

# A Support Vector Approach for Cross-Modal Search of Images and Texts

Yashaswi Verma\*, C. V. Jawahar

Center for Visual Information Technology, IIIT Hyderabad, India - 500032  
yashaswi.verma@research.iiit.ac.in, jawahar@iiit.ac.in

## Abstract

Building bilateral semantic associations between images and texts is among the fundamental problems in computer vision. In this paper, we study two complementary cross-modal prediction tasks: (i) predicting text(s) given a query image (“Im2Text”), and (ii) predicting image(s) given a piece of text (“Text2Im”). We make no assumption on the specific form of text; i.e., it could be either a set of labels, phrases, or even captions. We pose both these tasks in a retrieval framework. For Im2Text, given a query image, our goal is to retrieve a ranked list of semantically relevant texts from an independent text-corpus (i.e., texts with no corresponding images). Similarly, for Text2Im, given a query text, we aim to retrieve a ranked list of semantically relevant images from a collection of unannotated images (i.e., images without any associated textual meta-data).

We propose a novel Structural SVM based unified framework for these two tasks, and show how it can be efficiently trained and tested. Using a variety of loss functions, extensive experiments are conducted on three popular datasets (two medium-scale datasets containing few thousands of samples, and one web-scale dataset containing one million samples). Experiments demonstrate that our framework gives promising results compared to competing baseline cross-modal search techniques, thus confirming its efficacy.

**Keywords:** Image search; Image description; Cross-media analysis

## 1. Introduction

During the past decade, there has been a massive explosion of multimedia content on the Internet. As a result, several interesting as well as challenging research problems have emerged, one of them being automatically describing image content using text. While most of the earlier as well as recent research has focused on automatically annotating images using semantic labels [1, 2, 3, 4, 5, 6, 7], in the past few years, describing images using phrases [8, 9, 10, 11], or one or more simple captions [9, 10, 11, 12, 13, 14, 15, 16] have attained significant attention. A complementary problem to these is to automatically associate one or more semantically relevant images given a piece of text (such as label, phrase or caption), and is commonly referred to as the image retrieval task [2, 4, 7, 15, 16, 17, 18, 19].

Although huge amount of *independent* visual and textual data are available today, only a small portion of them is linked with semantic associations. Hence, it comes as a natural choice to develop new models that can efficiently learn the complex associations between the two modalities using this small portion, and later apply them to automatically build associations between the two in the larger, independent space. In this work, we address this problem of learning cross-modal associations between visual and textual data. We study two complementary tasks: (i) retrieving semantically relevant text(s) given a query image (*Im2Text*), and (2) retrieving semantically relevant image(s) given a query text (*Text2Im*). We pose both these tasks as retrieval problems, where the output samples are ranked based

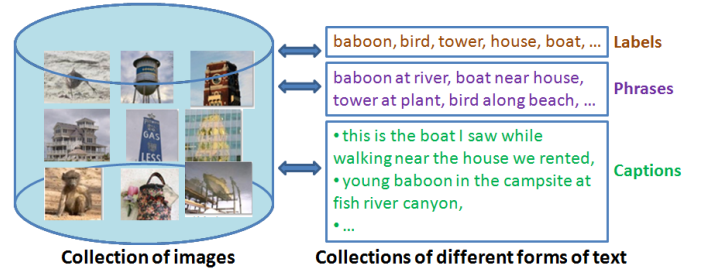


Figure 1: We propose a Structural SVM based unified framework that learns bilateral associations between images and different forms of texts (labels/phrases/captions). Our approach can be used to perform cross-modal retrieval on an independent database of textual data given a query image (“Im2Text”), and vice-versa (“Text2Im”).

on their relevance to the query. In contrast to several existing methods such as [2, 3, 4, 5, 9, 10, 14, 15] that make use of data from both the modalities (image and text) during the prediction phase, our approach is similar to the cross-modal retrieval works like [7, 16, 19, 20, 21] that *do not* make such an assumption. This means that for Im2Text, given a query image, our method can retrieve a ranked list of semantically relevant texts from a plain text-corpus that has no associated images. Similarly, for Text2Im, given a query text, it can retrieve a ranked list of images from an independent collection of images without any associated textual meta-data. Figure 1 illustrates the theme of this work.

The major contributions of this work are:

\*Corresponding author

1. We propose a novel Structural Support Vector Machine (or Structural SVM) [22] based unified framework for both Im2Text and Text2Im, which provides the following three advantages. First, Structural SVM provides a nat-  
 45 tural framework to work with complex and structured input/output spaces, and a unified framework helps in better understanding and appreciating the complementary nature of the two problems. Second, our general-purpose learning module can be easily adopted for different forms,  
 50 of data (diverse modalities with paired cross-modal samples and feature vector based representations) with little modifications. Thirdly, availability of efficient algorithms for Structural SVM training (such as the cutting-plane algorithm [22]) makes it feasible to efficiently learn max-margin models that scale well with data size. As per  
 55 our knowledge, this is the first attempt to examine and validate the applicability of Structural SVM for performing cross-modal multimedia retrieval.
2. Since our framework is based on Structural SVM, it al-  
 60 lows us to learn model parameters using a variety of loss functions. To demonstrate this adaptability, we exam-  
 ine three loss functions in this work. These loss func-  
 tions do not make any assumption on the specific form of data, and also connect well with representations popu-  
 65 larly used for data from diverse modalities.
3. As a part of our experimental analysis, we examine gen-  
 eralization of ours as well as other competing baseline  
 70 methods across datasets when textual data is in the form of captions/descriptions. For this, we learn models from one dataset, and perform retrieval on others.

To validate the applicability of our method, we conduct ex-  
 75 periments on three diverse and popular datasets, namely, UIUC  
 Pascal Sentence dataset [23], IAPR TC-12 benchmark [24], and  
 SBU-Captioned Photo dataset [14]. Among these, Pascal and  
 IAPR datasets are medium scale datasets containing few thou-  
 sands of samples, and SBU is a web-scale dataset containing  
 80 one million samples. Also, while the images in Pascal and SBU  
 datasets are associated with short captions that are a few sen-  
 tences long, those in the IAPR dataset are coupled with long  
 captions that give a detailed description of an image. Exten-  
 sive evaluations on these datasets demonstrate the superiority  
 85 of the proposed framework as compared to competing baseline  
 techniques.

This paper is an extension of our conference version [25].  
 85 Here we build upon this work in the following ways:

1. In addition to the two loss functions described in [25],  
 90 we demonstrate the applicability of our framework using a new loss function (Eq. 7) that is based on normalized correlation. Experiments show that this new loss func-  
 tions usually provides better performance than the two  
 95 proposed in [25].
2. Along with empirical analysis, we provide a deeper com-  
 parison of our approach with competing baseline cross-  
 modal retrieval techniques.
3. We include additional evaluation using recent features for  
 100 images [26] and text [27] on cross-modal image-caption

retrieval task. This validates the applicability of our ap-  
 proach using modern features as well.

4. We include a detailed analysis of the training and run-  
 time efficiency of our approach using synthetic datasets  
 containing up to 0.1 and 10 million samples respectively.
5. We further strengthen the quantitative analysis by using  
 two additional evaluation metrics, and also include qual-  
 itative results. These provide additional insights into our  
 approach.

The paper is organized as follows. In Section 2, we re-  
 view the closely related work. Section 3 describes the proposed  
 approach. In Section 4, we provide a deeper analysis of the  
 proposed approach compared to competing baselines, and ana-  
 lyze the training and testing time in Section 5. Section 6 dis-  
 cusses the representations used for visual and textual data in this  
 work. In Section 7, we present experimental analysis. Finally,  
 Section 8 presents the conclusions and directions for future re-  
 search.

## 2. Related Work

Here, first we discuss related work on unimodal, multi-  
 modal, and cross-modal retrieval, particularly focusing on im-  
 ages and text as the two modalities. Then we review a few  
 works that perform multi/cross-modal learning in some diverse  
 applications. Finally, we also review recent works addressing  
 the problem of caption generation for images, which is closely  
 related to the task of describing images using cross-modal cap-  
 tion retrieval.

**Image-Text Retrieval:** The problems of image and text re-  
 105 trieval are well-studied research topics [17, 18, 28, 29, 30].  
 A large number of existing approaches are based on retrieval  
 of unimodal data; i.e., both query as well as retrieved data be-  
 long to the same modality (e.g., either image [28] or text [29]).  
 Another approach that is popular among web-based search en-  
 gines is to use textual meta-data associated with images dur-  
 ing retrieval. Given a textual query, it is directly matched with  
 this meta-data instead of looking at the corresponding image.  
 However, such images constitute only a small portion of the  
 enormous amount of images available on the Internet, most of  
 which are without such meta-data. This limitation has led to  
 a growing interest in the problem of automatic image annota-  
 tion [1, 2, 3, 4, 5, 6, 7, 31, 32, 33]. Such models can sup-  
 port label-based queries during image retrieval without assum-  
 ing availability of any associated textual meta-data. Among  
 these, perhaps WSABIE [6] is the only method that has been  
 applied for web-scale annotation task. Another recent work [7]  
 demonstrates the applicability of Canonical Correlation Analy-  
 sis (CCA) for image annotation and retrieval on large datasets  
 (containing few hundred thousands of samples).

In parallel, there have also been several advances in the area  
 of multi-modal retrieval problems [34, 17, 35, 36], where re-  
 trieval is performed based on multiple modalities. These are  
 based on either learning a separate model for each modality and  
 then combining their predictions, or combining features from  
 different modalities and then learning a single model over them.

However, these approaches require data from all the modalities during the prediction phase. Moreover, some of them make use of multi-modal queries [34], making these somewhat difficult for large scale retrieval tasks.

In the recent years, cross-modal matching and retrieval have been actively studied [7, 16, 19, 21, 25, 37, 38, 39, 40, 41]. Among these, the CCA algorithm [20] is one of the most popular methods. It learns a latent projection space where the correlations between paired features from two modalities are maximized. In this space, samples from different modalities are matched using some simple nearest-neighbour based technique. Inspired from its simplicity and efficiency, several approaches have been proposed that perform cross-modal matching based on CCA [16, 37, 39]. While in [16, 20, 39], CCA is used to perform cross-modal retrieval of images and their associated descriptions, [37] uses it to learn associations between images and tags. Other than the CCA, methods such as Partial Least Squares (PLS) [42] and Bilinear Model (BLM) [43] have been proposed for cross-modal problems. There has also been some work on using deep neural networks for learning associations between images and texts [44, 45]. Note that most of the above mentioned approaches make use of two modalities in learning the latent space for cross-modal matching. However, in some cases, additional information is also available in the form of category labels (third modality/view). To make use of this, there have been some recent attempts in learning the latent embedding space using multi-view data [7, 21, 38, 39].

In summary, most of the existing cross-modal search algorithms try to learn a latent space that captures the intrinsic correlations present in the data. This latent space provides a homogeneous representation for samples from diverse modalities, which in turn allows direct cross-modal matching. As we will see in the following sections, our framework can be easily integrated with such representations, though with an increased computational load.

**Multi-modal Representations:** In addition to cross-modal matching of natural scene images and text, there have also been attempts in other domains that focus on dealing with diverse multi-modal representations. Some of the examples include scene text understanding [46], multi-modal clustering [47], modeling pairwise relations [48] and multi-modal image annotation [49, 50]. In [46], images of scene text and text-strings are first embedded into a vector space, and then a compatibility function is learned that allows to perform both image retrieval as well as recognition. Since a large portion of images on the web are associated with noisy and/or sparse meta-data (e.g., text, GPS coordinates camera specifications, etc.), a constrained multi-modal clustering approach was proposed in [47]. In [48], relational meta-data in the form of social connections was harnessed to model pairwise relations between images. Two recent papers [49, 50] demonstrated the utility of additional metadata (such as user-generated tags [49] and label relations based on WordNet taxonomy [50]) in boosting image annotation performance. Similar to these approaches, our interest is in learning higher level semantics using diverse modalities. However, we will concentrate on the task of cross-modal retrieval, and

demonstrate the applicability of our approach considering images and text as the two modalities.

**Image Caption Generation:** In parallel, there have been several attempts in the last few years that use short captions to describe images [9, 10, 11, 12, 13, 14, 15, 51]. Most of these works first try to predict the visual content of an image using some off-the-shelf computer vision technique (such as pre-trained object detectors and/or scene classifiers [12, 13], feature-based similarity with database images [10, 11], or both [9, 14]). This information is then fused using some Natural Language Generation (NLG) technique to construct image descriptions. All these works have shown that though *generating* captions provides a much larger set of possible descriptions, most usually failed to match descriptions generated or provided by humans. One primary reason for this is the limitations of NLG, which is still an emerging field. Few other works [14, 15] try to partly address this by directly transferring existing (human-written) captions to new images, by matching query image with annotated images. However, these approaches are primarily multi-modal, since they make use of both the modalities (image and text) during the testing phase. As we will show in our experiments, even without image-to-image matching, or using strong visual cues obtained from pre-trained object detectors/classifiers, cross-modal retrieval approaches, such as ours, can provide competitive performance compared to methods like [10, 14] that do make use of these.

Lately, there have been significant advances in the image captioning task, with most of the approaches focusing on deep neural network based models; e.g., [52, 53, 54, 55, 56]. These can be broadly categorized into two approaches. The first approach takes the activations from last hidden layer of an object detection convolutional neural network (CNN) model and feeds them into a recurrent neural network (RNN) language model, also referred to as a multi-modal RNN (MRNN) [53, 54, 55]. The second approach is based on first predicting a bag of words using a convolutional neural network (CNN) model that are likely to depict the visual content, and then using a maximum entropy language model over the predicted words for caption generation [56]. As discussed in [57], one limitation of state-of-the-art caption generation methods like above is that they reproduce generic caption from training data quite often, and do not perform well on images that are compositionally very different from previously seen images. To address this, a large-scale dataset with region-to-phrase correspondence for image descriptions was introduced in [52]. Such an explicit correspondence is expected to provide better supervision that would help in developing richer models for a variety of image-text compositions. These works are related to ours as they also model semantic associations between image and textual content. However, rather than using a neural network based model as in [53, 54, 55, 56], we model this association using a novel Structural SVM based approach, which provides a new perspective on this task. In experiments, while we make use of domain-specific representations, our approach is generic like existing cross-modal retrieval methods [20, 19, 7, 39], and can easily be applied to cross-modal retrieval tasks in diverse domains.

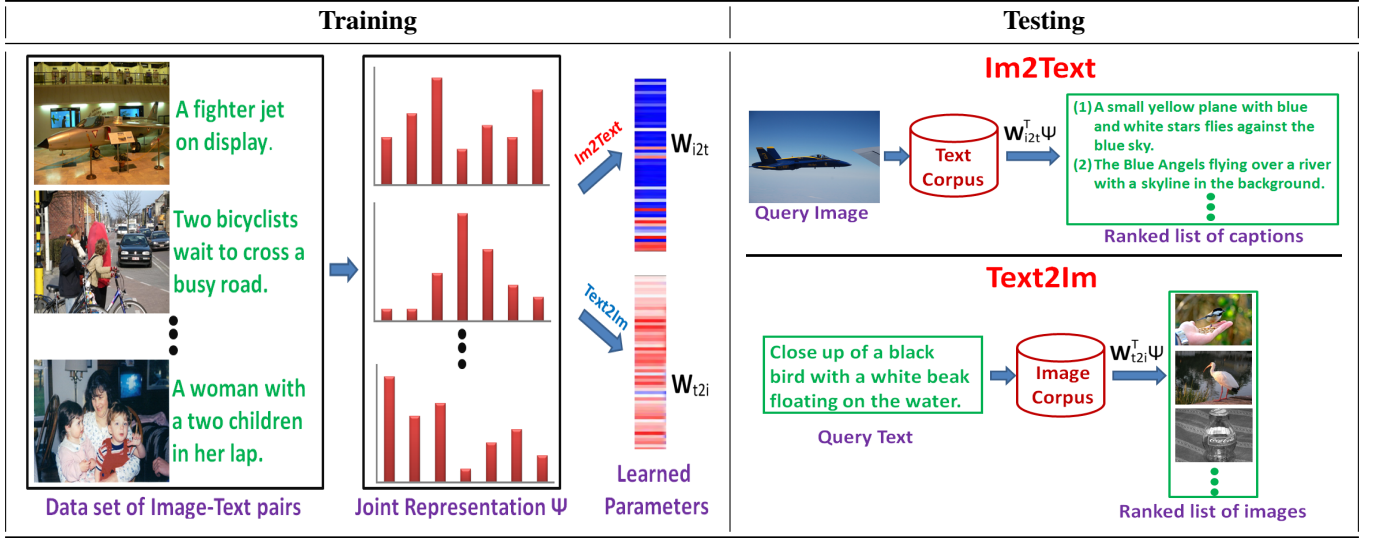


Figure 2: While training, given a dataset consisting of pairs of images and corresponding texts (here captions), we learn models for the two tasks (Im2Text and Text2Im) using a joint image-text representation. While testing for Im2Text, given a query image, we perform retrieval on a collection of only textual samples using the learned model. Similarly, for Text2Im, given a query text, retrieval is performed on a database consisting only of images.

### 3. Bilateral Image-Text Retrieval

In the conventional classification task, the goal is to assign a category from a finite set of discrete categories to a given (test) sample. A popular approach to do this is by training a category specific max-margin classifier using one-vs.-rest (or multi-class) Support Vector Machine (SVM) [58]. However, this becomes prohibitive when (1) the number of categories is exponentially large, and (2) the categories encode higher-level structure rather than being just simple labels. To overcome these, Structural SVM was introduced in [22]. Structural SVM is an oracle framework that can be adapted for a variety of tasks like object detection, classification with taxonomies, label sequence learning, etc. by appropriately defining its components that suit the problem at hand. In this paper, we make an initial attempt to address the problem of cross-modal multimedia search using Structural SVM. As per our knowledge, almost all the existing methods for cross-modal search are based on nearest-neighbour based similarity matching (in a learned homogeneous latent space). As we will show, Structural SVM naturally suits this task, where both input as well as output modalities can be quite complex in general (image $\leftrightarrow$ text in our case), and may have inherent structure in them. Moreover, availability of efficient algorithms for Structural SVM training (e.g., the cutting-plane algorithm [22]) make it scalable to large scale datasets.

#### 3.1. Approach

Here we present our framework for cross-modal search. During the training phase, we learn the associations between images and texts based on a joint representation. During the testing phase, we use the learned model to perform cross-modal search. Figure 2 illustrates our framework. As the proposed approach performs two complementary tasks (Im2Text and Text2Im), we will refer to it as *Bilateral Image-Text Retrieval (BITR)*.

First, we consider the task of retrieving semantically relevant text(s) given a query image (i.e., Im2Text). In Section. 3.4, we will discuss how the same framework is applicable for Text2Im as well. Let  $\mathcal{D} = \{(I_1, T_1), \dots, (I_N, T_N)\}$  be a collection of  $N$  images and corresponding texts. Each image  $I_i$  is represented using a  $p$ -dimensional feature vector  $\mathbf{x}_i$  in space  $\mathcal{X} = \mathbb{R}^p$ . Similarly, each text  $T_i$  is represented using a  $q$ -dimensional feature vector  $\mathbf{y}_i$  in space  $\mathcal{Y} = \mathbb{R}^q$ . We consider the problem of learning functions  $f : \mathcal{X} \rightarrow \mathcal{Y}$  using the input-output pairs  $\{(\mathbf{x}_i, \mathbf{y}_i)\} \in \mathcal{X} \times \mathcal{Y}$ . Similar to the Structural SVM framework [22], our objective is to learn a discriminant function  $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$  that can be used to predict the optimal output  $\mathbf{y}^*$  given an input  $\mathbf{x}$  by maximizing  $F$  over the space  $\mathcal{Y}$ ; i.e.,

$$\mathbf{y}^* = f(\mathbf{x}; \mathbf{w}) = \underset{\mathbf{y} \in \mathcal{Y}}{\operatorname{argmax}} F(\mathbf{x}, \mathbf{y}; \mathbf{w}) \quad (1)$$

where  $\mathbf{w}$  is the parameter vector that needs to be learned. We make the standard assumption of  $F = \mathbf{w} \cdot \Psi(\mathbf{x}, \mathbf{y})$ ; i.e.,  $F$  is a linear function of the joint feature representation  $\Psi(\cdot)$  of the input-output pair. In the above setting, our goal is to learn  $\mathbf{w}$  such that the maximum number of the following constraints are satisfied:

$$\forall i : \{ \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}_i) > \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) \} \quad \forall \mathbf{y} \in \mathcal{Y} \setminus \mathbf{y}_i \quad (2)$$

The above set of constraints signifies that for every sample  $\mathbf{x}_i$ , the parameter vector  $\mathbf{w}$  should be learned such that the prediction score for the true output (i.e.,  $F(\mathbf{x}_i, \mathbf{y}_i; \mathbf{w})$ ) remains higher than that for any other output. Since this is a hard problem, its solution is approximated by introducing non-negative slack variables. The task of learning  $\mathbf{w}$  is then formulated as the fol-

lowing optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, \xi \geq 0} \quad & \frac{1}{2} \|\mathbf{w}\|_2^2 + \frac{C}{N} \sum_{i=1}^N \xi_i \\ \text{s.t.} \quad & \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}_i) \geq \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) + \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i \\ & \forall i, \mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\} \end{aligned} \quad (3)$$

where  $\|\cdot\|_2^2$  denotes squared  $L_2$ -norm,  $C > 0$  is a constant that controls the trade-off between the regularization term and the loss term,  $\xi_i$  denotes the slack variable, and  $\Delta(\mathbf{y}_i, \mathbf{y})$  denotes the loss function that acts as a margin for penalizing any prediction other than the true output.<sup>1</sup> In the above optimization problem, the joint representation  $\Psi(\mathbf{x}, \mathbf{y})$  and the loss function  $\Delta(\mathbf{y}_i, \mathbf{y})$  are problem specific functions that need to be defined based on the given task.

### 3.2. Details

Now we describe the different components of our approach (i.e., the joint representation and the loss function), and how to efficiently solve the optimization problem in Eq. 3 for learning the parameter vector  $\mathbf{w}$ .

#### 3.2.1. Joint Image-Text Representation

The purpose of  $\Psi(\mathbf{x}, \mathbf{y})$  is to provide a joint representation for input and output data depending upon their individual representations. In cross-modal search (and in general), one popular way of representing a sample is in the form of a feature vector. This feature vector is computed based on domain knowledge of the modality under consideration. Each dimension of the feature vector carries some information that is specific to a given sample, and thus helps in distinguishing it from other samples within that modality. Another well-known practice is to normalize a feature vector before using it (e.g., using either  $L_1$  or  $L_2$  normalization), and is commonly adopted by almost all the practical systems including cross-modal search techniques such as [19].

Now let us consider an image-text pair  $(I, T)$ , where  $I$  is represented using a feature vector  $\mathbf{x} \in \mathcal{X}$  and  $T$  using another feature vector  $\mathbf{y} \in \mathcal{Y}$ , both of which are appropriately normalized. Since these two feature vectors are computed using different techniques and can have different dimensionality (i.e.,  $p$  need not be equal to  $q$ ), direct comparison between the two may be impractical. However as mentioned above, each dimension of a feature vector carries some information that is specific to the sample it represents. Hence, one feasible choice to learn correspondence between  $\mathbf{x}$  and  $\mathbf{y}$  is by considering all possible pairs of their individual elements. Intuitively, this will capture ‘‘cross-interactions’’ between the elements of the two vectors. When we learn a weight vector ( $\mathbf{w}$ ) over these pairs, each entry in this weight vector would denote the significance/degree of interaction between the corresponding cross-modal feature-element pair.

Thus we propose to use the joint representation constructed from the input-output representations  $\mathbf{x}$  and  $\mathbf{y}$  using their tensor product. That is, each dimension of  $\mathbf{x}$  is multiplicatively combined with every dimension of  $\mathbf{y}$  to get

$$\Psi(\mathbf{x}, \mathbf{y}) = \mathbf{x} \otimes \mathbf{y} \in \mathbb{R}^r, \quad (4)$$

where  $r = p \times q$ . This representation has the apparent advantage of not only efficiently capturing linear interactions between the input and output modalities but also providing computational benefits during inference, as we will discuss in Section 5.2.

#### 3.2.2. Loss Function

The function  $\Delta(\mathbf{y}_i, \mathbf{y})$  in Eq. 3 is a problem specific loss function. It acts as a margin in the Structural SVM framework, and is used to penalize incorrect predictions against the true output. Given an input-output pair  $(\mathbf{x}_i, \mathbf{y}_i)$  and any other prediction  $\mathbf{y}$ , the function is defined such that its value depends on the degree of dissimilarity between  $\mathbf{y}_i$  and  $\mathbf{y}$ . That is, if  $\mathbf{y}_i$  and  $\mathbf{y}$  are dissimilar, the value of  $\Delta(\mathbf{y}_i, \mathbf{y})$  should be high and vice-versa.

Projecting/mapping the samples in the output data ( $T_i$ s) to a vector space  $\mathcal{Y}$  allows us to adopt a suitable distance/similarity metric defined in vector space as our choice of loss function. Though this mapping can be highly non-linear in nature, the assumption here is that the projected space keeps the semantic proximity of the data intact; i.e., data points that are semantically similar are closer to each other in the projected vector space, than the data points that are semantically dissimilar to each other.<sup>2</sup> Based on this intuition, we define three different loss functions that are based on popular distance/similarity metrics: Manhattan distance  $\Delta_M(\cdot)$ , squared Euclidean distance  $\Delta_E(\cdot)$ , and normalized correlation (or cosine similarity)  $\Delta_C(\cdot)$ . These loss functions are given by:

$$\Delta_M(\mathbf{y}_i, \mathbf{y}) = \|\mathbf{y}_i - \mathbf{y}\|_1, \quad (5)$$

$$\Delta_E(\mathbf{y}_i, \mathbf{y}) = \|\mathbf{y}_i - \mathbf{y}\|_2^2, \quad (6)$$

$$\Delta_C(\mathbf{y}_i, \mathbf{y}) = 1 - \mathbf{y}_i \cdot \mathbf{y}, \quad (7)$$

where  $\|\cdot\|_1$  denotes  $L_1$ -norm. Since both  $\Delta_M(\cdot)$  and  $\Delta_E(\cdot)$  are distance metrics, they satisfy the properties of a valid loss function [22]; i.e.,  $\Delta_Z(\mathbf{y}_i, \mathbf{y}_i) = 0$ ,  $\Delta_Z(\mathbf{y}_i, \mathbf{y}_j) \geq 0$  for  $i \neq j$ , and  $\Delta_Z(\mathbf{y}_i, \mathbf{y}_j) \geq \Delta_Z(\mathbf{y}_i, \mathbf{y}_i)$  for  $i \neq j$  (where  $Z \in \{M, E\}$ ). Under the assumption that both  $\mathbf{y}_i$  and  $\mathbf{y}$  are  $L_2$ -normalized,  $\Delta_C(\cdot)$  also satisfies these properties and thus is a valid loss function. The efficient evaluation of these loss functions helps in a faster computation of the most violated constraint, which is required while solving the optimization problem in Eq. 3.

#### 3.2.3. Finding the Most Violated Constraint

Since the number of constraints in Eq. 2 can be exponentially large, it could be practically infeasible to make even a single pass over all the constraints during optimization.<sup>3</sup> Hence

<sup>1</sup>In [22], two formulations are presented for Structural SVM training. These are based on ‘margin-rescaling’ and ‘slack-rescaling’. We adopt the margin-rescaling one, which uses different margins for different possible outputs based on their similarity with the true output.

<sup>2</sup>This is a fundamental assumption that is usually at the heart of some machine learning algorithms.

<sup>3</sup>Potentially infinite in our case, since  $\mathcal{Y}$  is a continuous real-valued vector space.

it becomes crucial to efficiently find a small set of active constraints that would ensure a sufficiently accurate solution. This is achieved using the cutting-plane algorithm proposed in [22]. As shown in [22], this algorithm finds a solution that is close to optimal. Rather than considering all the constraints corresponding to a given pair  $\{\mathbf{x}_i, \mathbf{y}_i\}$ , it aims at finding the constraint that is violated the most, also called the most violated constraint. This in turn reduces the solution space by creating a nested sequence of tighter relaxations of the original problem.

Given an input-output pair  $(\mathbf{x}_i, \mathbf{y}_i)$ , the most violated constraint is the constraint corresponding to the incorrect output  $\hat{\mathbf{y}}$  predicted with the maximum score using the current learned parameter vector  $\mathbf{w}$ . It is given by:

$$\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\}} \Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) - \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}_i) \quad (8)$$

Since the last term is constant with respect to  $\mathbf{y}$ , this can be re-written as:

$$\hat{\mathbf{y}} = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\}} \Delta(\mathbf{y}_i, \mathbf{y}) + \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) \quad (9)$$

For the three loss functions in Eq. 5, 6, and 7, this maps to the following problems respectively:

$$\hat{\mathbf{y}}_M = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\}} \|\mathbf{y}_i - \mathbf{y}\|_1 + \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) \quad (10)$$

$$\hat{\mathbf{y}}_E = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\}} \|\mathbf{y}_i - \mathbf{y}\|_2^2 + \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) \quad (11)$$

$$\hat{\mathbf{y}}_C = \operatorname{argmax}_{\mathbf{y} \in \mathcal{Y} \setminus \{\mathbf{y}_i\}} 1 - \mathbf{y}_i \cdot \mathbf{y} + \mathbf{w} \cdot \Psi(\mathbf{x}_i, \mathbf{y}) \quad (12)$$

It can be easily verified that each of the above three equations corresponds to maximizing a convex function. In practice, since every feature vector is normalized, each of its elements remains bounded within a range. This allows us to solve the above problems efficiently using an iterative gradient-ascent method. After each iteration of gradient-ascent, the current output is projected depending on the particular type of normalization considered. More details on this can be found in our publicly available implementation.<sup>4</sup>

### 3.3. Inference: Retrieving a Ranked List of Output

Consider an independent database  $\mathcal{T}' = \{T'_1, \dots, T'_{|\mathcal{T}'|}\}$  consisting of only textual samples, where each  $T'_k$  is represented using a feature vector  $\mathbf{y}'_k \in \mathcal{Y}$ . Given a query image  $J$  represented by  $\mathbf{x} \in \mathcal{X}$ , Im2Text requires ranking the elements of  $\mathcal{T}'$  according to their relevance with  $J$  using the learned parameter vector  $\mathbf{w}$ . This can be performed by sorting the elements of  $\mathcal{T}'$  based on the score  $F(\mathbf{x}, \mathbf{y}'_k; \mathbf{w}) = \mathbf{w} \cdot \Psi(\mathbf{x}, \mathbf{y}'_k)$ ,  $\forall k \in \{1, \dots, |\mathcal{T}'|\}$  (where higher score means more relevance and vice-versa), thus allowing to retrieve a ranked list of texts.

### 3.4. Performing “Text2Im”

Now we consider the task of retrieving semantically relevant image(s) given a query text (i.e., Text2Im). Similar to Im2Text, we are given a collection  $\mathcal{D} = \{(I_1, T_1), \dots, (I_N, T_N)\}$  of images and corresponding texts. Each image  $I_i$  is represented using a  $p$ -dimensional feature vector  $\mathbf{x}_i$  in space  $\mathcal{X} = \mathbb{R}^p$ , and each text  $T_i$  is represented using a  $q$ -dimensional feature vector  $\mathbf{y}_i$  in space  $\mathcal{Y} = \mathbb{R}^q$ . Our objective now becomes to learn a discriminant function  $F : \mathcal{Y} \times \mathcal{X} \rightarrow \mathbb{R}$  that can be used to predict the optimal output (image)  $\mathbf{x}^*$  given an input (text)  $\mathbf{y}$  by maximizing  $F$  over the space  $\mathcal{X}$ . That is,

$$\mathbf{x}^* = f(\mathbf{y}; \mathbf{w}) = \operatorname{argmax}_{\mathbf{x} \in \mathcal{X}} F(\mathbf{y}, \mathbf{x}; \mathbf{w}), \quad (13)$$

where  $\mathbf{w}$  is the parameter vector that needs to be learned, and  $F = \mathbf{w} \cdot \Psi(\mathbf{y}, \mathbf{x})$ . Since we make no specific assumption for the particular representations used for visual and textual data (except that they are represented in the form of feature vectors), the joint representation and loss functions defined above for Im2Text will remain equally applicable for Text2Im as well. Hence, in order to perform Text2Im, we can adopt the same methodology as that for Im2Text. However, note that here since we are dealing with a different (inverse) problem, we will learn a separate model ( $\mathbf{w}$ ).

## 4. Comparison with Some Previous Approaches

As discussed in Section 2, CCA [20, 19] and WSABIE [6] are two well-known methods that can scale to large datasets and have been shown to work well for learning cross-modal associations. Here we present a comparison of these two with the proposed approach.

### 4.1. Comparison with CCA

CCA can be shown to minimize the squared Euclidean distance between pairs of samples from two modalities in the projected space [20, 59]. Let  $\mathbf{U}$  and  $\mathbf{V}$  denote the two projection matrices and  $\mathbf{a}$  and  $\mathbf{b}$  denote a pair of samples from the two modalities respectively. Thus, CCA can be seen to match the samples using the similarity function  $\exp(-\|\mathbf{U}\mathbf{a} - \mathbf{V}\mathbf{b}\|_2^2) = \exp(\mathbf{a}^t (\mathbf{U}^t \mathbf{V}) \mathbf{b})$ . This maps to minimizing the loss  $l(1, z) = -\log(z)$  during training. We can observe that both CCA as well as BITR rely on bilateral scoring functions. An important difference is that while CCA makes use of only similar pairs of samples across modalities, BITR explicitly models the dissimilar pairs and pushes them apart. However, as we will discuss in the next section, this in turn makes the training of BITR much slower than CCA. Second, while CCA decouples the two projection matrices and constraints each to be low rank, BITR learns a joint full rank parameter vector  $\mathbf{w}$  and makes use of  $L_2$  regularization to avoid overfitting. Third, as discussed above, our formulation can work with a variety of loss functions that suit the cross-modal retrieval task.

<sup>4</sup><http://researchweb.iit.ac.in/~yashaswi.verma/crossmodal/bittr.zip>



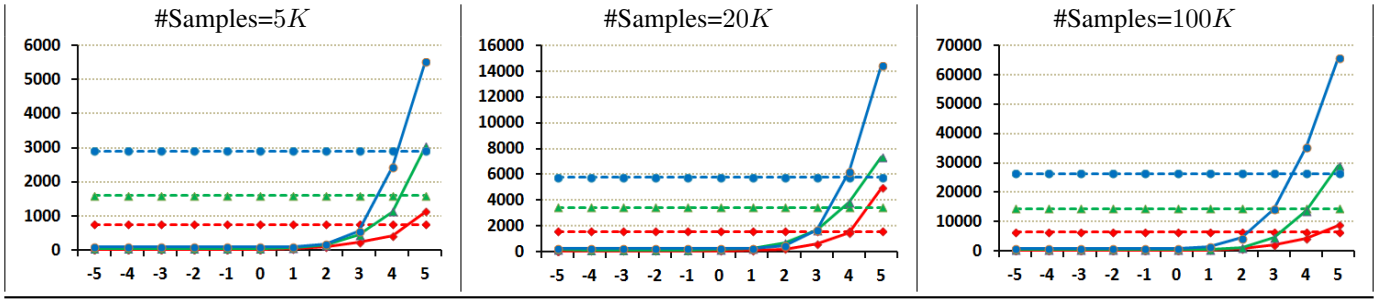


Figure 3: Comparison of the training time using WSABIE and BITR. The horizontal axis denotes the value of the  $C$  parameter (power of 10), and the vertical axis denotes the training time in seconds. Dashed lines correspond to WSABIE and solid lines correspond to BITR. Each colour denotes the dimensionality of feature vector (same for both the modalities): {red, green, blue} map to {50, 100, 150} in that order.

#### 4.2. Comparison with Wsabie

WSABIE was originally proposed for the task of label-ranking, and hence can not be directly applied to captions. For our comparisons, we thus modify the WSABIE algorithm, such that instead of learning a separate parameter vector for each label, it learns a single parameter matrix for all the captions. This is analogous to the parameter matrix learned for visual features in the WSABIE algorithm (details are provided in the Appendix). Similar to CCA and BITR, WSABIE also relies on a bilateral scoring function. However, unlike BITR and analogous to CCA, WSABIE decouples the projection matrices for the two modalities, and constraints their individual norms without performing an explicit regularization. Second, during optimization, WSABIE considers any random (negative) sample that violates the margin condition to update the model, whereas BITR picks the sample corresponding to the most violated constraint (Eq. 8). This makes the training of WSABIE more scalable than BITR, however the model learned using BITR is more accurate than that using WSABIE (as also validated in the experimental analysis).

### 5. Training time and Run-time Analysis

Here we will analyze the training and run-time efficiency of the proposed approach.<sup>5</sup> We will consider the task of Im2Text, with similar reasoning being applicable to Text2Im as well.

#### 5.1. Training time analysis

In Figure 3, we compare the training time of WSABIE [6] and BITR using synthetic features. Following [7], we use early stopping for WSABIE after iterating for 20 passes of training samples. Here we do not show the training time of CCA [20, 19] because its standard implementations are quite efficient, and it took less than 1 second to learn the projection matrices in the below mentioned set-up. For the comparison, we vary the number of training samples in {5K, 20K, 100K} and the dimensionality of image/text features in {50, 100, 150}.

<sup>5</sup>Using our Matlab implementation on a 2.4 GHz Intel Xeon (E5-2600) processor with 48 GB of RAM.

In the figure, the horizontal axis denotes the value of the  $C$  parameter (power of 10), and the vertical axis denotes the training time in seconds. From the figure, we observe that the training time of BITR is under 15 minutes even for 100K samples when  $C$  is small. However, on increasing  $C$  beyond 10, there is a steep rise in the training time. This is expected because on increasing  $C$ , the algorithm tries to better fit the model to the training data. For example, using 100K samples and 150 dimensional image and text features (joint representation of 22500 dimensions), with  $C = 10^{-5}$  it takes just around 15 minutes to train the model, whereas with  $C = 10^5$  it takes around 18 hours. This analysis demonstrates even though the training time of BITR can be quite high for large values of  $C$ , it is still feasible and thus easily scalable to large datasets.

#### 5.2. Run-time analysis

It is interesting to note that in order to evaluate the function  $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$ , we do not require to explicitly compute the joint representation  $\Psi(\mathbf{x}, \mathbf{y})$ . Since  $\Psi(\mathbf{x}, \mathbf{y})$  is a tensor product of the vectors  $\mathbf{x} \in \mathbb{R}^p$  and  $\mathbf{y} \in \mathbb{R}^q$ , it is a vector of products of pairs of elements from  $\mathbf{x}$  and  $\mathbf{y}$ :

$$\Psi(\mathbf{x}, \mathbf{y}) = [\mathbf{x}(1)\mathbf{y}(1), \dots, \mathbf{x}(p)\mathbf{y}(1), \dots, \mathbf{x}(p)\mathbf{y}(q)]^t \in \mathbb{R}^r$$

where the superscript  $t$  denotes vector transpose. Since  $\mathbf{w}$  is also a vector in  $\mathbb{R}^r$ , it can be re-written in matrix form:

$$\mathbf{W} = [\mathbf{w}_1 \mathbf{w}_2 \dots \mathbf{w}_q] \in \mathbb{R}^{p \times q},$$

where each  $\mathbf{w}_k \in \mathbb{R}^p$  denotes the consecutive  $p$  elements of  $\mathbf{w}$  in the  $k^{th}$  interval. Using the above, it is easy to verify that the function  $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$  can be re-written as:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{x}^t \mathbf{W} \mathbf{y} \quad (14)$$

Rather than evaluating the function  $F(\mathbf{x}, \mathbf{y}_k; \mathbf{w})$  individually for each sample in the retrieval set  $\mathcal{T}'$ , the above transformation allows to evaluate it for a batch of samples in  $\mathcal{T}'$  in a single pass. Here we will illustrate this by computing it for all the samples in  $\mathcal{T}'$  in a single pass. Let  $\mathbf{Y} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_{|\mathcal{T}'|}] \in \mathbb{R}^{q \times |\mathcal{T}'|}$  denote the matrix formed by concatenating the feature representations of all the samples in  $\mathcal{T}'$ . For a given (image)

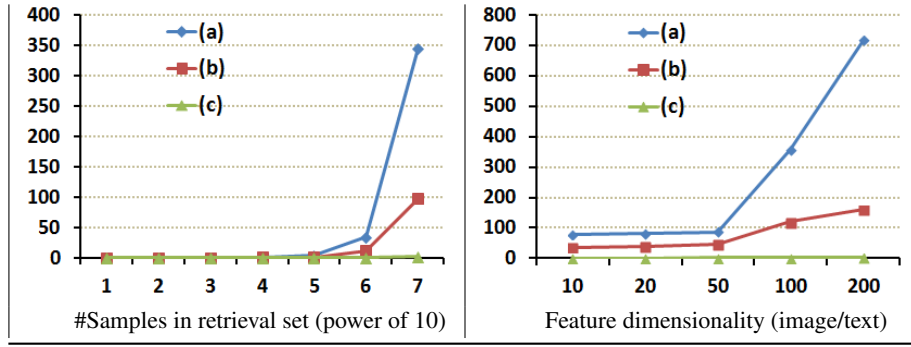


Figure 4: Comparison of time required (in seconds on vertical axis) for ranking the samples in a retrieval set  $\mathcal{T}'$  for a single query, when the prediction score is computed (a) individually for each sample after computing the joint representation, (b) individually for each sample without computing the joint representation, and (c) jointly for all the samples without computing the joint representation. **Left:** On varying the size of the retrieval set by keeping feature dimensionality of both visual and textual features to be 100 ( $p = q = 100$ ). **Right:** On varying the feature dimensionality (same for image/text samples) for a retrieval set containing  $10^7$  samples.

query  $J$  represented by feature vector  $\mathbf{x}$ , let  $\mathbf{s} \in \mathbb{R}^{|\mathcal{T}'|}$  be a vector such that its  $k^{th}$  element denotes the prediction score corresponding to the  $k^{th}$  sample in  $\mathcal{T}'$ . Then it can be computed as:

$$\mathbf{s} = (\mathbf{x}^t \mathbf{W} \mathbf{Y})^t. \quad (15)$$

After computing this, the ranking follows by sorting the elements of  $\mathcal{T}'$  based on their corresponding scores in  $\mathbf{s}$  in descending order. In popular matrix multiplication softwares (such as Matlab), the joint computation of similarity scores for a batch of samples can be much faster than computing them individually. This in turn provides significant boost in the run-time efficiency.

Assuming the features are already computed, Figure 4 (left) compares the relative time required for ranking the samples in a synthetic retrieval set  $\mathcal{T}'$  for a single query. In cross-modal search scenarios, the samples from both the modalities are usually represented using feature vectors containing a few tens or hundreds of elements [19]. Keeping this in mind, we keep  $p = q = 100$  (recall that the dimensionality of the joint feature representation is  $r = p \times q$ , which is  $10^4$  in this case), and vary the number of samples in  $\mathcal{T}'$  in  $\{10^1, 10^2, \dots, 10^7\}$ . We consider three situations, when the prediction score is computed (a) one sample at a time by first computing the joint representation, (b) one sample at a time without computing the joint representation (Eq. 14), and (c) jointly for all the samples without computing the joint representation (Eq. 15). From the figure, we observe that for all these three, the total time (including similarity score computation and sorting) increases almost linearly with the number of samples. However, even with linear increment, the total time required for (c) is significantly lower than that for (a) and (b). For example, when the retrieval set has ten million samples, the time taken when using (a), (b) and (c) are around 345.8, 97.6, and 1.7 seconds respectively. For all three, around 1.2 seconds are taken in sorting the samples based on their scores. If we do not consider this, then (c) takes just around 0.5 seconds in computing the prediction scores for all the samples, which is faster than the time required for the sorting operation.

In Figure 4 (right), we compare the relative time required for ranking the samples in a synthetic retrieval set  $\mathcal{T}'$  containing  $10^7$  samples for a single query. Here we vary the dimensionality of input and output modalities (same for both  $p$  and  $q$ ) as  $\{10, 20, 50, 100, 200\}$ . These result into joint feature representations of dimensions  $\{100, 400, 2.5K, 10K, 40K\}$  respectively. We consider the three situations (a), (b), and (c) as mentioned above. Here we observe that for all these three cases, the total time increases with feature dimensionality. However, in this case, the increments are not simply linear. For lower dimensions, they are sub-linear, while for higher dimensions, they are super-linear. For both (a) and (b), the total time taken is not practically appealing even for lower dimensional features. E.g., these are around 78.2 and 36.4 seconds for (a) and (b) respectively when  $p = q = 10$ . On the other hand, the total time for (c) using  $p = q = 10$  and  $p = q = 200$  are just around 1.3 and 2.3 seconds respectively. On discarding the time taken in sorting the elements after score computation (around 1.2 seconds), these become just around 0.1 and 1.1 seconds respectively.

From Figure 4, we can conclude that a direct (naïve) implementation could mar the efficiency of our approach during inference. However, using simple transformations that allow batch processing, it is possible to achieve significant speed-ups, thus making it feasible to perform retrieval on large datasets containing millions of samples.

In our experiments, we compare the BITR approach with two baseline methods: CCA [20, 19] and WSABIE [6]. Comparing the run-time of BITR with CCA and WSABIE, we can easily observe that for each of these methods, in practice we need to project the features in the retrieval set just once and this can be done off-line. Now given a query, we can rank the samples by simply taking their dot product. Hence, the run-time of all the three methods becomes equivalent.

## 6. Image and Text Representation

We consider different types of representations for visual and textual data. These representations are compact, yet known to



be effective in capturing data semantics. The first representation captures data characteristics in the form of probability distributions over unimodal topics. We refer to this as topic-based representation (TR). The second representation is based on learning cross-modal correlations between input and output modalities over TR. We refer to this as correlated topic-based representation (CTR). The third representation is based on modern CNN and word2vec features for images and text respectively.

It should be noted that since the complexity of learning a Structural SVM model depends on both number of training samples ( $N$ ) as well as dimensionality of the joint feature representation ( $r = p \times q$ ), in practice it is desirable to work with representations that are compact to maintain computational load. As discussed before, using compact representations for data is also practiced by other cross-modal retrieval techniques such as [19, 7, 39]. Hence we adapt the representations accordingly to satisfy this requirement.

### 6.1. Topic-based Representation

This representation is based on unimodal probability distributions over topics, that are learned using Latent Dirichlet Allocation (LDA) model [60]. LDA is a popular probabilistic generative topic model and can effectively capture complex semantics of data in a compact manner. It considers a given document as a collection of discrete units/words. Based on co-occurrences of these words, it discovers high-level topics, and represents these in the form of multinomial distributions over words. Given a new document, LDA represents it as a probability distribution over the previously learned topics.

#### 6.1.1. Representing Images

Since LDA requires each image to be represented as a collection of words, first we need to learn the visual words' vocabulary. For this, we randomly sample  $0.5M$  SIFT descriptors [61] extracted densely at multiple scales from the training images of the SBU dataset [14], and learn 1000 words using the k-means algorithm. Each image is then represented as a bag-of-words histogram of these visual words. From this histogram based representation, the visual topics are learned using LDA by considering 5000 random (training) images from the SBU dataset.

Now, given a new image, first we extract SIFT descriptors densely at multiple scales, and represent it as a bag-of-words histogram of visual words as before. This is then used by LDA to construct a representation in the form of a probability distribution over the topics learned earlier.

#### 6.1.2. Representing Text

1. *Representing Captions:* To learn textual topics, we use the captions in the training subset of the SBU dataset [14]. Using these, we get a vocabulary of around  $0.18M$  words, after simple pre-processing like removing stop-words. This vocabulary is used to represent the captions in the form of bag-of-words histograms, which are then used to learn textual topics using LDA. A new caption is represented as a bag-of-words histogram using

the above vocabulary, which is then used to obtain a representation in the form of a probability distribution over the learned topics.

2. *Representing Phrases:* Here we assume an annotated (training) dataset where each image is tagged with a set of phrases. We learn textual topics by considering each phrase as a discrete unit and then representing each phrase as a probability distribution over them similar to captions.
3. *Representing Labels:* Similar to the previous case, here we assume an annotated dataset of images tagged with a set of labels. While learning topics, each label is considered as a discrete unit. Once the topics are learned using LDA, each label is represented as a probability distribution over them.

### 6.2. Correlated Topic-based Representation

In this representation, we incorporate cross-modal correlations into the topic-based representations for visual and textual data analogous to [19]. This is done by mapping the data into a maximally correlated vector subspace, that is learned using CCA [20]. This is based on the assumption that the samples coming from two different modalities contain some joint information that can be encoded using correlations between them [20].

Note that while TR contains only non-negative (latent probability) values, CTR contains both positive as well as negative values. This is because it is obtained by projecting TR using a linear transformation learned through CCA, which projects an input vector into a maximally correlated real-valued vector space.

### 6.3. Modern Representations

In our conference paper [25], we had considered TR and CTR as the two different representations for images and text. However, lately features computed using CNN for images [26, 62] and word2vec for text [27] have been popularly used in several tasks that deal with visual and textual data. Hence, we also evaluate using these features on the cross-modal image-caption retrieval task in Section 7.7. In practice, we compute features for images using a CNN model pre-trained on the ImageNet dataset [26] for image classification, that was shown to perform well for other visual recognition tasks as well. For captions, we use the pre-trained model of [27] by taking average of vector representations of all the words in a caption.

## 7. Experiments

We demonstrate the applicability of our approach and extensively compare it with competing baseline methods on various cross-modal multimedia search tasks.

### 7.1. Datasets

We consider three popular datasets in our experiments:

Dataset	Samples	#Captions/Img.	Words/Caption
Pascal	1000	5	$9.82 \pm 3.51$
IAPRTC-12	19627	1	$24.98 \pm 10.67$
SBU	1M	1	$12.14 \pm 6.01$

Table 1: Statistics of the three datasets used in our experiments. The last column shows the average number of words per caption.

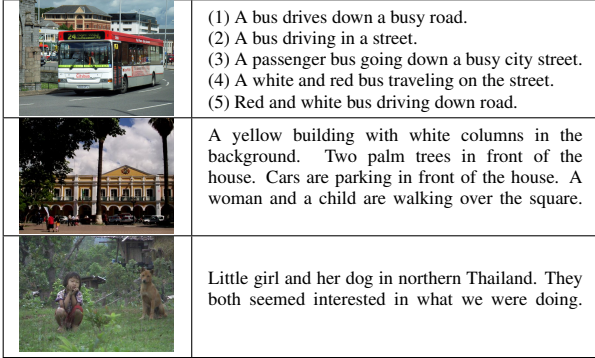


Figure 5: Sample images with ground-truth descriptions from Pascal (top), IAPR (middle) and SBU (bottom) datasets.

- **UIUC Pascal Sentence Dataset:** This was introduced in [23] and has become a de facto benchmark in the domain of image-caption understanding. It contains 1000 images, each of which is annotated with 5 captions from independent human-annotators.
- **IAPR TC-12 Benchmark:** This was introduced in [24] for the task of cross-language information retrieval. It has 19627 images, each of which is associated with a long description of up to 5 sentences.
- **SBU-Captioned Photo Dataset:** This was published in [14] and contains one million captioned images downloaded from Flickr. To our knowledge, this is the largest publicly available dataset of captioned photographs.

Table 1 shows general statistics of these datasets and Figure 5 shows example images along with their ground-truth descriptions. These datasets have been used by several approaches for image-to-caption generation [10, 12, 13, 51], and image-caption retrieval [14, 15, 25, 44]. For both Pascal and IAPR datasets, the captions/descriptions were written by guided human annotators. However, for the SBU dataset, the captions were written by the users who had uploaded those photographs on Flickr. Due to this, these captions are quite diverse and noisy. Moreover, they usually contain associated sentiments and abstract semantics that are not physically present in the image (e.g., see the third example in Figure 5). This makes the SBU dataset particularly challenging for cross-modal search task.

## 7.2. Evaluation Metrics

We adopt following evaluation metrics depending upon the form of textual data.

### 7.2.1. Captions

For captions, we consider two types of evaluation metrics that have been adopted by (1) image caption generation methods (such as [9, 10, 11, 12, 13]), and (2) image-caption retrieval methods (such as [16, 44]).

In the first setting, we consider BLEU [63] and Rouge [64] metrics for evaluation, that are popularly used for evaluating automatic summarization and machine translation approaches<sup>6</sup>. Here, the samples in the test set comprise the query set, and retrieval is performed on the full training set. For both Im2Text and Text2Im, we report mean one-gram BLEU and Rouge scores. For Im2Text, these scores are averaged over the top five retrieved captions, by matching them with the ground-truth caption of query image. For Text2Im, we compute these scores in an inverse manner; i.e., by matching the query caption with the ground-truth captions of the top five retrieved images. For both these metrics, a higher score means better performance.

In the second setting, we consider Recall@K (R@K) and MedianRank (MedR) as the metrics for evaluation. For a given query, these are used to evaluate how correctly an approach can retrieve the true output (image/caption), assuming it to be present in the retrieval set. For Im2Text, this is performed by considering the images in the test set as queries, and performing retrieval over the captions in the test set. Similarly, for Text2Im, this is done by querying the captions in the test set and performing retrieval over the images in the test set. Recall@K measures for what percentage of queries, their correct output is present in the top K (K=50 in our case) retrieved samples. MedianRank measures the median of the retrieval ranks of the correct outputs corresponding to all the queries. For Recall@K, higher score means better performance, and for MedianRank, lower score means better performance.<sup>7</sup>

### 7.2.2. Phrases and Labels

We adopt Precision@K (P@K) and mean Average Precision (mAP) for performance evaluation. For Im2Text, given a query image, we rank the phrases/labels and match them with the ground-truth of the query. For Text2Im, given a query phrase/label, we rank the images in the test set and evaluate based on the presence of the query in their ground-truth. For Im2Text, Precision@K measures the number of true labels that are predicted in the top-K retrieved labels (considering K=5 in our case). For Text2Im, it measures the number of top-K retrieved images that are tagged with query label in their ground-truth. For both Im2Text and Text2Im, mAP measures the mean of average precision for all the queries. For both these metrics, higher score means better performance.

## 7.3. Baselines for Comparisons

We compare our methods against two popular baselines: WSABIE [6] and CCA [20, 19] in all the experiments. Both

<sup>6</sup>To compute BLEU scores, we use the code released by NIST (version-13a). To compute Rouge scores, we use Release-1.5.5 obtained from <http://www.berouge.com/Pages/default.aspx>.

<sup>7</sup>Recall@K and MedianRank are the additional metrics that we consider here, which were not considered in [25].

CCA and WSABIE learn separate projection matrices for input and output data. In practice, they both may converge to a lower dimensional projection space compared to the dimensionality of the given data without really affecting the performance. However in all our experiments, we project data into the same space for both these methods. This not only avoids information loss but also allows fair comparisons and avoids the need of tuning the optimal number of projections required by each. For CCA, we use normalized correlation in order to compute nearest-neighbour based similarity between two projected cross-modal features, which was found to perform better than using other measures such as  $L_1$  or  $L_2$  distance in [19].

Along with CCA and WSABIE, we also consider weighted k-nearest neighbours (wKNN) algorithm (similar to [4]) and one-versus-rest SVM for additional comparisons in Experiment-3 and Experiment-4 while considering phrases and labels as textual data (respectively), as these methods are applicable in those settings and are popularly used as strong baselines in several retrieval-related tasks involving discrete categories.

#### 7.4. Implementation Details

- In all the experiments, each visual and textual sample is represented using a 100 dimensional feature vector. Note that while the CCA baseline [19, 20, 65] projects the samples from both modalities into a common space (whose dimensionality is at most the minimum of the dimensionality of the input feature spaces), BCTR does not require the features from both modalities to have the same dimensionality for cross-modal matching. However, we keep it same for fair comparisons. Also, while the training time complexity of CCA is cubic in feature dimensionality, that for BCTR is quadratic. Based on this, the chosen dimensionality was found to provide a good trade-off between efficiency and efficacy in preliminary experiments.
- For our approach, we report results using the three loss functions given in Eq. 5, 6, and 7, and will refer to them as BCTR-M, BCTR-E, and BCTR-C respectively.
- In all the experiments, the particular representation being employed will be denoted using “TR” or “CTR”.
- In all the experiments, the  $C$  parameter is tuned using five-fold cross-validation in the range  $\{10^{-5}, 10^{-4}, \dots, 10^4, 10^5\}$  for BCTR, WSABIE and SVM (Exp-3 and Exp-4).
- In the case of topic-based representation, the feature representations (separately for both the modalities) are  $L_1$  normalized while considering the loss function  $\Delta_M(\cdot)$  and  $L_2$  normalized while considering the other two loss functions. In the case of CTR, the projected representations are  $L_2$  normalized throughout. These normalization criteria are followed in all the experiments for all the evaluated methods. The choice of these normalizations is based on the practices that are popular while doing feature normalization.

#### 7.5. Retrieval Schemes

We consider the following retrieval schemes depending upon the form of textual data.

##### 7.5.1. Experiment-1: Image-Caption Retrieval

Here we consider textual data to be in the form of captions. We conduct this experiment on all the three datasets as described in Section 7.1.

(1) For the SBU dataset, we follow the train/test splits used in [14], which includes 500 test samples and 999.5K training samples. For all the compared approaches, the parameters are learned using a subset of 0.1 million samples randomly picked from the training data.

(2) For the other two datasets (IAPR and Pascal), we compute performance over all the samples as in [13, 10]. This is done by creating ten partitions of dataset. Each time, one partition is used for testing and the others for training. The final performance is computed by averaging the performance over all the splits.

##### 7.5.2. Experiment-2: Cross-dataset Image-Caption Retrieval

In this experiment, we analyze the generalization ability of different cross-modal search methods across datasets. For this, we consider textual data to be in the form of captions as in Experiment-1. However, instead of learning models for each dataset individually, we use the model learned using SBU dataset in Experiment-1 and evaluate the performance on IAPR and Pascal datasets. For computing BLEU and Rouge scores, we consider queries as all the images from Pascal or IAPR dataset, and perform retrieval on all the captions of SBU dataset for Im2Text. Similarly, for Text2Im, we consider queries as all the captions from Pascal or IAPR dataset, and perform retrieval on the full image collection of SBU dataset. For computing Recall@K and MedianRank, we use the model learned using SBU dataset, and perform retrieval over the samples in Pascal and IAPR datasets by partitioning them into ten splits as in Experiment-1 (for direct comparison with the results obtained in Experiment-1). The goal of this experiment is to study the effect of dataset specific biases in different methods, and as per our knowledge, this is the first such study in the cross-modal search domain. This experiment also demonstrates the applicability of different methods on retrieval using large query sets (1000 for Pascal and 19627 for IAPR) and retrieval set (all one million samples of the SBU dataset).

##### 7.5.3. Experiment-3: Image-Phrase Retrieval

Here we consider textual data to be in the form of phrases, and demonstrate results on IAPR dataset. These phrases are relation tuples that are automatically extracted from the available captions of this dataset. To extract these, the captions are processed using the Stanford CoreNLP toolkit [66]. As suggested in [67], “collapsed-ccprocessed dependencies” are used which are useful for the relation extraction task. In practice, we consider three types of phrases that cover the basic (visual/textual) aspects (i.e., *noun*, *preposition*, and *verb*) of an image/caption. These include phrases of the forms (*noun*, *verb*)

	Method	Im2Text				Text2Im			
		%BLEU-1↑	%Rouge-1↑	R@50↑	MedR↓	%BLEU-1↑	%Rouge-1↑	R@50↑	MedR↓
Pascal	CCA	<b>31.49</b>	13.97	<b>47.10</b>	<b>11.05</b>	<b>32.54</b>	14.32	<b>57.60</b>	<b>6.50</b>
	Wsabie (TR)	30.77	19.64	11.40	26.25	31.35	19.83	10.10	44.90
	Wsabie (CTR)	31.19	<b>21.72</b>	13.50	22.15	31.02	<b>21.54</b>	12.40	41.45
	BITR-M (TR)	31.51	22.34	41.20	14.05	32.40	22.01	10.90	45.35
	BITR-M (CTR)	32.04	23.74	42.10	15.35	33.01	24.16	55.00	7.45
	BITR-E (TR)	32.86	20.98	39.70	13.45	34.04	22.89	11.20	44.15
	BITR-E (CTR)	33.80	23.06	40.30	13.85	34.91	24.01	<b>57.20</b>	6.90
	BITR-C (TR)	32.67	22.75	46.50	11.30	33.73	23.15	10.60	44.55
	BITR-C (CTR)	<b>34.85</b>	<b>23.97</b>	<b>51.40</b>	<b>9.10</b>	<b>34.89</b>	<b>24.38</b>	56.80	<b>6.80</b>
IAPR	CCA	<b>29.46</b>	<b>30.31</b>	<b>16.32</b>	<b>404.35</b>	<b>30.50</b>	<b>30.16</b>	<b>18.53</b>	<b>301.45</b>
	Wsabie (TR)	26.70	23.75	2.73	932.10	26.01	24.16	2.73	999.10
	Wsabie (CTR)	28.13	24.97	4.76	772.30	27.54	27.00	3.79	798.95
	BITR-M (TR)	31.72	30.43	10.57	519.85	28.58	26.50	2.58	987.45
	BITR-M (CTR)	32.27	31.30	10.30	480.25	30.28	28.20	11.98	409.70
	BITR-E (TR)	31.67	30.47	10.68	493.30	28.74	26.46	2.63	979.45
	BITR-E (CTR)	33.91	32.40	12.41	416.85	30.90	29.01	<b>16.48</b>	<b>334.00</b>
	BITR-C (TR)	32.19	31.65	9.37	594.45	29.63	27.11	2.66	956.05
	BITR-C (CTR)	<b>34.18</b>	<b>32.81</b>	<b>13.78</b>	<b>335.95</b>	<b>31.49</b>	<b>29.66</b>	14.36	355.60
SBU	CCA	<b>13.91</b>	<b>11.47</b>	<b>16.20</b>	<b>189.50</b>	<b>14.53</b>	11.05	<b>19.80</b>	<b>190.00</b>
	Wsabie (TR)	7.74	6.64	8.40	254.50	13.94	11.59	10.60	246.50
	Wsabie (CTR)	12.50	10.43	11.60	237.00	14.15	<b>11.72</b>	11.80	232.00
	BITR-M (TR)	9.86	8.36	15.40	213.00	14.27	11.70	10.40	248.50
	BITR-M (CTR)	14.01	11.21	16.40	212.00	15.92	13.20	21.60	159.50
	BITR-E (TR)	10.08	8.58	13.20	209.00	16.62	<b>13.32</b>	10.60	249.50
	BITR-E (CTR)	14.28	11.38	19.20	195.00	14.94	11.61	21.40	181.50
	BITR-C (TR)	14.68	<b>11.82</b>	16.00	212.50	<b>17.90</b>	13.18	11.00	251.00
	BITR-C (CTR)	<b>15.19</b>	11.39	<b>24.60</b>	<b>144.50</b>	17.82	11.66	<b>25.20</b>	<b>149.00</b>

Table 2: Comparison of the performance using baseline methods (CCA [19] and Wsabie [6]) and variants of our method for image-caption retrieval (Experiment-1). The best results using both are highlighted in bold. (↑: higher means better; ↓: lower means better.)

(e.g., “person walk”), (*noun, preposition, noun*) (e.g., “person on road”) and (*verb, preposition, noun*) (e.g., “walk on road”).

For evaluation, we create ten partitions from the IAPR dataset. Similar to Experiment-1, we report averaged results over ten trials, each time considering one partition for testing and the rest for training. For Im2Text, given a query image, we rank the phrases. For Text2Im, given a query phrase, we rank the images in an analogous manner. It should be noted that since we consider each phrase as a discrete category, we compare using the original WSABIE [6] algorithm, and not the modified one as in the case of captions.

#### 7.5.4. Experiment-4: Image-Label Retrieval

Here we assume textual data to be in the form of labels and demonstrate results on the IAPR dataset. We use the set of labels as used in the recent image annotation works [3, 4, 5]. Similar to Experiment-3, for Im2Text, we rank the labels given a query image, and vice-versa for Text2Im. Also, we use the original WSABIE algorithm for comparisons.

## 7.6. Results and Discussion

### 7.6.1. Experiment-1: Image-Caption Retrieval

Table 2 compares the performance of different methods on all the datasets for both the tasks. Following observations can be made from these results: (i) For all the four methods (i.e., WSABIE, BITR-M, BITR-E and BITR-C), the performance usually improves (sometimes by a large margin) by using CTR as compared to TR. This reflects the advantage of explicitly incorporating cross-correlations into data representations. (ii) For Pascal dataset, relative performances of different methods follow almost similar trends for both Im2Text and Text2Im. However, there is comparatively more diversity in the other two datasets. This could be because Pascal dataset is relatively much smaller than the other two datasets, and the diversity of semantic concepts it covers is also less. This may result in dataset specific biases, and thus reflects the necessity of evaluating on big and diverse datasets such as SBU. (iii) For most of the cases, BITR-C (CTR) outperforms the other two variants of BITR. This implies that normalized correlation based loss function models the cross-modal patterns better than the other two loss functions. (iv) The performance of BITR-C (CTR) is either better than or comparable to the CCA [19] approach through-

Method	Pascal				IAPR				SBU			
	%B-1	%B-2	%B-3	%R-1	%B-1	%B-2	%B-3	%R-1	%B-1	%B-2	%B-3	%R-1
Ordonez et al. [14]	—	—	—	—	—	—	—	—	13.00	—	—	—
Gupta et al. [10]	36.00	9.00	1.00	21.00	15.00	6.00	1.00	14.00	—	—	—	—
BITR-C (CTR)	34.85	10.69	6.37	23.97	34.18	13.29	7.28	32.81	15.19	5.83	2.95	11.39

Table 3: Comparison between previously reported results and our results using BITR-C (CTR) for Im2Text under Experiment-1 (B-n means n-gram BLEU score, and R-1 means 1-gram Rouge score).

	Method	Im2Text				Text2Im			
		%BLEU-1↑	%Rouge-1↑	R@50↑	MedR↓	%BLEU-1↑	%Rouge-1↑	R@50↑	MedR↓
Pascal	CCA	20.29	<b>15.29</b>	<b>16.50</b>	<b>38.35</b>	20.15	<b>15.54</b>	<b>19.00</b>	<b>35.30</b>
	Wsabie (TR)	<b>20.37</b>	15.07	10.70	49.40	<b>21.01</b>	15.29	10.10	49.25
	Wsabie (CTR)	20.10	15.28	10.50	48.10	20.81	15.07	11.40	47.70
	BITR-M (TR)	20.78	15.73	13.80	42.45	21.55	15.74	10.30	49.25
	BITR-M (CTR)	21.75	17.04	15.50	42.50	22.53	<b>17.45</b>	22.30	29.85
	BITR-E (TR)	19.07	14.49	13.00	43.05	19.89	14.60	11.20	49.30
	BITR-E (CTR)	21.40	16.01	16.00	40.50	21.76	16.51	21.60	32.30
	BITR-C (TR)	21.27	15.98	14.80	41.55	22.82	15.46	10.10	49.35
	BITR-C (CTR)	<b>22.17</b>	<b>17.47</b>	<b>22.10</b>	<b>30.60</b>	<b>23.76</b>	17.41	<b>24.30</b>	<b>28.00</b>
IAPR	CCA	14.60	11.68	<b>6.30</b>	<b>706.20</b>	14.68	11.59	<b>7.11</b>	<b>641.50</b>
	Wsabie (TR)	<b>14.95</b>	<b>11.78</b>	2.32	988.75	14.05	10.98	2.72	1000.40
	Wsabie (CTR)	14.71	11.77	3.28	905.75	<b>14.84</b>	<b>11.85</b>	2.56	943.15
	BITR-M (TR)	15.23	12.75	4.50	816.90	15.04	12.53	2.50	976.55
	BITR-M (CTR)	16.54	14.18	4.97	768.85	15.86	13.44	7.52	564.85
	BITR-E (TR)	12.84	9.86	4.46	774.95	12.58	9.41	2.69	978.05
	BITR-E (CTR)	15.29	12.22	5.11	764.45	13.12	10.68	7.53	568.30
	BITR-C (TR)	15.84	13.64	4.72	805.30	14.49	11.48	2.70	970.15
	BITR-C (CTR)	<b>17.19</b>	<b>14.78</b>	<b>8.04</b>	<b>550.70</b>	<b>16.76</b>	<b>14.35</b>	<b>7.84</b>	<b>515.90</b>

Table 4: Comparison of the performance using baseline methods (CCA [19] and Wsabie [6]) and variants of our method for cross-dataset image-caption retrieval (Experiment-2). The best results using both are highlighted in bold. (↑: higher means better; ↓: lower means better.)

out, thus indicating the superiority of the proposed Structural SVM based cross-modal search framework over the CCA technique.

Table 3 shows comparisons on Im2Text with the reported results of a caption generation based approach [10] that uses image-to-image matching and a caption retrieval based approach [14] that uses both image-to-image matching as well as pre-trained object detectors and scene classifiers. The approach of [10] was shown to outperform other popular methods such as [12, 13], hence we do not include comparisons with them. Since both [14] and [10] depend on an annotated dataset consisting of both the modalities (image and text) modalities during the testing phase, and [10] generates captions rather than retrieving them, these results are not directly comparable. However, it is worth noticing that even by matching images directly with captions, our method performs either comparable to or superior than [10, 14]. This reflects its effectiveness in learning cross-modal semantic associations between images and captions.

#### 7.6.2. Experiment-2: Cross-dataset Image-Caption Retrieval

Table 4 shows the results for this experiment. Here we can observe that: (i) For all the methods, the performance degrades significantly compared to that in Experiment-1. This reflects the impact of dataset specific biases, and thus emphasizes the necessity of performing cross-dataset evaluations. (ii) As in Experiment-1, BITR-C performs better than other methods in almost all the cases. This suggests that the loss function  $\Delta_C(\cdot)$  (Eq. 7) could be a better choice in practice than the other two loss functions  $\Delta_M(\cdot)$  and  $\Delta_E(\cdot)$  for real-world applications. (iii) Unlike Experiment-1, the relative gains using BITR-C compared to CCA [19] are now much more pronounced. This demonstrates the better generalization ability across datasets achieved using our framework than CCA.

#### 7.6.3. Experiment-3: Image-Phrase Retrieval, and Experiment-4: Image-Label Retrieval

Table 5 and Table 6 compare different methods when textual data is in the form of phrases and labels respectively. Note that in contrast to all other methods, wKNN makes use of image-to-image similarity during the testing phase. Due to this, despite its simplicity, it mostly achieves very encouraging results

Method	Im2Text		Text2Im	
	%P@5	%mAP	%P@5	%mAP
CCA	5.22	4.13	1.82	2.62
wKNN	<b>8.38</b>	4.23	<b>2.04</b>	<b>2.98</b>
SVM	6.85	<b>5.12</b>	1.92	2.44
Wsabie (TR)	5.32	4.18	1.99	2.34
Wsabie (CTR)	5.86	4.54	1.58	2.30
BITR-M (TR)	5.95	4.38	1.79	2.28
BITR-M (CTR)	6.35	4.76	2.75	3.14
BITR-E (TR)	6.13	4.94	1.74	2.48
BITR-E (CTR)	8.63	5.01	2.84	3.49
BITR-C (TR)	5.98	4.97	1.95	2.55
BITR-C (CTR)	<b>8.68</b>	<b>5.16</b>	<b>2.88</b>	<b>3.68</b>

Table 5: Comparison of the performance using baseline methods (CCA [19], wKNN [4], SVM [58] and Wsabie [6]) and variants of our method for image $\leftrightarrow$ phrase retrieval on the IAPR dataset (Experiment-3). The best results using both are highlighted in bold. (Higher score means better performance.)

Method	Im2Text		Text2Im	
	%P@5	%mAP	%P@5	%mAP
CCA	17.29	17.08	4.88	2.97
wKNN	<b>19.31</b>	<b>19.78</b>	<b>8.25</b>	<b>4.30</b>
SVM	19.15	19.41	5.22	3.13
Wsabie (TR)	17.67	17.19	2.19	2.34
Wsabie (CTR)	17.79	18.11	2.48	2.89
BITR-M (TR)	19.06	17.95	7.90	4.28
BITR-M (CTR)	18.16	17.99	7.51	4.71
BITR-E (TR)	18.56	18.59	7.39	4.85
BITR-E (CTR)	19.32	19.34	8.44	4.95
BITR-C (TR)	18.75	18.64	8.16	4.59
BITR-C (CTR)	<b>19.48</b>	<b>19.65</b>	<b>8.71</b>	<b>5.18</b>

Table 6: Comparison of the performance using baseline methods (CCA [19], wKNN [4], SVM [58] and Wsabie [6]) and variants of our method for image $\leftrightarrow$ label retrieval on the IAPR dataset (Experiment-4). The best results using both are highlighted in bold. (Higher score means better performance.)

compared to other methods. Our methods (particularly BITR-C (CTR)) demonstrate competitive performance, and perform either comparable to or better than all other methods. We also observe that the results for phrases and labels follow quite similar trends. This is expected since in both the experiments, we consider each phrase/label as a discrete unit, thus focusing only on the co-occurrence of phrases/labels. An interesting direction for future research would be to build better representations for phrases that could capture hierarchical semantic correlations (among words co-occurring within a phrase, and among phrases co-occurring within an annotation).

#### 7.6.4. Qualitative Results

Figure 6 shows some qualitative results on the IAPR dataset. We observe that our method is able to correctly identify specific objects such as “building”, “bed”, “table”, “church”, “cyclist”, etc. Also, it is quite interesting that for Im2Text in Experiment-2, the predicted caption is quite meaningful and

representative of the image content even though it is from a different (SBU) dataset. This demonstrates the effectiveness of our approach in learning semantic relationships across the two modalities.

#### 7.7. Evaluation Using Contemporary Features

Recently, image features computed using CNN models [26, 62, 68] have become the de facto standards for several visual recognition tasks. Similarly, textual features based on word2vec [27]<sup>8</sup> are being popularly used in linguistic applications. While word2vec gives a 300-dimensional vector representation for text, many CNN models give a feature vector for images with a few thousands of dimensions. E.g., [26, 62, 68] give a 4096-dimensional image representation. If we directly use these two representations in BITR, the dimensionality of the joint feature vector would become around 1.2 million ( $= 4096 \times 300$ ), and this in turn would be computationally very expensive. However, some recent works like [69] have shown that it is possible to reduce significantly the size of image representation once they are learned, thus making our method compatible with CNN features. Precisely, in [69], it was shown that applying dimensionality reduction using Principal Component Analysis (PCA) on the CNN features can provide a very compact representation with almost no degradation in performance. Following this, first we compute a 4096-dimensional CNN representation for images using [26], and then compress it to 128-dimensional vector using PCA. This, along with 300-dimensional textual feature vector computed using word2vec, gives a 38400-dimensional joint feature vector, thus making BITR compatible with these features.

We evaluate these features on the image-caption retrieval task as discussed under Experiment-1 (Section 7.5.1). Table 7 compares the performance of different methods. As compared to using simple bag-of-words based features (*c.f.* Table 2), the new features provide better performance for all the methods when we consider generation-based evaluation metrics BLEU and Rouge. Similarly, for retrieval-based evaluation metrics, the performance improves on all the datasets, except on Pascal where it degrades for MedianRank. This indicates that for small datasets, now more number of samples have relevant results in the top-K predictions, however their individual ranks go down. Analogous to the previous results, we can observe that: (i) BITR-C outperforms the other two variants of BITR in most of the cases, thus confirming the practical utility of normalized correlation based loss function. (ii) Also, the performance of BITR-C is either comparable to or better than CCA on all the three datasets. Overall, this experiment demonstrates the applicability of our approach in general, and validates that it can be used with modern CNN and word2vec features as well.

#### 7.8. Discussion and Take-home-messages

As we observed in the experiments, what features one uses will have a critical impact on the performance. Moreover, different combinations of features and loss functions may perform

<sup>8</sup><http://code.google.com/p/word2vec/>





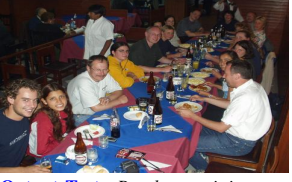





Experiment-1		Experiment-2	
Im2Text	Text2Im	Im2Text	Text2Im
<b>Query Image:</b>  <b>Output Text:</b> A long three-storey building with a glass facade; a road with blue boards and a grey fence in the foreground and a blue sky in the background.	<b>Query Text:</b> A room with white walls, black door and window frames, a black carpet and three single beds made of wood with black and white bedcovers. <b>Output Image:</b> 	<b>Query Image:</b>  <b>Output Text:</b> People are sitting at a laid table made of wood in a little dark restaurant.	<b>Query Text:</b> Two local teachers are standing in a classroom with many children sitting at their wooden desks. <b>Output Image:</b> 
Experiment-3		Experiment-4	
Im2Text	Text2Im	Im2Text	Text2Im
<b>Query Image:</b>  <b>Output Text:</b> hill with bushes	<b>Query Text:</b> church with corner <b>Output Image:</b> 	<b>Query Image:</b>  <b>Output Text:</b> water	<b>Query Text:</b> cyclist <b>Output Image:</b> 

Figure 6: Qualitative results on IAPR TC-12 dataset for the two tasks for the four experiments.

better than others for different problems. In the experiments, our primary motivation for performing feature transformation was to maintain computational load. In practice, it is possible to apply our method even if there is no higher level transformation at all. This is because we can form the joint representation  $\Psi$  by computing an outer product between any real-valued input and output feature vectors. Also, the three loss functions that we use are based on general distance/similarity metrics (Manhattan distance, Euclidean distance and cosine similarity). Each of these metrics are applicable to real-valued vectors. Only the cosine similarity based loss function (Eq. 7) makes an assumption that the feature vectors are  $L_2$ -normalized, and this normalization can be easily applied on a real-valued vector.

In a broader sense, our framework can be viewed as a support vector based counterpart of the nearest-neighbour based cross-modal matching techniques such as CCA [19]. The goal of both the techniques is to compute similarity of a sample in one modality with that in another. Similar to the distance/similarity measures like Manhattan distance, Euclidean distance and normalized correlation used in cross-modal matching [19], we have shown our framework to work with these measures by mapping them as loss functions of Structural SVM. This analogy is further evident from the fact that while the similarity metric based on correlation was found to achieve the best performance in [19], similar results are observed in our experiments as well, where the BTR-C variant (that uses correlation based loss function) mostly performs better than the other two. However, unlike the nearest-neighbour based method of [19], our approach usually provides better performance in both within-dataset (Experiment-1,3,4) as well as cross-dataset (Experiment-2) settings. This is because it is

based on Structural SVM that provides good generalization and max-margin guarantees. Particularly in the cross-dataset experiments, our approach consistently outperforms CCA, sometimes significantly.

As discussed in Section 2, several techniques for cross-modal retrieval such as [7, 39, 21] are based on learning a transformation of cross-modal input/output features. During inference, they usually adopt some simple similarity criteria such as cosine similarity in the transformed space. Our approach can serve as an improved inference technique for all such methods, where rather than using cosine similarity, one can learn a support vector model  $w$  over the transformed features and use it for inference. Though this will add another layer of training, it will be a one-time process. Moreover, there will be almost no effect on the testing time as discussed in Section 5.2.

## 8. Conclusion and Future work

We have presented a novel Structural SVM based framework to perform cross-modal multimedia retrieval. We have demonstrated the applicability of our method to cross-modal search on two medium and one web-scale dataset. For both Im2Text and Text2Im, our method achieved promising results and outperformed competing baseline techniques. In this work, we have considered visual (image) and textual data as the two modalities, nevertheless the fundamental ideas discussed can be applied to cross-modal retrieval tasks in other domains as well.

Directions for future research include implementing an efficient training algorithm for our approach that could scale to millions of samples with high-dimensional joint feature represen-

	Method	Im2Text				Text2Im			
		%BLEU-1↑	%Rouge-1↑	R@50↑	MedR↓	%BLEU-1↑	%Rouge-1↑	R@50↑	MedR↓
Pascal	CCA	<b>34.69</b>	14.29	<b>72.80</b>	<b>21.25</b>	<b>34.07</b>	16.49	<b>85.10</b>	<b>20.65</b>
	Wsabie	32.98	<b>23.43</b>	35.40	28.30	33.78	<b>22.49</b>	36.90	33.45
	BITR-M	34.66	24.73	70.50	24.35	35.02	25.97	80.40	19.10
	BITR-E	36.12	25.78	74.30	19.60	35.43	26.37	84.70	21.15
	BITR-C	<b>37.12</b>	<b>26.02</b>	<b>75.10</b>	<b>18.75</b>	<b>37.41</b>	<b>26.81</b>	<b>85.20</b>	<b>18.15</b>
IAPR	CCA	<b>31.63</b>	<b>32.78</b>	<b>23.18</b>	<b>277.05</b>	<b>32.79</b>	<b>31.21</b>	<b>22.68</b>	<b>259.00</b>
	Wsabie	29.64	26.85	9.42	571.17	28.40	28.71	7.24	604.89
	BITR-M	34.71	33.98	19.36	261.75	31.02	29.43	21.68	304.85
	BITR-E	35.68	34.59	21.69	283.25	31.95	30.88	<b>22.38</b>	<b>262.35</b>
	BITR-C	<b>36.72</b>	<b>35.45</b>	<b>22.67</b>	<b>243.90</b>	<b>33.61</b>	<b>31.76</b>	19.82	423.55
SBU	CCA	<b>14.52</b>	12.37	<b>21.40</b>	<b>153.50</b>	<b>16.81</b>	<b>13.78</b>	<b>25.90</b>	<b>158.50</b>
	Wsabie	13.95	<b>12.80</b>	16.90	190.00	16.35	13.42	15.30	194.50
	BITR-M	15.14	13.66	23.70	164.00	18.25	15.26	24.30	207.50
	BITR-E	16.28	<b>15.19</b>	26.10	142.50	19.03	15.47	26.80	175.00
	BITR-C	<b>17.23</b>	14.92	<b>29.80</b>	<b>129.00</b>	<b>19.74</b>	<b>15.86</b>	<b>31.40</b>	<b>138.00</b>

Table 7: Comparison of the performance using baseline methods (CCA [19] and Wsabie [6]) and variants of our method for image↔caption retrieval (Experiment-1) using CNN image features [26, 69] and word2vec [27] textual features. The best results using both are highlighted in bold. (↑: higher means better; ↓: lower means better.)

tations, and building better representations for phrases and captions that could capture hierarchical correlations among words.

## Appendix A. Extending WSABIE for Captions

Here, first we briefly discuss the WSABIE algorithm [6], and then present the proposed extension of WSABIE to adapt it for captions.

### Appendix A.1. WSABIE

WSABIE (Web Scale Annotation by Image Embedding) learns a mapping space where both images and annotations (e.g. labels) are represented. The mapping functions for both the modalities are learned jointly by minimizing the WARP (Weighted Approximate-Rank Pairwise) loss, that is based on optimizing precision at  $k$ . Each image is represented by  $x \in \mathbb{R}^p$ , and each annotation  $i \in \mathcal{Y} = \{1, \dots, Y\}$ , where  $Y$  is the (fixed) vocabulary size. Then, a mapping is learned from image feature space to the joint space  $\mathbb{R}^P$ :

$$\Phi_I(x) : \mathbb{R}^p \rightarrow \mathbb{R}^P. \quad (\text{A.1})^{100}$$

while jointly learning a mapping function for annotations:

$$\Phi_W(i) : \{1, \dots, Y\} \rightarrow \mathbb{R}^P. \quad (\text{A.2})$$

Both these mappings are chosen to be linear; i.e.,  $\Phi_I(x) = Vx$ , and  $\Phi_W(i) = W_i$  where  $W_i$  indexes the  $i^{th}$  column of a  $P \times Y$  matrix. The goal is to learn the possible annotations of a given image such that the highest ranked ones best describe the semantic content of the image. For this, the following model is considered:

$$f_i(x) = \Phi_W(i)^T \Phi_I(x) = W_i^T Vx, \quad (\text{A.3})$$

### Algorithm 1 WSABIE Algorithm

**Require:** labeled data  $(x_i, y_i), y_i \in \{1, \dots, Y\}$

**repeat**

Pick a random labeled example  $(x_i, y_i)$

Let  $f_{y_i}(x_i) = \Phi_W(y_i)^T \Phi_I(x_i)$

Set  $N = 0$

**repeat**

Pick a random annotation  $\bar{y} \in \{1, \dots, Y\} \setminus y_i$ .

Let  $f_{\bar{y}}(x_i) = \Phi_W(\bar{y})^T \Phi_I(x_i)$

$N = N + 1$

**until**  $f_{\bar{y}}(x_i) > f_{y_i}(x_i) - 1$  or  $N \geq Y - 1$

**if**  $f_{\bar{y}} > f_{y_i}(x_i) - 1$  **then**

Make a gradient step to minimize:

$$L(\lfloor \frac{Y-1}{N} \rfloor) |1 - f_{y_i}(x_i) + f_{\bar{y}}(x_i)|_+$$

Project weights to enforce constraints in Eq. A.4.

**end if**

**until** validation error does not improve.

where the possible annotations  $i$  are ranked according to the magnitude of  $f_i(x)$  in descending order. This family of models have constrained norm:

$$\begin{aligned} \|V_i\|_2 &\leq \lambda, i = 1, \dots, p, \\ \|W_i\|_2 &\leq \lambda, i = 1, \dots, Y. \end{aligned} \quad (\text{A.4})$$

which acts as a regularizer. Algorithm 1 shows the pseudo-code for learning model variables using a stochastic gradient descent algorithm that minimizes WARP loss (where  $L(k) = \sum_{j=1}^k \alpha_j$ , with  $\alpha_j = \frac{1}{j}$ ).

### Appendix A.2. Adapting WSABIE for Captions

In case of captions, we have a (training) set of captions  $\mathcal{C} = \{c_i\}$  rather than a fixed annotation vocabulary. In order to adapt

---

**Algorithm 2** Adapted WSABIE Algorithm for Captions

**Require:** labeled data  $(x_i, c_i)$ ,  $y$  is a feature vector representing caption  $c \in \mathcal{C}$

**repeat**

Pick a random labeled example  $(x_i, c_i)$

Let  $g_{y_i}(x_i) = \Phi_Z(y_i)^T \Phi_I(x_i)$

Set  $N = 0$

**repeat**

Pick a random caption  $\bar{c} \in \mathcal{C} \setminus c_i$ .

Let  $g_{\bar{y}}(x_i) = \Phi_Z(\bar{y})^T \Phi_I(x_i)$

$N = N + 1$

**until**  $g_{\bar{y}}(x_i) > g_{y_i}(x_i) - 1$  or  $N \geq |\mathcal{C}| - 1$

**if**  $g_{\bar{y}} > g_{y_i}(x_i) - 1$  **then**

Make a gradient step to minimize:

$$L(\lfloor \frac{|\mathcal{C}|-1}{N} \rfloor) |1 - g_{y_i}(x_i) + g_{\bar{y}}(x_i)|_+ \quad (A.5)$$

Project weights to enforce constraints in Eq. A.7.

**end if**

**until** validation error does not improve.

---

WSABIE for captions, we modify the feature mapping given in Eq. A.2 such that instead of learning a separate parameter vector for each annotation, we learn a single parameter matrix<sup>150</sup> for all the captions. Given a caption  $c \in \mathcal{C}$  represented by  $y \in \mathbb{R}^q$ , a mapping is learned from caption feature space to the joint space  $\mathbb{R}^P$ :

$$\Phi_Z(y) : \mathbb{R}^q \rightarrow \mathbb{R}^P, \quad (A.5)^{155}$$

where  $Z$  is a  $P \times q$  matrix. Now, given a set of captions, the goal is to learn the possible caption(s) of a given image such that the highest ranked ones best describe the semantic content<sup>160</sup> of the image. For this, the following model is considered:

$$g_y(x) = \Phi_Z(y)^T \Phi_I(x) = y^T Z^T V x. \quad (A.6)$$

Similar to Eq. A.4, this family of models have constrained<sup>165</sup> norm:

$$\begin{aligned} \|V_i\|_2 &\leq \lambda, i = 1, \dots, p, \\ \|Z_i\|_2 &\leq \lambda, i = 1, \dots, q. \end{aligned} \quad (A.7)^{170}$$

which acts as a regularizer. Algorithm 2 shows the pseudo-code for learning the model variables using a stochastic gradient descent algorithm. It is similar to Algorithm 1 except that instead of randomly picking an annotation from vocabulary, now we randomly pick a caption from the training set consisting of image-caption pairs.

## Acknowledgement

Yashaswi Verma is partly supported by Microsoft Research India PhD fellowship 2013.

## References

- [1] P. Duygulu, K. Barnard, J. F. G. de Freitas, D. A. Forsyth, Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary, in: ECCV, 2002.
- [2] S. L. Feng, R. Manmatha, V. Lavrenko, Multiple Bernoulli relevance models for image and video annotation, in: CVPR, 2004.
- [3] A. Makadia, V. Pavlovic, S. Kumar, Baselines for image annotation, Int. J. Comput. Vision 90 (1) (2010) 88–105.
- [4] M. Guillaumin, T. Mensink, J. Verbeek, C. Schmid, TagProp: Discriminative metric learning in nearest neighbour models for image auto-annotation, in: ICCV, 2009.
- [5] Y. Verma, C. V. Jawahar, Image annotation using metric learning in semantic neighbourhoods, in: ECCV, 2012.
- [6] J. Weston, S. Bengio, N. Usunier, WSABIE: Scaling up to large vocabulary image annotation, in: IJCAI, 2011.
- [7] Y. Gong, Q. Ke, M. Isard, S. Lazebnik, A multi-view embedding space for modeling internet images, tags, and their semantics, IJCV 106 (2) (2013) 210–233.
- [8] M. A. Sadeghi, A. Farhadi, Recognition using visual phrases, in: CVPR, 2011.
- [9] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, Y. Choi, Collective generation of natural image descriptions, in: ACL, 2012.
- [10] A. Gupta, Y. Verma, C. V. Jawahar, Choosing linguistics over vision to describe images, in: AAAI, 2012.
- [11] Y. Verma, A. Gupta, P. Mannem, C. V. Jawahar, Generating image descriptions using semantic similarities in the output space, in: CVPR Workshop, 2013.
- [12] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, T. L. Berg, Baby Talk: Understanding and generating simple image descriptions, in: CVPR, 2011.
- [13] Y. Yang, C. L. Teo, H. D. III, Y. Aloimonos, Corpus-guided sentence generation of natural images, in: EMNLP, 2011.
- [14] V. Ordonez, G. Kulkarni, T. L. Berg, Im2Text: Describing images using 1 million captioned photographs, in: NIPS, 2011.
- [15] A. Farhadi, M. Hejrati, A. Sadeghi, P. Young, C. Rashtchian, J. Hockenmaier, D. Forsyth, Every picture tells a story: Generating sentences for images, in: ECCV, 2010.
- [16] M. Hodosh, P. Young, J. Hockenmaier, Framing image description as a ranking task: Data, models and evaluation metrics, JAIR 47 (2013) 853–899.
- [17] M. Paramita, M. Sanderson, P. Clough, Diversity in photo retrieval: overview of the ImageCLEFPhoto task 2009, CLEF working notes.
- [18] R. Datta, D. Joshi, J. Li, J. Wang, Image retrieval: Ideas, influences and trends of new age, ACM Computing Surveys 40 (2) (2008) 1–60.
- [19] N. Rasiwasia, J. C. Pereira, E. Coviello, G. Doyle, G. R. G. Lanckriet, R. Levy, N. Vasconcelos, A new approach to cross-modal multimedia retrieval, in: ACM MM, 2010.
- [20] D. R. Hardoon, S. Szedmak, J. Shawe-Taylor, Canonical correlation analysis: An overview with application to learning methods, Neural Comput. 16 (12) (2004) 2639–2664.
- [21] C. Kang, S. Xiang, S. Liao, C. Xu, C. Pan, Learning consistent feature representation for cross-modal multimedia retrieval, IEEE Transactions on Multimedia 17 (3) (2015) 370–381.
- [22] I. Tschantzaris, T. Hofmann, T. Joachims, Y. Altun, Support vector machine learning for interdependent and structured output spaces, in: ICML, 2004.
- [23] C. Rashtchian, P. Young, M. Hodosh, J. Hockenmaier, Collecting image annotation using amazon’s mechanical turk, in: NAACLHLT Workshop on Creating Speech and Language Data with Amazon’s Mechanical Turk, 2010.
- [24] M. Grubinger, P. D. Clough, H. Müller, T. Deselaers, The IAPR benchmark: A new evaluation resource for visual information systems, in: International Conference on Language Resources and Evaluation, 2006.
- [25] Y. Verma, C. V. Jawahar, Im2Text and Text2Im: Associating images and texts for cross-modal retrieval, in: BMVC, 2014.
- [26] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A deep convolutional activation feature for generic visual recognition, 2014.
- [27] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: NIPS, 2013.

- [28] A. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, *PAMI* 22 (12) (2000) 1349–1380.
- [29] C. Meadow, B. Boyce, D. Kraft, C. Barry, Text information retrieval systems, Emerald Group Pub Ltd.
- [30] A. F. Smeaton, P. Over, W. Kraaij, Evaluation campaigns and trecvid, in: *MIR: Proceedings of the 8th ACM International Workshop on Multimedia*, Information Retrieval, 2006.
- [31] Y. Verma, C. V. Jawahar, Exploring SVM for image annotation in presence of confusing labels, in: *BMVC*, 2013.
- [32] Z. Niu, G. Hua, X. Gao, Q. Tian, Semi-supervised relational topic model for weakly annotated image recognition in social media, in: *CVPR*, 2014.
- [33] L. Ballan, T. Uricchio, L. Seidenari, A. D. Bimbo, A cross-media model for automatic image annotation, in: *ICMR*, 2014.
- [34] N. Srivastava, R. Salakhutdinov, Multimodal learning with deep boltzmann machines, in: *NIPS*, 2012.
- [35] H. J. Escalante, C. A. Hernández, L. E. Sucar, M. Montes, Late fusion of heterogeneous methods for multimedia image retrieval, in: *MIR*, 2008.
- [36] J. C. Caicedo, J. BenAbdallah, F. A. González, O. Nasraoui, Multimodal representation, indexing, automated annotation and retrieval of image collections via non-negative matrix factorization, *Neurocomput.* 76 (1) (2012) 50–60.
- [37] S. J. Hwang, K. Grauman, Learning the relative importance of objects from tagged images for retrieval and cross-modal search, *Int. J. Comput. Vision* 100 (2) (2012) 134–153.
- [38] A. Sharma, A. Kumar, H. D. III, D. W. Jacobs, Generalized multiview analysis: A discriminative latent space, in: *CVPR*, 2012.
- [39] N. Rasiwasia, D. Mahajan, V. Mahadevan, G. Aggarwal, Cluster canonical correlation analysis, in: *AISTATS*, 2014.
- [40] K. Wang, R. He, W. Wang, L. Wang, T. Tan, Learning coupled feature spaces for cross-modal matching, in: *ICCV*, 2013.
- [41] T. Mei, Y. Rui, S. Li, Q. Tian, Multimedia search reranking: A literature survey, *ACM Comput. Surv.* 46 (3) (2014) 38:1–38:38.
- [42] R. Rosipal, N. Krämer, Overview and recent advances in partial least squares, in: *SLSFS*, 2006.
- [43] J. B. Tenenbaum, W. T. Freeman, Separating style and content with bilinear models, *Neural Comput.* 12 (6) (2000) 1247–1283.
- [44] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, A. Y. Ng, Grounded compositional semantics for finding and describing images with sentences, *TACL* 2 (2013) 207–218.
- [45] A. Karpathy, A. Joulin, L. Fei-Fei, Deep fragment embeddings for bidirectional image sentence mapping, in: *NIPS*, 2014.
- [46] J. Rodriguez, F. Perronnin, Label embedding for text recognition, in: *BMVC*, 2013.
- [47] K. Duan, D. J. Crandall, D. Batra, Multimodal learning in loosely-organized web images, in: *CVPR*, 2014.
- [48] J. J. McAuley, J. Leskovec, Image labeling on a network: Using social-network metadata for image classification, in: *ECCV*, 2012.
- [49] J. Johnson, L. Ballan, L. Fei-Fei, Love thy neighbors: Image annotation by exploiting image metadata, in: *ICCV*, 2015.
- [50] H. Hu, G.-T. Zhou, Z. Deng, Z. L. and Greg Mori, Learning structured inference neural networks with label relations, in: *CVPR*, 2016.
- [51] Y. Ushiku, T. Harada, Y. Kuniyoshi, Understanding images with natural sentences, in: *ACM MM*, 2011.
- [52] B. Plummer, L. Wang, C. Cervantes, J. Caicedo, J. Hockenmaier, S. Lazebnik, Flickr30k entities: Collecting region-to-phrase correspondences for richer image-to-sentence models, in: *ICCV*, 2015.
- [53] A. Karpathy, L. Fei-Fei, Deep visual-semantic alignments for generating image descriptions, in: *CVPR*, 2015.
- [54] J. Mao, W. Xu, Y. Yang, J. Wang, A. L. Yuille, Explain images with multimodal recurrent neural networks, in: *NIPS Deep Learning Workshop*, 2014.
- [55] R. Kiros, R. Salakhutdinov, R. S. Zemel, Unifying visual-semantic embeddings with multimodal neural language models, *TACL*.
- [56] H. Fang, S. Gupta, F. Iandola, R. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. Platt, L. Zitnick, G. Zweig, From captions to visual concepts and back, in: *CVPR*, 2015.
- [57] K. Papineni, S. Roukos, T. Ward, W. Zhu, Language models for image captioning: The quirks and what works, in: *ACL*, 2015.
- [58] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*, Cambridge University Press, 2000.
- [59] A. K. Menon, D. Surian, S. Chawla, Cross-modal retrieval: A pairwise classification approach, in: *SIAM International Conference on Data Mining*, 2015.
- [60] D. Blei, A. Ng, M. Jordan, Latent dirichlet allocation, *JMLR* 12 (1) (2003) 234–278.
- [61] D. G. Lowe, Distinctive image features from scale-invariant keypoints, *IJCV* 60 (2) (2004) 91–110.
- [62] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: *CVPR*, 2014.
- [63] K. Papineni, S. Roukos, T. Ward, W. Zhu, BLEU: A method for automatic evaluation of machine translation, in: *ACL*, 2002.
- [64] C.-Y. Lin, E. Hovy, Automatic evaluation of summaries using n-gram co-occurrence statistics, in: *NAACLHLT*, 2003.
- [65] H. Hotelling, Relations between two sets of variates, *Biometrika* 28 (1936) 321–377.
- [66] C. D. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. J. Bethard, D. McClosky, The Stanford CoreNLP natural language processing toolkit, in: *ACL: System Demonstrations*, 2014.  
URL <http://nlp.stanford.edu/software/corenlp.shtml>
- [67] M.-C. de Marneffe, C. D. Manning, The stanford typed dependencies representation, in: *COLING Workshop on Cross-framework and Cross-domain Parser Evaluation*, 2008.
- [68] M. D. Zeiler, R. Fergus, Visualizing and understanding convolutional networks, in: *ECCV*, 2014.
- [69] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: *ECCV*, 2014.