar{Y}_m . The proof completes once

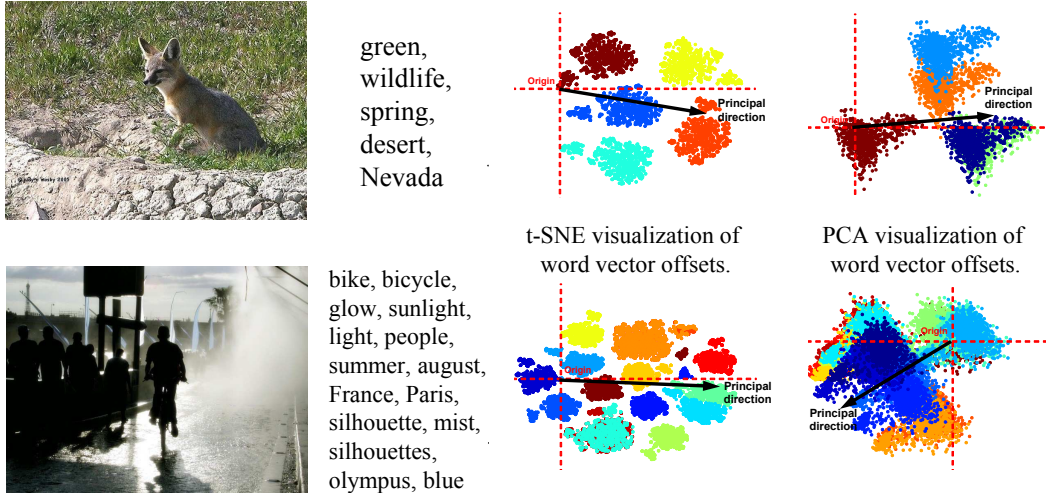


Figure 2: Visualization of the offsets between relevant tags’ word vectors and irrelevant ones’. *Note that each vector from the origin to a point is an offset between two word vectors.* The relevant tags are shown beside the images [8].

we find a vector w in the word vector space that satisfies the ranking constraints $\langle w, p \rangle > \langle w, n \rangle, \forall p \in Y_m, \forall n \in \bar{Y}_m$. To this end, we train a linear ranking SVM [24] for each visual association rule using all the corresponding pairs (p, n) , then rank the word vectors by the SVM, and finally examine how many constraints are violated. In particular, we employ MiAP, the larger the better (cf. Section 5), to compare the SVM’s ranking list with those ranking constraints. We repeat the above process for all the validation images, resulting in 21,863 unique visual association rules.

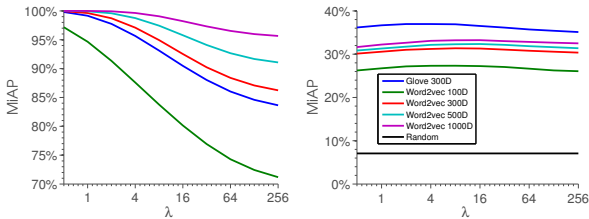


Figure 3: The existence (left) and generalization (right) of the principal direction for each visual association rule in words induced by an image.

Results. The MiAP results averaged over all the distinct regulations are reported in Figure 3(left), in which we test the 300D GloVe vectors [45] and word2vec [39] of dimensions 100, 300, 500, and 1000. The horizontal axis shows different regularizations we use for training the ranking SVMs. Larger λ regularizes the models more (details in Suppl.). In the 300D GloVe space and the word2vec spaces of 300, 500, and 1000 dimensions, more than two ranking SVMs, with small λ values, give rise to nearly perfect ranking results (MiAP ≈ 1), showing that the seen tags \mathcal{S} are linearly rank-able under almost every visual association rule—all the ranking constraints imposed by the relevant Y_m and irrelevant \bar{Y}_m tags to image m are satisfied.

However, we shall be cautious before drawing any conclusions *beyond* the experimental vocabulary \mathcal{S} of seen tags.

An image m incurs a visual association rule essentially over all words, though the same rule implies different partitions of distinct experimental vocabularies (e.g., the seen tags \mathcal{S} and unseen ones \mathcal{U}). Accordingly, we would expect the principal direction for the seen tags is also shared by the unseen tags under the same rule, if the answer is YES to the questions at the end of Section 3.2.

Generalization to unseen tags. We test whether the same principal direction exists for the seen tags and unseen ones under every visual association rule induced by an image. This can be (only partially) justified by applying the ranking SVMs previously learned, to the unseen tags’ vectors, because we do not know the “true” principal directions. We consider the with 81 unseen tags \mathcal{U} as the “test data” for the trained ranking SVMs, each due to an image incurred visual association. NUS-WIDE provides the annotations of the 81 tags for the images. The results, shown in Figure 3(right), are significantly better than the most basic baseline, randomly ranking the tags (the black curve close to the origin), demonstrating that the directions output by SVMs are generalizable to the new vocabulary \mathcal{U} of words.

Observation. Therefore, we conclude that the word vectors are an efficient media to transfer knowledge—the rankability along the principal direction—from the seen tags to the unseen ones. We have empirically verified that the visual association rule (Y_m, \bar{Y}_m) in words due to an image m can be represented by the linear rank-ability of the corresponding word vectors along a principal direction. Our experiments involve $|\mathcal{S}| + |\mathcal{U}| = 1,006$ words in total. Larger-scale and theoretical studies are required for future work.

4. Approximating the linear ranking functions

This section presents our Fast0Tag approach to image tagging. We first describe how to solve image tagging by

approximating the principal directions thanks to their existence and generalization, empirically verified in the last section. We then describe detailed approximation techniques.

4.1. Image tagging by ranking

Grounded on the observation from Section 3, that there exists a principal direction w_m , in the word vector space, for every visual association rule (Y_m, \bar{Y}_m) in words by an image m , we propose a straightforward solution to image tagging. The main idea is to approximate the principal direction by learning a mapping function $f(\cdot)$, between the visual space and the word vector space, such that

$$f(x_m) \approx w_m, \quad (2)$$

where x_m is the visual feature representation of the image m . Therefore, given a test image x , we can immediately suggest a list of tags by ranking the word vectors of the tags along the direction $f(x)$, namely, by the ranking scores,

$$\langle f(x), t \rangle, \quad \forall t \in \mathcal{S} \cup \mathcal{U} \quad (3)$$

no matter the tags are from the seen set \mathcal{S} or unseen set \mathcal{U} .

We explore both linear and nonlinear neural networks for implementing the approximation function $f(x) \approx w$.

4.2. Approximation by linear regression

Here we assume a linear function from the input image representation x to the output principal direction w , i.e.,

$$f(x) := Ax, \quad (4)$$

where A can be solved in a closed form by linear regression. Accordingly, we have the following from the training

$$w_m = Ax_m + \epsilon_m, m = 1, 2, \dots, M \quad (5)$$

where w_m is the principal direction of all offset vectors of the seen tags, for the visual association rule (Y_m, \bar{Y}_m) due to the image m , and ϵ_m are the errors. Minimizing the mean squared errors gives us a closed form solution to A .

One caveat is that we do not know the exact principal directions w_m at all—the training data only offer images x_m and the relevant tags Y_m . Here we take the easy alternative and use the directions found by ranking SVMs (cf. Section 3) in eq. (5). There are thus *two stages* involved to learn the linear function $f(x) = Ax$. The first stage trains a ranking SVM over the word vectors of seen tags for each visual association (Y_m, \bar{Y}_m) . The second stage solves for the mapping matrix A by linear regression, in which the targets are the directions returned by the ranking SVMs.

Discussion. We note that the linear transformation between visual and word vector spaces has been employed before, e.g., for zero-shot classification [1, 15] and image

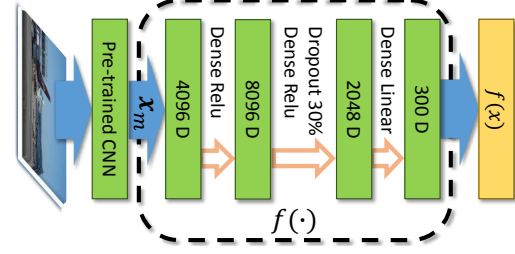


Figure 4: The neural network used in our approach for implementing the mapping function $f(x; \theta)$ from the input image, which is represented by the CNN features x , to its corresponding principal direction in the word vector space.

annotation/classification [56]. This work differs from them with a prominent feature, that the mapped image $f(x) = Ax$ has a clear meaning; it depicts the principal direction, which has been empirically verified, for the tags to be assigned to the image. We next extend the linear transformation to a nonlinear one, through a neural network.

4.3. Approximation by neural networks

We also exploit a nonlinear mapping $f(x; \theta)$ by a multi-layer neural network, where θ denotes the network parameters. Figure 4 shows the network architecture. It consists of two RELU layers followed by a linear layer to output the approximated principal direction, w , for an input image x . We expect the nonlinear mapping function $f(x; \theta)$ to offer better modeling flexibility than the linear one.

Can we still train the neural network by regressing to the M directions obtained from ranking SVMs? Both our intuition and experiments tell that this is a bad idea. The number M of training instances is small relative to the number of parameters in the network, making it hard to avoid overfitting. Furthermore, the directions by ranking SVMs are not the true principal directions anyway. There is no reason for us to stick to the ranking SVMs for the principal directions.

We instead unify the two stages in Section 4.2. Recall that we desire the output of the neural network $f(x_m; \theta)$ to be the principal direction, along which all the relevant tag vectors $p \in Y_m$ of an image m rank ahead of the irrelevant ones $n \in \bar{Y}_m$. Denote by

$$\nu(p, n; \theta) = \langle f(x_m; \theta), n \rangle - \langle f(x_m; \theta), p \rangle,$$

the amount of violation to any of those ranking constraints. We minimize the following loss to train the neural network,

$$\theta^* \leftarrow \arg \min_{\theta} \sum_{m=1}^M \omega_m \ell(x_m, Y_m; \theta), \quad (6)$$

$$\ell(x_m, Y_m; \theta) = \sum_{p \in Y_m} \sum_{n \in \bar{Y}_m} \log(1 + \exp\{\nu(p, n; \theta)\})$$

where $\omega_m = (|Y_m| |\bar{Y}_m|)^{-1}$ normalizes the per-image RankNet loss [5] $\ell(x_m, Y_m; \theta)$ by the number of ranking

constraints imposed by the image m over the tags. This formulation enables the function $f(x)$ to directly take account of the ranking constraints by relevant p and irrelevant n tags. Moreover, it can be optimized with no challenge at all by standard mini-batch gradient descent.

Practical considerations. We use Theano [4] to solve the optimization problem. A mini-batch consists of 1,000 images, each of which incurs on average 4,600 pairwise ranking constraints of the tags—we use all pairwise ranking constraints in the optimization. The normalization ω_m for the per-image ranking loss suppresses the violations from the images with many positive tags. This is desirable since the numbers of relevant tags of the images are unbalanced, ranging from 1 to 20. Without the normalization the MiAP results drop by about 2% in our experiments. For regularization, we use early stopping and a dropout layer [21] with the drop rate of 30%. The optimization hyper-parameters are selected by the validation set (cf. Section 5).

In addition to the RankNet loss [5] in eq. (6), we have also experimented some other choices for the per-image loss, including the hinge loss [9], Crammer-Singer loss [10], and pairwise max-out ranking [24]. The hinge loss performs the worst, likely because it is essentially not designed for ranking problems, though one can still understand it as a point-wise ranking loss. The Crammer-Singer, pairwise max-out, and RankNet are all pair-wise ranking loss functions. They give rise to comparable results and RankNet outperforms the other two by about 2% in terms of MiAP. This may attribute to the ease of control over the optimization process for RankNet. Finally, we note that the list-wise ranking loss [57] can also be employed.

5. Experiments

This section presents our experimental results. We contrast our approach to several competitive baselines for the conventional image tagging task on the large-scale NUS-WIDE [8] dataset. Moreover, we also evaluate our method on the zero-shot and seen/unseen image tagging problems (cf. Section 3.1). For the comparison on these problems, we extend some existing zero-shot classification algorithms and consider some variations of our own approach.

5.1. Dataset and configuration

NUS-WIDE. We mainly use the NUS-WIDE dataset [8] for the experiments in this section (more results in Suppl.). NUS-WIDE is a standard benchmark dataset for image tagging. It contains 269,648 images in the original release and we are able to retrieve 223,821 of them since some images are either corrupted or removed from Flickr. We follow the recommended experiment protocol to split the dataset into a training set with 134,281 images and a test set with 89,603 images. We further randomly separate 20% from the train-

ing set as our validation set for 1) tuning hyper-parameters in our method and the baselines and 2) conducting the empirical analyses in Section 3.

Annotations of NUS-WIDE. NUS-WIDE releases three sets of tags associated with the images. The first set comprises of 81 “groundtruth” tags. They are carefully chosen to be representative of the Flickr tags, such as containing both general terms (e.g., *animal*) and specific ones (e.g., *dog* and *flower*), corresponding to frequent tags on Flickr, etc. Moreover, they are annotated by high-school and college students and are much less noisy than those directly collected from the Web. This 81-tag set is usually taken as the groundtruth for benchmarking different image tagging methods. The second and the third sets of annotations are both harvested from Flickr. There are 1,000 popular Flickr tags in the second set and nearly 5,000 raw tags in the third.

Image features and word vectors. We extract and ℓ_2 normalize the image feature representations of VGG-19 [48]. Both GloVe [45] and Word2vec [39] word vectors are included in our empirical analysis experiments in Section 3 and the 300D GloVe vectors are used for the remaining experiments. We also ℓ_2 normalize the word vectors.

Evaluation. We evaluate the tagging results of different methods using two types of metrics. One is the mean image average precision (MiAP), which takes the whole ranking list into account. The other consists of the precision, recall, and F-1 score for the top K tags in the list. We report the results for $K = 3$ and $K = 5$. Both metrics are commonly used in the previous works on image tagging. We refer the readers to Section 3.3 of [34] for how to calculate MiAP and to Section 4.2 of [19] for the top- K precision and recall.

5.2. Conventional image tagging

Here we report the experiments on the **conventional** tagging. The 81 concepts with “groundtruth” annotations in NUS-WIDE are used to benchmark different methods.

Baselines. We include TagProp [20] as the first competitive baseline. It is representative among the nearest neighbor based methods, which in general outperform the parametric methods built from generative models [3, 6], and gives rise to state-of-the-art results in the experimental study [34]. We further compare with two most recent parametric methods, WARP [19] and FastTag [7], both of which are built upon deep architectures though using different models. For a fair comparison, we use the same VGG-19 features for all the methods—the code of TagProp and FastTag is provided by the authors and we implement WARP based on our neural network architecture. Finally, we compare to WSABIE [56] and CCA, both correlating images and relevant tags in a low dimensional space. All the hyper-parameters (e.g., the number of nearest neighbors in TagProp and early stopping for WARP) are selected using the validation set.

Table 2: Comparison results of the **zero-shot** and **seen/unseen** image tagging tasks with 81 unseen tags and 925 seen tags.

Method %	Zero-shot image tagging							Seen/unseen image tagging						
	MiAP	$K = 3$			$K = 5$			MiAP	$K = 3$			$K = 5$		
		P	R	F1	P	R	F1		P	R	F1	P	R	F1
Random	7.1	2.2	3.8	2.8	2.2	6.1	3.2	1.2	0.6	0.3	0.4	0.6	0.5	0.5
Seen2Unseen	16.7	7.3	12.5	9.2	7.0	19.7	10.3	2.8	2.1	1.1	1.4	1.9	1.6	1.8
LabelEM [2]	23.7	11.9	20.2	14.9	10.2	28.9	15.1	8.8	8.7	4.4	5.8	7.9	6.6	7.2
LabelEM+ [2]	24.9	12.5	21.4	15.8	10.7	30.4	15.8	10.2	11.3	5.7	7.6	9.6	8.1	8.8
ConSE [43]	32.4	17.7	30.1	22.3	13.7	38.8	20.2	12.5	16.7	8.4	11.2	13.5	11.3	12.3
Fast0Tag (lin.)	40.1	21.8	37.2	27.5	17.0	48.4	25.2	18.8	22.9	11.5	15.4	18.7	15.7	17.1
Fast0Tag (net.)	42.2	22.6	38.4	28.4	17.6	50.0	26.0	19.1	21.7	11.0	14.5	18.4	15.5	16.8
RankSVM	37.0	19.7	33.3	24.7	15.2	42.9	22.5	—	—	—	—	—	—	—

Table 1: Comparison results of the **conventional** image tagging with 81 tags on NUS-WIDE.

Method %	MiAP	$K = 3$			$K = 5$		
		P	R	F1	P	R	F1
CCA	19	9	15	11	7	20	11
WSABIE [56]	28	16	27	20	12	35	18
TagProp [20]	53	29	50	37	22	62	32
WARP [19]	48	27	45	34	20	57	30
FastTag [7]	41	23	39	29	19	54	28
Fast0Tag (lin.)	52	29	50	37	21	60	31
Fast0Tag (net.)	55	31	52	39	23	65	34

Results. Table 1 shows the comparison results of TagProp, WARP, FastTag, WSABIE, CCA, and our Fast0Tag models implemented respectively by the linear mapping and nonlinear neural network. We can see that TagProp performs significantly better than WARP and FastTag. However, TagProp’s training and test complexities are very high, being respectively $O(M^2)$ and $O(M)$ w.r.t. the training set size M . In contrast, both WARP and FastTag are more efficient, with $O(M)$ training complexity and constant testing complexity, thanks to their parametric formulation. Our Fast0Tag with linear mapping gives comparable results to TagProp and Fast0Tag with the neural network outperforms the other methods. Also, both implementations have as low computation complexities as WARP and FastTag.

5.3. Zero-shot and Seen/Unseen image tagging

This section presents some results for the two novel image tagging scenarios, **zero-shot** and **seen/unseen** tagging.

Fu et al. [18] formalized the **zero-shot** image tagging problem, aiming to annotate test images using a pre-fixed set \mathcal{U} of unseen tags. Our Fast0Tag naturally applies to this scenario, by simply ranking the unseen tags with eq. (3). Furthermore, this paper also considers **seen/unseen** image tagging which finds both relevant seen tags from \mathcal{S} and relevant unseen tags from \mathcal{U} for the test images. The set of unseen tags \mathcal{U} could be open and dynamically growing.

In our experiments, we treat the 81 concepts with high-quality user annotations in NUS-WIDE as the unseen set \mathcal{U}

for evaluation and comparison. We use the remaining 925 out of the 1000 frequent Flickr tags to form the seen set \mathcal{S} —75 tags are shared by the original 81 and 1,000 tags.

Baselines. Our Fast0Tag models can be readily applied to the zero-shot and seen/unseen image tagging scenarios. For comparison we study the following baselines.

Seen2Unseen. We first propose a simple method which extends an arbitrary traditional image tagging method to also working with previously unseen tags. It originates from our analysis experiment in Section 3. First, we use any existing method to rank the seen tags for a test image. Second, we train a ranking SVM in the word vector space using the ranking list of the seen tags. Third, we rank unseen (and seen) tags using the learned SVM for zero-shot (and seen/unseen) tagging.

LabelEM. The label embedding method [2] achieves impressive results on zero-shot classification for fine-grained object recognition. If we consider each tag of $\mathcal{S} \cup \mathcal{U}$ as a unique class, though this implies that some classes will have duplicated images, the LabelEM can be directly applied to the two new tagging scenarios.

LabelEM+. We also modify the objective loss function of LabelEM when we train the model, by carefully removing the terms that involve duplicated images. This slightly improves the performance of LabelEM.

ConSE. Again by considering each tag as a class, we include a recent zero-shot classification method, ConSE [43] in the following experiments.

Note that it is computationally infeasible to compare with [18], which might be the first work to our knowledge on expanding image tagging to handle unseen tags, because it considers all the possible combinations of the unseen tags.

Results. Table 2 summarizes the results of the baselines and Fast0Tag when they are applied to the zero-shot and seen/unseen image tagging tasks. Overall, Fast0Tag, with either linear or neural network mapping, performs the best.

Additionally, in the table we add two special rows whose results are mainly for reference. The Random row corresponds to the case when we return a random list of tags in

Table 3: Annotating images with up to 4,093 unseen tags.

Method	%	MiAP	$K = 3$			$K = 5$		
			P	R	F1	P	R	F1
Random		0.3	0.1	0.1	0.1	0.1	0.1	0.1
Fast0Tag (lin.)		9.8	9.4	7.2	8.2	7.4	9.5	8.4
Fast0Tag (net.)		8.5	8.0	6.2	7.0	6.5	8.3	7.3

\mathcal{U} for zero-shot tagging (and in $\mathcal{U} \cup \mathcal{S}$ for seen/unseen tagging) to each test image. We compare this row with the row of Seen2Unseen, in which we extend TagProp to handle the unseen tags. We can see that the results of Unseen2Seen are significantly better than randomly ranking the tags. This tells us that the simple Seen2Unseen is effective in expanding the labeling space of traditional image tagging methods. Some tag completion methods [47] may also be employed for the same purpose as Seen2Unseen.

Another special row in Table 2 is the last one with RankSVM for zero-shot image tagging. We obtain its results through the following steps. Given a test image, we assume the annotation of the seen tags, \mathcal{S} , are known and then learn a ranking SVM with the default regularization $\lambda = 1$. The learned SVM is then used to rank the unseen tags for this image. One may wonder that the results of this row should thus be the upper bound of our Fast0Tag implemented based on linear regression, because the ranking SVM models are the targets of the linear regression. However, the results show that they are not. This is not surprising, but rather it reinforces our previous statement that the learned ranking SVMs are not the “true” principal directions. The Fast0Tag implemented by the neural network is an effective alternative for seeking the principal directions.

It would also be interesting to compare the results in Table 2 (zero-shot image tagging) with those in Table 1 (conventional tagging), because the experiments for the two tables share the same testing images and the same candidate tags; they only differ in which tags are used for training. We can see that the Fast0Tag (net.) results of the zero-shot tagging in Table 2 are actually comparable to the conventional tagging results in Table 1, particularly about the same as FastTag’s. These results are encouraging, indicating that it is unnecessary to use all the candidate tags for training in order to have high-quality tagging performance.

Annotating images with 4,093 unseen tags. What happens when we have a large number of unseen tags showing up at the test stage? NUS-WIDE provides noisy annotations for the images with over 5,000 Flickr tags. Excluding the 925 seen tags that are used to train models, there are 4,093 remaining unseen tags. We use the Fast0Tag models to rank all the unseen tags for the test images and the results are shown in Table 3. Noting that the noisy annotations weaken the credibility of the evaluation process, the results are reasonably low but significantly higher than the random lists.

Qualitative results. Figure 5 shows the top five tags for some exemplar images [8], returned by Fast0Tag under






Images	Conventional Tagging	Zero-Shot Tagging	See/Unseen Tagging	4k Zero-Shot Tagging	TagProp (Conventional)
	Water Beach Ocean Surf <i>Cat</i>	Water Ocean Surf Beach <i>Snow</i>	Water Ocean Wave Sea Surf	Water Ocean <i>NGO</i> <i>Dam</i> Surf	Water Surf Ocean Beach <i>Whales</i>
	Reflection Water Building Cityscape Harbor	Cityscape Sunset <i>Bridge</i> Reflection Harbor	Night Skyline Cityscape Sunset City	Waterfront <i>Danbe</i> Cityscape Sunset <i>Venice</i>	Water Reflection <i>Sky</i> Building Cityscape
	Coral Fish Ocean Water <i>Rocks</i>	Coral Fish Water Ocean <i>Sand</i>	Coral Underwater Marine Scuba Reef	Coral <i>Korea</i> <i>Mushroom</i> <i>Lichen</i> Fish	Coral Fish Water Ocean <i>Flowers</i>
	Plane Airport Sky Clouds <i>Military</i>	Plane Sky Airport <i>Snow</i> Clouds	Aircraft <i>Aviation</i> Airplane Jet Flying	Aircrafts <i>Takeoff</i> Jets Airlines	Airport Plane Clouds <i>Military</i>
	Train Railroad <i>Bridge</i> Road <i>Fire</i>	Railroad Train <i>Bridge</i> Road <i>Fire</i>	Locomotive Railroad Railway Train Rail	Locomotives Railroad Railways Train Trains	Train Railroad <i>Clouds</i> <i>Sky</i> Road

Figure 5: The top five tags for exemplar images [8] returned by Fast0Tag on the conventional, zero-shot, and seen/unseen image tagging tasks, and by TagProp for conventional tagging. (Correct tags: green; mistaken tags: red and *italic*. Best viewed in color.)

the conventional, zero-shot, and seen/unseen image tagging scenarios. Those by TagProp under the conventional tagging are shown on the rightmost. The tags in green color appear in the groundtruth annotation; those in red color and *italic* font are the mistaken tags. Interestingly, Fast0Tag performs equally well for traditional and zero-shot tagging and makes even the same mistakes. More results are in Suppl.

6. Conclusion

We have systematically studied a particular visual regulation over words, the visual association rule which partitions words into two disjoint sets according to their relevances to an image, as well as how it can be captured by the vector offsets in the word vector space. Our empirical results show that, for any image, there exists a principal direction in the word vector space such that the relevant tags’ vectors rank ahead of the irrelevant ones’ along that direction. The experimental analyses involve 1,006 words; larger-scale and theoretical analyses are required for future work. Built upon this observation, we develop a Fast0Tag model to solve image tagging by estimating the principal directions for input images. Our approach is as efficient as FastTag [7] and is capable of annotating images with a large number of previously **unseen** tags. Extensive experiments validate the effectiveness of our Fast0Tag approach.

Acknowledgments

This work is partially supported by NSF IIS 1566511. We thank the anonymous area chair and reviewers, especially the assigned Reviewer 30, for their helpful comments.

References

- [1] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid. Label-Embedding for Attribute-Based Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 819–826, June 2013.
- [2] Z. Akata, S. Reed, D. Walter, H. Lee, and B. Schiele. Evaluation of Output Embeddings for Fine-Grained Image Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2927–2936, 2015.
- [3] K. Barnard, P. Duygulu, D. Forsyth, N. De Freitas, D. M. Blei, and M. I. Jordan. Matching words and pictures. *The Journal of Machine Learning Research*, 3:1107–1135, 2003.
- [4] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley, and Y. Bengio. Theano: a CPU and GPU math expression compiler. In *Proceedings of the Python for scientific computing conference (SciPy)*, volume 4, page 3. Austin, TX, 2010.
- [5] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of the 22nd international conference on Machine learning*, pages 89–96. ACM, 2005.
- [6] G. Carneiro, A. B. Chan, P. J. Moreno, and N. Vasconcelos. Supervised learning of semantic classes for image annotation and retrieval. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):394–410, 2007.
- [7] M. Chen, A. Zheng, and K. Weinberger. Fast image tagging. In *Proceedings of the 30th international conference on Machine Learning*, pages 1274–1282, 2013.
- [8] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng. NUS-WIDE: a real-world web image database from National University of Singapore. In *Proceedings of the ACM international conference on image and video retrieval*, page 48. ACM, 2009.
- [9] C. Cortes and V. Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.
- [10] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2:265–292, 2002.
- [11] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.
- [12] A. Dehghan, H. Idrees, and M. Shah. Improving Semantic Concept Detection through the Dictionary of Visually-distinct Elements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2585–2592, 2014.
- [13] A. Farhadi, I. Endres, D. Hoiem, and D. Forsyth. Describing objects by their attributes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1778–1785. IEEE, 2009.
- [14] S. Feng, R. Manmatha, and V. Lavrenko. Multiple bernoulli relevance models for image and video annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages II–1002. IEEE, 2004.
- [15] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, and others. Devise: A deep visual-semantic embedding model. In *Advances in Neural Information Processing Systems*, pages 2121–2129, 2013.
- [16] J. Fu, Y. Wu, T. Mei, J. Wang, H. Lu, and Y. Rui. Relaxing From Vocabulary: Robust Weakly-Supervised Deep Learning for Vocabulary-Free Image Tagging. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1985–1993, 2015.
- [17] Y. Fu, Y. Yang, T. Hospedales, T. Xiang, and S. Gong. Transductive Multi-label Zero-shot Learning. pages 7.1–7.11. British Machine Vision Association, 2014.
- [18] Y. Fu, Y. Yang, T. M. Hospedales, T. Xiang, and S. Gong. Transductive Multi-class and Multi-label Zero-shot Learning. *arXiv preprint arXiv:1503.07884*, 2015.
- [19] Y. Gong, Y. Jia, T. Leung, A. Toshev, and S. Ioffe. Deep convolutional ranking for multilabel image annotation. *arXiv preprint arXiv:1312.4894*, 2013.
- [20] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Tagprop: Discriminative metric learning in nearest neighbor models for image auto-annotation. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 309–316. IEEE, 2009.
- [21] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*, 2012.
- [22] D. Jayaraman and K. Grauman. Zero-shot recognition with unreliable attributes. In *Advances in Neural Information Processing Systems*, pages 3464–3472, 2014.
- [23] J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 119–126. ACM, 2003.
- [24] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM, 2002.
- [25] M. Kalayeh, H. Idrees, and M. Shah. NMF-KNN: Image Annotation Using Weighted Multi-view Non-negative Matrix Factorization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 184–191, June 2014.
- [26] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. *arXiv preprint arXiv:1412.2306*, 2014.
- [27] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *arXiv preprint arXiv:1411.2539*, 2014.
- [28] C. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 951–958, June 2009.
- [29] C. H. Lampert, H. Nickisch, and S. Harmeling. Attribute-based classification for zero-shot visual object categorization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(3):453–465, 2014.

- [30] V. Lavrenko, R. Manmatha, and J. Jeon. A model for learning the semantics of pictures. In *Advances in neural information processing systems*, page None, 2003.
- [31] R. Lebre, P. Pinheiro, and R. Collobert. Phrase-based Image Captioning. In D. Blei and F. Bach, editors, *Proceedings of the 32nd International Conference on Machine Learning (ICML-15)*, pages 2085–2094. JMLR Workshop and Conference Proceedings, 2015.
- [32] S. Lee, W. De Neve, and Y. M. Ro. Visually weighted neighbor voting for image tag relevance learning. *Multimedia tools and applications*, 72(2):1363–1386, 2014.
- [33] X. Li, C. G. Snoek, and M. Worring. Learning social tag relevance by neighbor voting. *Multimedia, IEEE Transactions on*, 11(7):1310–1322, 2009.
- [34] X. Li, T. Uricchio, L. Ballan, M. Bertini, C. G. M. Snoek, and A. Del Bimbo. Socializing the Semantic Gap: A Comparative Survey on Image Tag Assignment, Refinement and Retrieval. *arXiv:1503.08248 [cs]*, Mar. 2015. arXiv: 1503.08248.
- [35] A. Makadia, V. Pavlovic, and S. Kumar. Baselines for image annotation. *International Journal of Computer Vision*, 90(1):88–105, 2010.
- [36] T. Mei, Y. Wang, X.-S. Hua, S. Gong, and S. Li. Coherent image annotation by learning semantic distance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [37] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv:1301.3781 [cs]*, Jan. 2013. arXiv: 1301.3781.
- [38] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [39] T. Mikolov, W.-t. Yih, and G. Zweig. Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, pages 746–751, 2013.
- [40] F. Monay and D. Gatica-Perez. PLSA-based image auto-annotation: constraining the latent space. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 348–351. ACM, 2004.
- [41] E. Moxley, T. Mei, and B. S. Manjunath. Video annotation through search and graph reinforcement mining. *Multimedia, IEEE Transactions on*, 12(3):184–193, 2010.
- [42] Z. Niu, G. Hua, X. Gao, and Q. Tian. Semi-supervised relational topic model for weakly annotated image recognition in social media. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4233–4240. IEEE, 2014.
- [43] M. Norouzi, T. Mikolov, S. Bengio, Y. Singer, J. Shlens, A. Frome, G. S. Corrado, and J. Dean. Zero-shot learning by convex combination of semantic embeddings. *arXiv preprint arXiv:1312.5650*, 2013.
- [44] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell. Zero-shot learning with semantic output codes. In *Advances in neural information processing systems*, pages 1410–1418, 2009.
- [45] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP 2014)*, 12, 2014.
- [46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. Technical report, DTIC Document, 1985.
- [47] B. Sigurbjörnsson and R. Van Zwol. Flickr tag recommendation based on collective knowledge. In *Proceedings of the 17th international conference on World Wide Web*, pages 327–336. ACM, 2008.
- [48] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [49] R. Socher, J. Bauer, C. D. Manning, and A. Y. Ng. Parsing with compositional vector grammars. In *In Proceedings of the ACL conference*. Citeseer, 2013.
- [50] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *Advances in neural information processing systems*, pages 935–943, 2013.
- [51] A. Tariq and H. Foroosh. Feature-Independent Context Estimation for Automatic Image Annotation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1958–1965, 2015.
- [52] S. Tellex, B. Katz, J. Lin, A. Fernandes, and G. Marton. Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 41–47. ACM, 2003.
- [53] G. Tsoumakas and I. Katakis. Multi-label classification: An overview. *Dept. of Informatics, Aristotle University of Thessaloniki, Greece*, 2006.
- [54] L. Van der Maaten and G. Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579–2605):85, 2008.
- [55] J. Weston, S. Bengio, and N. Usunier. Large scale image annotation: learning to rank with joint word-image embeddings. *Machine learning*, 81(1):21–35, 2010.
- [56] J. Weston, S. Bengio, and N. Usunier. Wsabee: Scaling up to large vocabulary image annotation. In *IJCAI*, volume 11, pages 2764–2770, 2011.
- [57] F. Xia, T.-Y. Liu, J. Wang, W. Zhang, and H. Li. Listwise approach to learning to rank: theory and algorithm. In *Proceedings of the 25th international conference on Machine learning*, pages 1192–1199. ACM, 2008.
- [58] O. Yakhnenko and V. Honavar. Annotating images and image objects using a hierarchical dirichlet process model. In *Proceedings of the 9th International Workshop on Multimedia Data Mining: held in conjunction with the ACM SIGKDD 2008*, pages 1–7. ACM, 2008.
- [59] X. Zhu, W. Nejdl, and M. Georgescu. An adaptive teleportation random walk model for learning social tag relevance. In *Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval*, pages 223–232. ACM, 2014.
- [60] W. Y. Zou, R. Socher, D. M. Cer, and C. D. Manning. Bilingual Word Embeddings for Phrase-Based Machine Translation. In *EMNLP*, pages 1393–1398, 2013.