Robust Visual-Textual Sentiment Analysis: When Attention meets Tree-structured Recursive Neural Networks

Quanzeng You[1], Liangliang Cao[2], Hailin Jin[3], Jiebo Luo[1]
[1]University of Rochester, Buffalo, NY USA
[2]Yahoo Labs, New York, NY USA
[3]Adobe Research, San Jose, CA USA
{qyou, jluo}@cs.rochester.edu, liangliang@yahoo-inc.com, hljin@adobe.com

ABSTRACT

Sentiment analysis is crucial for extracting social signals from social media content. Due to huge variation in social media, the performance of sentiment classifiers using single modality (visual or textual) still lags behind satisfaction. In this paper, we propose a new framework that integrates textual and visual information for robust sentiment analysis. Different from previous work, we believe visual and textual information should be treated jointly in a structural fashion. Our system first builds a semantic tree structure based on sentence parsing, aimed at aligning textual words and image regions for accurate analysis. Next, our system learns a robust joint visual-textual semantic representation by incorporating 1) an attention mechanism with LSTM (long short term memory) and 2) an auxiliary semantic learning task. Extensive experimental results on several known data sets show that our method outperforms existing the state-of-the-art joint models in sentiment analysis. We also investigate different tree-structured LSTM (T-LSTM) variants and analyze the effect of the attention mechanism in order to provide deeper insight on how the attention mechanism helps the learning of the joint visual-textual sentiment classifier.

CCS Concepts

•Information systems → Multimedia information systems; Information systems applications; •Computing methodologies → Computer vision; •General and reference → *Measurement;*

Keywords

joint visual-textual sentiment analysis; tree-structured joint model; multimodality analysis; attention mechanism

1. INTRODUCTION

Many classical studies [7, 15, 14, 19, 42] in integrating visual and textural features follow similar procedures. First, image information and text information are treated separately using different domain-specific knowledge and techniques in computer vision and natural language processing. Next, the individual representations of two moralities will be integrated to build the final classifier. Very few studies have considered models that can explore deeper correlation between the visual and textural representations.

Figure 1: An example image tweet about a car accident, where the associated text significantly helps learning an effective visual representation for sentiment analysis. Words related to the objects and visual attributes are closely related to regions with matching colors.

Figure 1 shows an example that suggest an image and the associated text should be treated in a closely knit fashion. There are a number of objects and attribute descriptors in the text, including "people", "car", "bus", and "accident", which are associated with specific regions in the image. If we are to parse such correlated text and image content accurately, these words and image regions should be modeled jointly in an intimate fashion.

The recent progress in sentence parsing suggests that the semantic features of sentences can be well modeled by a tree-structured model with semantic dependency [28, 15]. We propose to utilize such a structured model for multimodal visual-textual analysis. To model the visual features with the semantic parsing tree, we employ deep neural networks to model both textual and visual semantics. In particular, we generalize the standard Recurrent Neural Networks [31] by considering the regions of attention in the image, and build a unified model to leverage the information from the two modalities.

In this work, we intend to bridge the gap between vision and language using a tree-structured model on both image and text for joint visual-textual sentiment analysis. Figure 1 includes one image and sentence pair to illustrate our motivation. Generally, sentiment or opinion related words can be aligned to certain image regions. We are interested in discovering such kinds of alignments and subsequently building a robust joint model for sentiment analysis. Following previous studies on tree-structured Long Short-Term Memory (LSTM) over text [33, 44], we integrate image regions into the model using an attention mechanism. In particular, we propose a

MM '16, October 15-19, 2016, Amsterdam, Netherlands

DOI: http://dx.doi.org/10.1145/2964284.2971475

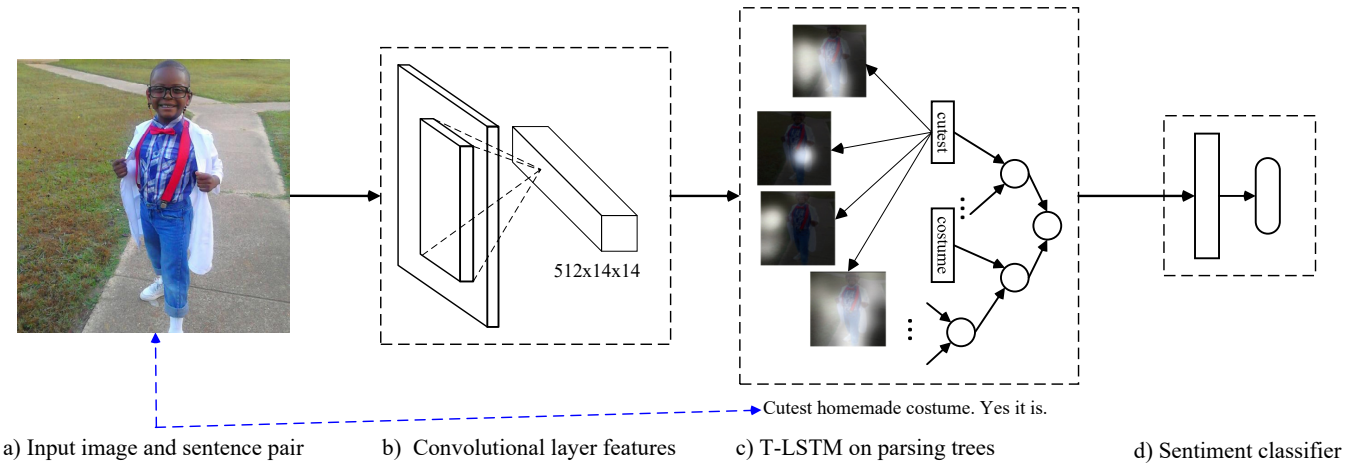


Figure 2: Steps of building a tree-structured LSTM (T-LSTM) model for joint sentiment analysis given a pair of sentence and image.

bilinear attention model to learn the relatedness between words and image regions. Consequently, the model is capable of structurally encoding both local vision and language features into a global semantic embedding feature space. Figure 2 shows the main steps of the proposed framework for joint sentiment analysis. The inputs to our model are pairs of image and its description. We employ CNN (convolutional neural network) for visual feature extraction. Next, we build a tree-structured Long Short-Term Memory (T-LSTM) model on top of the parsing trees to learn the joint feature representations. The attention mechanism is employed to learn the alignment between image regions and descriptive words. The hidden responses at the root node of the tree are provided as inputs to build a sentiment classifier in the end.

We apply our proposed model to the specific problem of analyzing sentiment from associated text and image. The problem of sentiment understanding of images has become popular in recent years [35, 4, 2, 42]. In particular, You *et al.* [42] proposed a cross-modality consistent regression (CCR) scheme for joint textual-visual sentiment analysis. Their approach employs deep visual and textual features to learn a regression model. By encouraging a consistent prediction between different modalities, their model achieved better performance over other different fusion models. However, almost all the previous work use simple early fusion or late fusion to combine two moralities, overlooking structured information coupling between the text and image. In contrast, our system introduces: 1) the use of an attention mechanism on local image regions and 2) joint visual-textual representation learning through tree-structured LSTM. Our system is capable of learning the semantic mapping between image regions and the words in the associated sentence. In addition, tree-structured LSTM facilitates the encoding of structured information into the joint semantic embedding. We compare the proposed model with several state-of-the-art baselines extensively on several data sets. The experimental results demonstrate the superiority of the proposed joint tree-structured model.

We make the following contributions in this work:

- We employ tree-structured LSTM (T-LSTM) for joint textual-visual sentiment analysis, which leads to better mapping between textual words and image regions,
- We introduce a bilinear attention model to facilitate a robust joint textual-visual feature representation, and prevent the model from being dominated by single modality.
- We adopt a Siamese network as an auxiliary task to learn the semantic embedding between text and image, helping the attention mechanism to achieve high effectiveness.
- Our proposed framework outperforms the state-of-arts in three existing datasets and one new collected dataset.

In the following sections, we first briefly introduce LSTM and tree-structured LSTM. Next, we present the details of the components of the proposed framework. Finally, we describe the design of the experiments and analyze the results.

2. RELATED WORK

Computer vision and natural language processing are important application domains of machine learning. Recently, deep learning has made significant advances in tasks related to both vision and language [16]. Conseuquently, the task of higher-level semantic understanding, such as machine translation [1], image aesthetic analysis [18], and visual sentiment analysis [3, 41] have become tractable. A more interesting and challenging task is to bridge the semantic gap between vision and language, and thus help solve more challenging problem.

The successes of deep learning make the understanding and jointly modeling vision and language content a feasible and attractive research topic. In the context of deep learning, many related publications have proposed novel models that address image and text simultaneously. Starting with matching images with word-level concepts [7] and recently onto sentence-level descriptions [15, 28, 19, 20, 14], deep neural networks exhibit significant performance improvements on these tasks. Despite of the fact that there are no semantic and syntactic structures, these models have inspired the idea of joint feature learning [30], semantic transfer [7] and design of margin ranking loss [38].

Notably, automatic image captioning is widely studied [20, 6, 5, 34, 14], which more intimately connects visual content and language semantics. Instead of learning a semantic mapping space for image and sentence pairs, automatic image captioning systems are expected to generate a sentence describing the given image. Image features are commonly integrated into the generation of the captions, which is typically modeled by a neural language model.

In general, these models handle two primary tasks: 1) how to represent image and text, and 2) how to learn the model on top of visual and textual features. Indeed, Convolutional Neural Networks (CNNs) [16, 27, 32] become the common approaches for extracting

visual features. Meanwhile, multimodal semantic mapping following a pairwise ranking loss is widely adopted for optimizing joint visual and textual models. Recently, different approaches, including sequential [15, 20, 14] and tree-structured models [28, 19], are selected to encode the text, among which Recurrent Neural Networks [31] and Recursive Neural Networks [29] are particularly popular. Indeed, both sequential and recursive models are closely related to the language and semantic attributes of text. On the other hand, visual content also have spatial and semantic structures [22]. Tree-structured models can benefit vision tasks in many ways [29, 40]. However, there have been few previous studies which connect the tree structures in both vision and text.

In this work, we focus on joint visual-textual sentiment analysis, which is an important task of bridging vision and language semantics. Different from the widely studied textual sentiment analysis [23], visual sentiment analysis is quite new and challenging. There are several recent works on visual sentiment analysis using initially pixel-level features [26], then mid-level attributes [2, 43], more recently deep visual features [41] and soical contextual information [37, 36]. These approaches have achieved acceptable performance on visual sentiment analysis. However, due to the complex nature of visual content, the performance of visual sentiment analysis still lags behind textual sentiment analysis.

There are also several publications on analyzing sentiment using multi-modalities, such as text and image. Both [35] and [4] employed both text and images for sentiment analysis, where late fusion is employed to combine the prediction results of using n-gram textual features and mid-level visual features [2]. More recently, You *et al.* [42] proposed a cross-modality consistent regression (CCR) scheme for joint textual-visual sentiment analysis. Their approach employed deep visual and textual features to learn a regression model. Their model achieved the best performance over other fusion models, however, overlook the structured mapping between image regions and words.

Our work is built on the long short-term memory (LSTM) model. In the next section, we will first introduce some basics of LSTM and then discuss our new model.

3. LONG SHORT-TERM MEMORY (LSTM) NETWORKS

For completeness, we present a brief introduction of the sequential LSTM model. With a Recurrent Neural Network (RNN), we are trying to predict the output sequence $\{y_1, y_2, \ldots, y_T\}$ given the input sequence $\{x_1, x_2, \ldots, x_T\}$. Between the input layer and the output layer, there is a hidden layer, and the current hidden state h_t is estimated using a recurrent unit (Eq.(1)):

$$h_t = f(h_{t-1}, x_t) \tag{1}$$

where x_t is the current input, h_{t-1} is the previous hidden state and $f(\cdot)$ accepts both x_t and h_{t-1} as inputs and produces the current output h_t. $f(\cdot)$ can be an activation function or other unit, such as long short-memory cell, which is one of the most widely deployed architectures [8]. Long short-memory cell can overcome the gradient *vanishing* issue [24]. Each LSTM cell c is controlled by an input gate i, an output gate o and an forget gate f, which is able to *remember* the error during the error propagation [12]. Subsequently, LSTM is capable of modeling long-range dependencies [14].

Let $W_.^i$, $W_.^f$, $W_.^o$ represent the parameters of the input, forget and output gate respectively. $\odot$ denotes the element-wise multiplication between two vectors. σ is the logistic sigmoid function. The precise form of LSTM cell is described in the following equations [9, 11, 10]. For conciseness, we omit all the bias terms of linear transformations in the paper.

$$\begin{aligned}
i_t &= \sigma(W_x^i x_t + W_h^i h_{t-1}) && (2)\\
f_t &= \sigma(W_x^f x_t + W_h^f h_{t-1}) && (3)\\
o_t &= \sigma(W_x^o x_t + W_h^o h_{t-1}) && (4)\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tanh\left(W_x^c x_t + W_h^c h_{t-1}\right) && (5)\\
h_t &= o_t \odot \tanh(c_t) && (6)
\end{aligned}$$

LSTM cells can also be used in non-sequential models. Recently, it has been utilized within tree structured models [33, 44]. Information is no longer propagated sequentially from current node to the next node as in sequential LSTM architectures. In tree-structured LSTM (T-LSTM) model, each internal node of the tree collects information from all of its child nodes and propagates the processed information to its parent node. In such a way, the hidden state obtained at the root node of the tree is considered to be the joint representation of the inputs at the leaf nodes of the tree. Each non-leaf node i in the tree is modeled using a LSTM cell. It will receive the hidden state of its j-th child as the j-th input. To accept multiple inputs, we use the following equations [33] to define the forward computation for a node i from all of its child nodes.

$$i_i = \sigma(W_x^i x_i + \sum_{n=1}^{N_i} W_n^i h_n^i) \tag{7}$$

$$f_k^i = \sigma(W_x^f x_i + \sum_{n=1}^{N_i} W_n^f h_n^i) \tag{8}$$

$$o_i = \sigma(W_x^o x_i + \sum_{n=1}^{N_i} W_n^h h_n^i) \tag{9}$$

$$c_i = \sum_{n=1}^{N_i} f_n^i \odot c_n^i + i_i \odot \phi(W_x^c x_i + \sum_{n=1}^{N_i} W_n^c h_n^i) \tag{10}$$

$$h_i = o_t \odot \psi(c_t) \tag{11}$$

where N_i is the number of child nodes for node i and h_n^i is the hidden state from its n-th child. Each leaf node is sequentially mapped to the words of the input sentence according to the parsing tree. During the backward propagation, the errors are back propagated from parent node to all of its child nodes[1]. Existing work of using T-LSTM for sentiment analysis are limited to textual analysis only. We will explain in the next section on how to generalize it to multimedia context.

A key challenge of using both image and text in T-LSTM is that one modality will dominate the model so that the resulted model's performance will be similar to the one with single modality. To solve this problem, we extend the attention model [1] using bilinear attention model. Different from the previous works in attention model [39] for image captioning. However, The model in [39] is based on sequential LSTM while this work is based on tree-structure LSTM. Another main difference between [39] and this work is that we use a bilinear model for attention, which is more effective in practice. Last but not the least, this paper studies the problem in a discriminative setting, while the method in [39] is based on a generative model, which is not applicable for sentiment estimation. The next section will explain our model in details.

4. PROPOSED SCHEME FOR SENTIMENT ANALYSIS

[1]We refer the readers to [44] for the detailed formulas of back propagation.

In this section, we present how to apply the tree-structured T-LSTM model for joint visual-textual sentiment analysis. Our approach is mainly motivated by the findings that tree-structured models can benefit vision tasks [29, 40]. Our system incorporates both the attention mechanism and an auxiliary semantic learning task to learn a robust joint visual-textual semantic representation. Consequently, we are able to learn a more accurate and robust sentiment classifier.

4.1 T-LSTM for Joint Sentiment Analysis

T-LSTM relies on tree-structures for learning and testing. Inspired by both [33] and [44], we also learn the T-LSTM over a parsing tree, which encodes the syntactic structure of a given sentence.

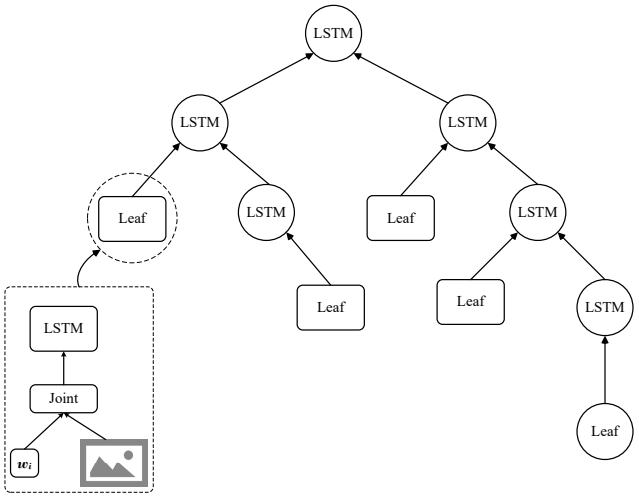


Figure 3: The scheme of the joint visual-textual tree-structured LSTM. We produce each *leaf* node (rectangle node) from a joint module. The joint module has two inputs, a word and regions of an image. Each internal node can receive information from multiple children nodes. The information is processed using a LSTM cell. The output of each node is forwarded to its parent node.

Most of previous works on T-LSTM are based on text (single modality). Next, we will explain how to generalize it to multi-modality scenario where a pair of sentence and image (t^m, v^m) are taken into account jointly. We take the hidden state at the root node of the T-LSTM as the representation for the sentence and image pair. Next, this representation can be supplied as input to build a softmax classifier for sentiment analysis. In such a way, we can solve the problem of joint visual-textual sentiment analysis. We employ the negative log-likelihood (NLL) to define the cost:

$$p(h[t^m, v^m]) = \text{softmax}\left(W_s h[t^m, v^m]\right) \tag{12}$$

$$L(t^m, v^m) = -\log\left(p(h[t^m, v^m]), l_m\right) \tag{13}$$

where $h[t^m, v^m]$ is the hidden state at the root node of a T-LSTM, W_s is the parameters for a linear model, and l_m is the sentiment label for the m-th image and sentence pair. The overall network can be trained using back propagation.

To learn and establish the joint representation $h[t^m, v^m]$ of a given pair of sentence and image (t^m, v^m), we focus on the leaf nodes, which directly accept textual words and visual representations as inputs. In particular, we want the *leaf* nodes of T-LSTM to jointly accept both individual words and image regions and produce its output based on both inputs. Eventually, the root node of the parsing tree will learn a joint embedding by receiving both the visual and the textual information propagated structurally from the *leaf* nodes. In such a way, we are able to integrate the visual information into the tree-structures. Figure 3 shows the overall framework of the proposed scheme. Each internal node is an LSTM memory cell. Meanwhile, each leaf node is a joint model, which tries to produce outputs from the input word and the input image regions.

A key challenge of introducing both image and text into T-LSTM is that one modality will dominate so that the resulted model's performance will be similar to the one with single modality. In the following subsection, we use a bilinear attention model as the joint module to learn the alignments and produce the outputs of leaf nodes simultaneously.

4.2 Bilinear Attention Model

Given a image and one descriptive sentence of the image, we assume that words of the sentence are likely associated with some regions in the image. Our goal is to automatically find such kind of connections between the words and the image regions. Let $T = \{t_1, t_2, \ldots, t_m\}$ denote a sentence with m words and let $V = \{v_1, v_2, \ldots, v_n\}$ denote the regions of an image, and n is the number of image regions. In attention model, for each word t_i, a score α_{ij} $(1 \leq j \leq n)$ is assigned to each image region v_j based on its relevance with the content of v_j. As a common approach to model relevance in vector space, a bilinear function is used to evaluate α_{ij}:

$$\alpha_{ij} \propto \varphi\left(t_i^T U v_j\right), \tag{14}$$

where the α_i.s are taken to normalize over all the $\{v_j\}$, $\varphi(\cdot)$ is a smooth function, and U is the weight matrix to be learned. One popular choice for $\varphi(\cdot)$ is the $\exp(\cdot)$ as in the softmax function.

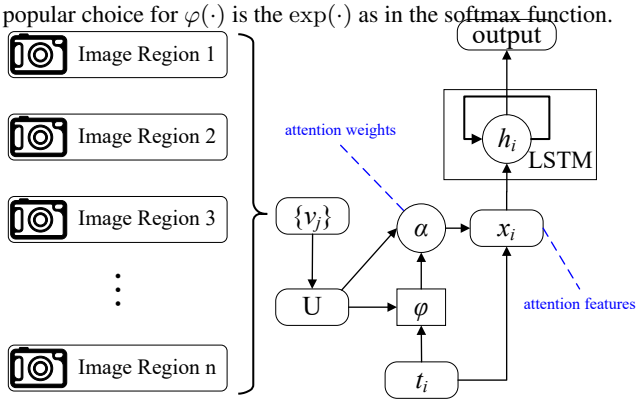


Figure 4: The bilinear attention model architecture for the leaf node. The attention model produces x_i, which is the concatenation of t_i and weighted sum of mappings of $\{v_j\}$. x_i is provided to a LSTM cell to produce the output for its parent node.

Once calculated, the attention scores are used to modulate the strength of attention on different image regions. The weighted sum of all candidate regions is mapped from visual feature space to the input word space of t_i:

$$v^i = \sum_{k=1}^{n} \alpha_{ik}(U v_k). \tag{15}$$

We obtain a weighted visual feature mapping v^i for the current word t_i. Let $x_i = [v^i; t_i]$ be the concatenation of v^i and t_i, which is the output produced by the attention model and supplied as the input to an LSTM memory cell (see Eq. (2) to Eq. (6)). In this way, we are able to integrate the visual information into the T-LSTM model. The whole bilinear attention model architecture for each leaf node is illustrated in Figure 4.

4.3 Semantic Embedding Learning

We hope that the attention model is able to assign *correct* attention weights between the input words and the feature regions for a given image and sentence pair. However, a sentiment analysis classifier mainly focuses on the performance of the joint features on sentiment analysis. There is no mechanism that explicitly helps the learning of alignments or correspondences between words and image regions. In other words, the attention model may not learn to assign semantically similar image regions to the corresponding words by optimizing on the gradients passed down from a sentiment classifier. Instead, a more secure approach is to explicitly utilize another task which will semantically learn the mapping or correspondence between words and image regions.

Inspired by the recent successes of deep visual-textual semantic embedding learning [14, 15, 19], we incorporate the semantic learning task to pilot the learning of attention model. Let (t^m,v^m) be a sentence and image pair, and v^n (randomly picked from the training set) be a contrasting image of t^m. We then use the previously introduced T-LSTM model with the bilinear attention mechanism to encode both (t^m,v^m) and (t^m,v^n). Next, the pairwise margin ranking function is optimized to learn the semantic embedding:

$$L'(t^m, v^m, v^n) = \max(0, \mu - g(h[t^m, v^m]) + g(h[t^m, v^n])) \tag{16}$$

where $g(\cdot)$ learns the embedding score given the hidden features $h[t^m, v^m]$ and $h[t^m, v^n]$ from T-LSTM. This objective function tries to make sure that the score defined by $g(\cdot)$ is at least greater than μ for correct pair (t^m, v^m) compared to the contrastive pair (t^m, v^n).

Similar to [19], we use a multi-layer perceptron (MLP) to learn the score of each sentence and image pair given their hidden state. In particular, we define $g(\cdot)$ as follows:

$$g(h_i) = W_2^e(\delta(W_1^e h_i)) \tag{17}$$

where W_2^e and W_1^e are the parameters of the MLP and $\delta(\cdot)$ is the activation function for the hidden state. In our implementation, we use $\tanh(\cdot)$ as the activation function.

Another observation is that the calculation of attention weights α in Eq.(14) is inappropriate for learning semantic embedding. Recall the attention weights α are non-negative. It is assumed that the sentence t^m is supposed to describe the content of the image v^m. With this assumption, the attention model tries to discover the correspondence between the text words of t_i^m and the image regions of v_j^m. Therefore, softmax function is chosen as the smooth function for $\varphi(\cdot)$ (Eq.(14)). However, in the semantic learning task, we have randomly picked a negative image v^n, whose content may be totally irrelevant of t^m. Therefore, the attention weights defined in Eq.(14) could mislead the attention model given an unrelated image and sentence pair v^n and t^m as training data. To avoid this kind of *misleading or false* attention, we introduce the following way to calculate the attention weights (for simplicity, we ignore the superscript):

$$\beta_{ij} = \max(0, t_i^T U v_j) \tag{18}$$

$$\alpha_{ij} = \phi(\beta_{ij}) \tag{19}$$

where ϕ is the smoothing function. We use $\tanh(\cdot)$ to smooth α instead of $\exp(\cdot)$ as in Eq.(14). In such a way, we intend to achieve the following: 1) negative correspondences are ignored (Eq.(18)) and 2) α falls into reasonable value ranges (Eq.(19)).

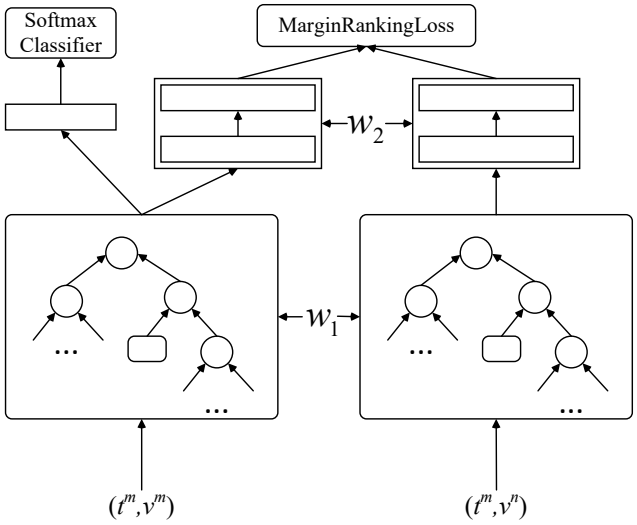


Figure 5: A multi-task learning framework for jointly training the sentiment classifier as well as the semantic embedding. (t^m, v^m) is the correct sentence and image pair and v^n is a randomly picked image from the training set. W_1 and W_2 are shared parameters for T-LSTM and the scoring MLP, respectively.

4.4 Multi-task Learning for Joint Sentiment Analysis

Our model can be trained by a simultaneous optimization of both the sentiment analysis and the embedding tasks. Figure 5 shows the overall structure of the proposed model with the two tasks. The semantic embedding learning task is implemented using a Siamese network, where the parameters of T-LSTM (W_1 in Figure 5) and the parameters of the scoring MLP (W_2 in Figure 5) are shared. The loss of one training image-sentence pair is the summation of the sentiment classifier loss (Eq. (13)) and the margin ranking loss (Eq.(16)). For each training pair (t^m, v^m), we randomly choose one image v^n to make a contrasting pair (t^m, v^n), which is also given as the input to the semantic embedding module. The overall loss function is

$$L_{mt} = L(t^m, v^m) + L'(t^m, v^m, v^n). \tag{20}$$

All the parameters are automatically learned by minimizing the two loss functions over a training set. We use a mini-batch gradient descent algorithm with an adaptive learning rate to optimize the loss functions.

5. EXPERIMENTS

To the best of our knowledge, the work in [42] was the first study on joint textual-visual sentiment analysis using deep neural features. They also collected three data sets from GettyImage and Twitter, respectively, to evaluate their cross-modality consistent regression (CCR) model. We focus on comparing our proposed models with the baselines in [42]. In addition, we also add two base-

Table 1: Summary of the models included in our evaluation. The first five models are from [42]. The last three models are our proposed models.

Model	Description
Single Visual Model	Logistic regression on deep visual features from pre-trained CaffeNet model.
Single Textual Model	Logistic regression on paragraph feature vectors [17] of text.
Early Fusion	Logistic regression on concatenated visual (CaffeNet) and textual features.
Late Fusion	Average of logistic regression sentiment score on both visual and textual features.
CCR	Cross-modality consistent regression (CCR) [42]
T-LSTM	Using T-LSTM model on the text only without attention.
T-LSTM Attention	Using T-LSTM model with attention model between text words and image regions.
T-LSTM Embedding	Learn the T-LSTM model and the semantic embedding simultaneously.

Table 2: Statistics of the four data sets.

Data Set	Positive	Negative	Total
Getty	311,940	276,281	588,221
Twitter	16,884	14,700	31,584
Twitter AMT	389	224	613
VSO-VT	129,524	124,993	254,517

lines 1) T-LSTM on text only, and 2) T-LSTM on both text and image but without the semantic embedding task. Table 1 summarizes the different joint models for sentiment analysis. We also provide a short description for each model.

5.1 Data Sets

To test our proposed algorithms, we include the three datasets from previous work [42] and additionally build a new dataset for joint textual-visual sentiment analysis . We will briefly describe the several data sets. Next, we will compare the performance results of the proposed approach with other state-of-the-art approaches.

The first data set is from Getty Image. It was built by querying the Getty image search engine with different sentiment keywords. The authors were able to collect a large data set containing about $588,000$ sentence-image pairs. Possibly, the data sets is noisy. However, the noise is tolerable due to the relatively formal and clean descriptions of images as argued in [42]. Meanwhile, since we use the same dataset, the noise is fair to all the candidate algorithms.

The second data set is from Twitter. More specifically, it contains image tweets (tweet messages that contain images). The dataset is relatively small after pre-processing. In total, it has $31,584$ *weakly* labeled image tweets. This dataset is also considered *weakly* labeled in that all the labels were generated by a predefined rule-based sentiment classifier. The last one is a small image-tweet dataset labeled by Amazon Mechanical Turk. In total, there are only 613 image tweets.

In addition, we build another weakly labeled dataset for evaluation. This new dataset is built on top of the visual sentiment ontology (VSO) [3], which consists of millions of images collected by querying Flickr with thousands of adjective and noun pairs (ANPs). Each ANP has hundreds of images collected from Flickr. However, this dataset only has the URLs of the images. There is no description for each image. Fortunately, Flickr has provided the API[2], which enables us to obtain the metadata (descriptions, upload date, tags, and so on) of an image by supplying its unique ID. Therefore, we are able to build a dataset for joint textual-visual sentiment analysis by collecting their Flickr descriptions using the provided API. After removing the invalid images that no longer exist, and eliminating images with too long (more than 100 words) and too short (less than 5 words) descriptions, we obtain $254,517$ images. Table 2 summarizes the statistics of the four datasets, where VSO-VT is the newly built dataset by us on top of the visual sentiment ontology.

[2]https://www.flickr.com/services/api/

5.2 Experimental Settings

To build a T-LSTM model, first we need to build a tree structure. In this work, we use the semantic constituency parsing tree, which was employed in [33] to build a tree-structured LSTM. We also employ the Stanford Parser[3], which is one of the state-of-the-art parsers, to build the parsing tree for each sentence.

Next, we need to choose feature representations for textual words and images. For word representation, there are two popular approaches. The first is one-hot representation with an embedding layer. In particular, the goal is to map a word w_i with representation $w_i = [0, \ldots, 1_i, \ldots, 0] \in R^{|V|}$ (only the i-th position is one in the one-hot representation) to $e_i \in R^m$, where $|V|$ is the size of the vocabulary and m is the size of embedding layer. The second approach is to directly employ the pre-trained distributed representations of words, such as Word2Vec [21] and GloVe [25]. Similar to [33], we use the pre-trained 300-dimensional GloVe features to represent words. This is particular helpful when the vocabulary size is too large, where insufficient text data may not lead to well learned word features in the one-hot representation setting.

In the visual part, Convolutional Neural Networks (CNN) have been widely used for robust visual feature extraction. In particular, features extracted from the pre-trained models on the ImageNet (http://www.image-net.org) dataset have succeeded in many visual related tasks. More recently, convolutional layer features are being studied for visual representations as well. Following [39], which employed convolutional layer features to learn an attention model for image caption generation, we use the same convolutional layer (conv5_4) in pre-trained VGG-19 model [27] to extract visual features for an input image. The feature size of the convolutional layer is 196×512. In other words, for each image, we have a total of 196 image candidate regions for the attention model.

The model is trained on GPU machines. We use the same split of [42] to make a fair comparison[4]. A separate validation dataset is used to select hyper-parameters and to control the stopping criteria. We train the model in a mini-batch mode, where 100 text-image pairs are randomly selected per batch. The hidden layer size is 512. All the source codes of the T-LSTM models will be released upon the publication of this work.

5.3 Experimental Results

[3]http://nlp.stanford.edu/software/lex-parser.shtml

[4]The splits are not publicly available. We obtain the splits by personal communication with the authors.

Table 3: Results of different T-LSTM variants and previously reported results on the Getty testing dataset. The first five models are from [42]. The last three models are the newly proposed models (see Table 1 for the details of the models).

Model	Precision	Recall	F1	Accuracy
Textual	0.806	0.544	0.655	0.696
Visual	0.747	0.745	0.746	0.732
Early Fusion	0.778	0.769	0.774	0.763
Late Fusion	0.785	0.775	0.780	0.769
CCR	0.846	0.759	0.800	0.800
T-LSTM	0.872	0.884	0.872	0.878
T-LSTM Attention	0.872	0.886	0.874	0.879
T-LSTM Embedding	**0.895**	**0.919**	**0.907**	**0.902**

Table 4: Results of different T-LSTM variants and previously reported results on the Twitter testing dataset.

Model	Precision	Recall	F1	Accuracy
Textual	0.746	0.693	0.727	0.722
Visual	0.584	0.561	0.573	0.553
Early Fusion	0.730	0.744	0.737	0.717
Late Fusion	0.634	0.610	0.622	0.604
CCR	0.831	0.805	0.818	0.809
T-LSTM	0.955	0.971	0.963	0.960
T-LSTM Attention	0.952	0.968	0.960	0.956
T-LSTM Embedding	**0.958**	**0.977**	**0.967**	**0.964**

5.3.1 *Results on the Getty testing dataset*

For all the T-LSTM variants, we use the same parameter settings. Table 3 shows the experimental results on the testing data set from Getty Images. T-LSTM model has significantly improve the performance using textual features only. This also coincides with the results in [33]. Indeed, the work in [42] mainly focus on the learning of a robust classifier. The textual features are learned in an unsupervised way and the visual features are extracted from pre-trained CaffeNet models. Both of them are fixed in the learning of a sentiment classifier. This explains why T-LSTM can significantly improve the performance, where feature mappings and structures are also part of the model to learn.

T-LSTM with the attention model on images slightly improve the performance compared with T-LSTM. However, its peer model T-LSTM Embedding shows the best performance on all metrics. We attribute the difference to the attention model. It suggests that an auxiliary semantic learning task could lead to a better attention mechanism and thus a better sentiment classifier. We will quantitatively analyze the effect of attention in Section 5.4.

5.3.2 *Results on two Twitter datasets*

In addition, we test the proposed models on the Twitter dataset. Table 4 summarizes the results. Similarly, all the T-LSTM variants have shown significant improvements over previously reported algorithms. The accuracies of all the models are over 95%. Since T-LSTM has achieved much better results alone, the performance gain of adding visual features seems marginal. In fact, using the attention mechanism without the embedding learning task leads to slight degradation of the performance. We believe this performance is highly related to the way how ground truth labels are collected for this Twitter dataset. The authors [42] have indicated that they collect the weak or noisy sentiment labels of the image tweets by analyzing the Tweets text only using VADER [13]. Because VADER is a rule-based classifier for Tweets, it is acceptable to analyze the performance of sentiment classifiers, which is built on top of content-based features [42]. In contrast, T-LSTM tries to employ the dependency tree structures, which may have unobserved information overlap with the rules in VADER [13]. This explains the significant performance improvements of T-LSTM over previous models on this dataset.

There is another small Twitter AMT dataset, which was labeled by Amazon Mechanical Turk (AMT). This can reduce the bias induced by using another classifier to collect weak labels. This dataset is pretty small, with only 613 image tweets. We only use it as testing data to further evaluate the generability of the three T-LSTM models. In particular, we evaluate the previously trained models on the weakly labeled Twitter dataset on this small dataset without further training or fine-tuning. All the results are shown in Table 5. Both T-LSTM Attention and T-LSTM Embedding have shown better performance than T-LSTM, which indicates that the inclusion of visual features could lead to a more general model. In particular, joint embedding learning also promotes the performance of the learned model. All three T-LSTM models have shown better performance than the previous approaches, where the models trained on the weakly labeled Twitter dataset are further fine-tuned on this small dataset. Since this dataset is labeled by AMT workers, the performance is relatively worse than the performance on the weakly labeled Twitter testing dataset in Table 4.

Table 5: Results of different T-LSTM variants and previously reported results on the Twitter testing dataset labeled by Amazon Mechanical Turk workers.

Model	Precision	Recall	F1	Accuracy
Textual	0.832	0.638	0.722	0.688
Visual	0.762	0.715	0.737	0.677
Early Fusion	0.776	0.740	0.758	0.700
Late Fusion	0.799	0.738	0.767	0.716
CCR	0.846	0.759	0.800	0.800
T-LSTM	**1.000**	0.807	0.893	0.878
T-LSTM Attention	**1.000**	**0.830**	0.907	0.892
T-LSTM Embedding	**1.000**	**0.848**	**0.918**	**0.904**

Table 6: Results of different T-LSTM variants and previously reported results on the newly collected VSO-VT dataset.

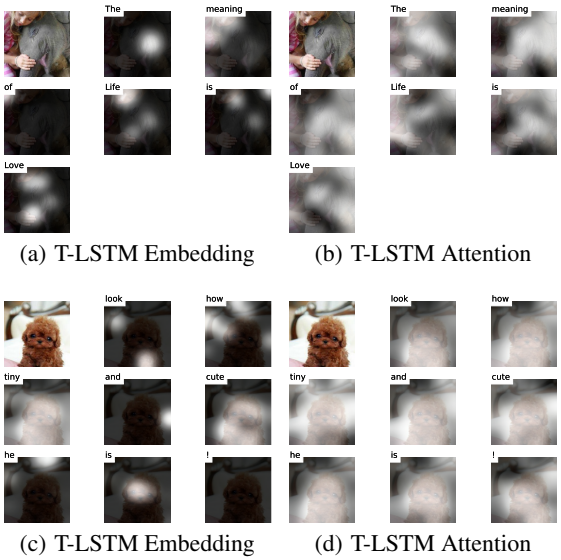
Model	Precision	Recall	F1	Accuracy
Textual	0.626	0.622	0.624	0.624
Visual	0.625	0.586	0.605	0.616
Early Fusion	0.616	0.646	0.631	0.621
Late Fusion	0.660	0.629	0.645	0.650
CCR	0.672	0.678	0.675	0.672
T-LSTM	0.806	0.803	0.805	0.804
T-LSTM Attention	0.816	0.810	0.813	0.813
T-LSTM Embedding	**0.821**	**0.833**	**0.833**	**0.833**

5.3.3 *Results on VSO-VT*

Table 6 shows the performances of different algorithms on the constructed VSO-VT dataset. In particular, we implement and configure the first five models following the descriptions in [42]. Because this newly build dataset is pretty noisy[5], all of the models

[5]Recall we use the sentiment of an ANP to label all the images belonging to that ANP.

have relatively worse performance compared with the results on Getty and Twitter test datasets. However, T-LSTM variants still shows better performance over CCR and other baseline algorithms. Meanwhile, T-LSTM Emb has consistently demonstrated the best performance by all metrics.



(a) T-LSTM Embedding (b) T-LSTM Attention

(c) T-LSTM Embedding (d) T-LSTM Attention

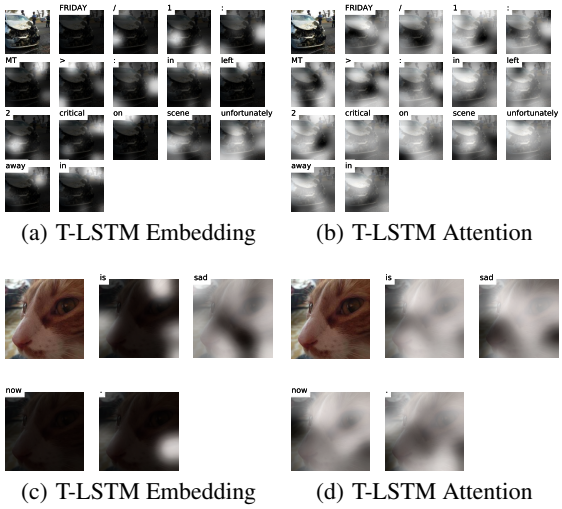
Figure 6: Visualization of *attention* on two examples with positive sentiment.

5.4 Qualitative Attention Analysis

We also try to visualize the attention weights at the leaf nodes of both T-LSTM Attention and T-LSTM Embedding. In particular, Xu *et al.* [39] have employed upsampling and Gaussian filtering to visualize attention weights. In this section, we follow the same steps to visualize the attention weights of both models. Recall that the Twitter AMT dataset is labeled by humans (Amazon Mechanical Turk workers). Therefore, we choose examples from these 613 image Tweets for illustration.

Figure 6 and Figure 7 show several positive and negative examples for T-LSTM Attention and T-LSTM Embedding, respectively. Overall, the T-LSTM Embedding model tends to learn more accurate attention than T-LSTM Attention. This indicates that the auxiliary embedding learning task helps the model to align words and image regions. Furthermore, the sentiment analysis task allows the attention mechanism to focus more on sentiment related regions. For instance, positive words *love* in Figure 6(a), *tiny* and *cute* in Figure 6(c) have all attended on the most related image regions. This is also true for *unfortunately* in Figure 7(a) and *sad* in Figure 7(c) of the two negative examples.

Another qualitative analysis is to check the top-confidence examples by different models [41, 42]. Even though the dataset is noisily labeled, Figure 8 shows the top ranked examples (according to the prediction score) of all the evaluated models on the VSO-VT dataset. Top ranked examples are quite different from model to model. For top ranked positives, T-LSTM Embedding favors images including people, while T-LSTM Attention model prefers more colorful objects, such as flowers and yummy food. T-LSTM's top ranked examples are more likely to include strong emotional words, such as "well", "amazing", "stunning" and "super", which also retrieves content-related images. Overall, CCR prefers images with long descriptions. This may be due to the fact that long



(a) T-LSTM Embedding (b) T-LSTM Attention

(c) T-LSTM Embedding (d) T-LSTM Attention

Figure 7: Visualization of *attention* on two examples with negative sentiment.

sentences could lead to better learned textual representations in unsupervised distributed learning. For top ranked negative examples, both T-LSTM Embedding and CCR try to return abandoned and depressed scenes. Overall, T-LSTM Embedding provides the most reasonable top ranked examples.

6. CONCLUSIONS

In this study, we present a new end-to-end framework for joint visual-textual sentiment analysis. Our system tries to integrate textual and visual information in a structured fashion. Our model also incorporates an attention mechanism in a tree-structured LSTM to learn the alignments or correspondences between image regions and descriptive words. Therefore, the tree-structured model is capable of propagating the information from children nodes to their parents nodes in a bottom-up fashion. Later, joint features are obtained at the root nodes of the trees and supplied to a multi-layer perceptron for training a sentiment classifier. Meanwhile, we also introduce an auxiliary task, visual-textual semantic embedding, to help the learning of the attention model. Extensive experimental results have demonstrated that the proposed joint models have significantly improved the performance of joint textual-visual sentiment analysis on several datasets. In particular, the visual-textual semantic embedding task leads to better attention and in turn a better joint sentiment classifier.

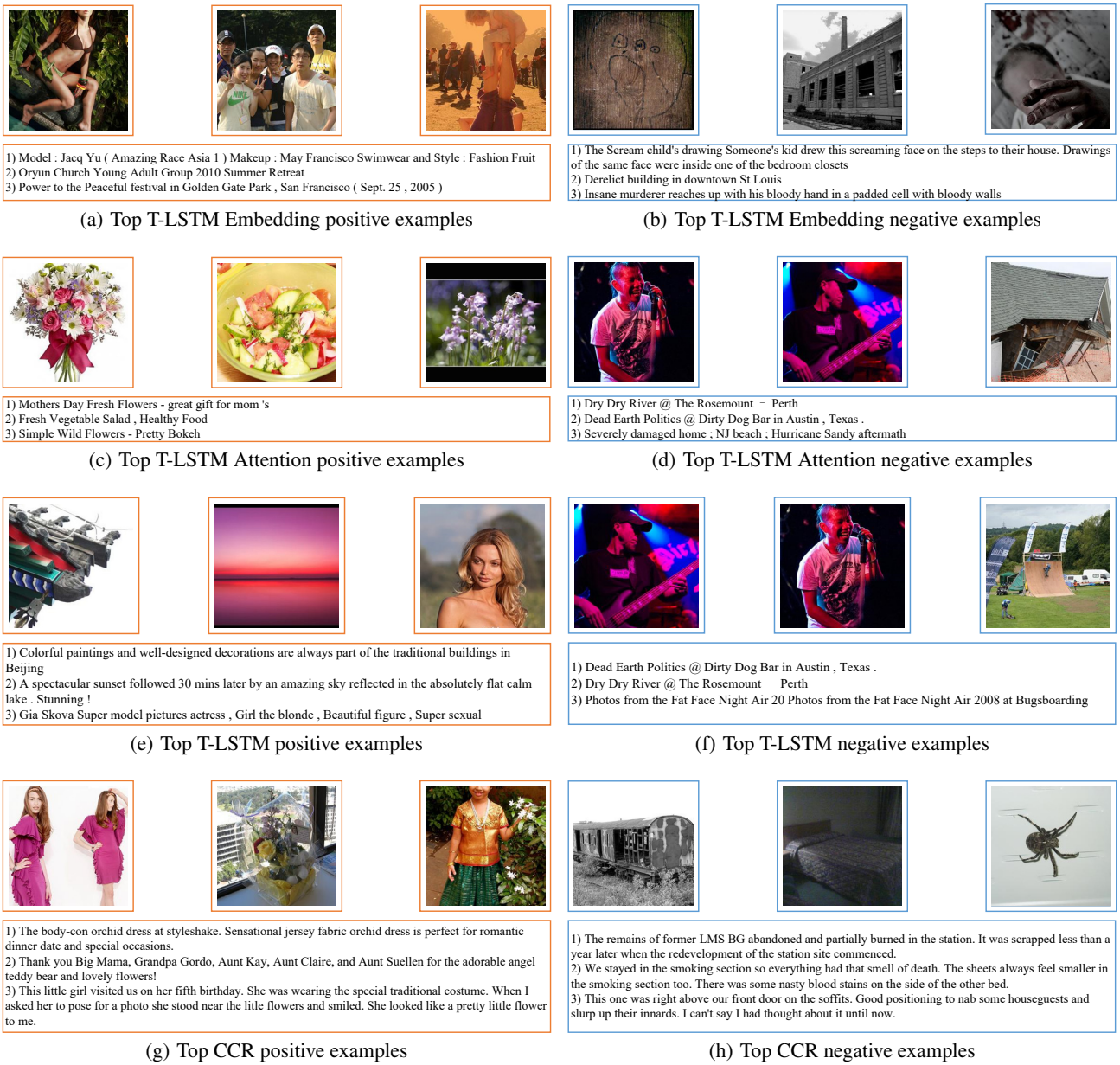
Acknowledgment

This work was generously supported in part by Adobe Research and New York State through the Goergen Institute for Data Science at the University of Rochester.

7. REFERENCES

[1] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. In *ICLR 2015*, 2014.

[2] D. Borth, T. Chen, R. Ji, and S.-F. Chang. Sentibank: large-scale ontology and classifiers for detecting sentiment



(a) Top T-LSTM Embedding positive examples

(b) Top T-LSTM Embedding negative examples

(c) Top T-LSTM Attention positive examples

(d) Top T-LSTM Attention negative examples

(e) Top T-LSTM positive examples

(f) Top T-LSTM negative examples

(g) Top CCR positive examples

(h) Top CCR negative examples

Figure 8: Top three ranked examples on the VSO-VT dataset by different approaches according to the prediction scores. The three descriptions corresponds to the three images from left to right in the same group.

and emotions in visual content. In *ACM MM*, pages 459–460. ACM, 2013.

[3] D. Borth, R. Ji, T. Chen, T. M. Breuel, and S. Chang. Large-scale visual sentiment ontology and detectors using adjective noun pairs. In *ACM MM*, pages 223–232, 2013.

[4] D. Cao, R. Ji, D. Lin, and S. Li. A cross-media public sentiment analysis system for microblog. *Multimedia Systems*, pages 1–8, 2014.

[5] X. Chen and C. Lawrence Zitnick. Mind's eye: A recurrent visual representation for image caption generation. In *CVPR*, June 2015.

[6] H. Fang, S. Gupta, F. Iandola, R. K. Srivastava, L. Deng, P. Dollar, J. Gao, X. He, M. Mitchell, J. C. Platt, C. Lawrence Zitnick, and G. Zweig. From captions to visual concepts and back. In *CVPR*, June 2015.

[7] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *NIPS*, pages 2121–2129, 2013.

[8] F. Gers. Long short-term memory in recurrent neural networks. *Unpublished PhD dissertation, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland*, 2001.

[9] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with lstm recurrent networks. *The Journal of*

Machine Learning Research, 3:115–143, 2003.

[10] A. Graves, N. Jaitly, and A.-R. Mohamed. Hybrid speech recognition with deep bidirectional lstm. In *Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 273–278. IEEE, 2013.

[11] A. Graves, A.-r. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *ICASSP*, pages 6645–6649. IEEE, 2013.

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[13] C. J. Hutto and E. Gilbert. VADER: A parsimonious rule-based model for sentiment analysis of social media text. In *ICWSM*, 2014.

[14] A. Karpathy and F. Li. Deep visual-semantic alignments for generating image descriptions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3128–3137, 2015.

[15] R. Kiros, R. Salakhutdinov, and R. S. Zemel. Unifying visual-semantic embeddings with multimodal neural language models. *CoRR*, abs/1411.2539, 2014.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1106–1114, 2012.

[17] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.

[18] X. Lu, Z. Lin, H. Jin, J. Yang, and J. Z. Wang. Rapid: Rating pictorial aesthetics using deep learning. In *Proceedings of the ACM International Conference on Multimedia (ACM MM)*, pages 457–466. ACM, 2014.

[19] L. Ma, Z. Lu, L. Shang, and H. Li. Multimodal convolutional neural networks for matching image and sentence. In *ICCV*, December 2015.

[20] J. Mao, W. Xu, Y. Yang, J. Wang, Z. Huang, and A. Yuille. Deep captioning with multimodal recurrent neural networks (m-rnn). *ICLR*, 2015.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26 (NIPS)*, pages 3111–3119, 2013.

[22] S. Nowozin and C. H. Lampert. Structured learning and prediction in computer vision. *Foundations and Trends in Computer Graphics and Vision*, 6(3-4):185–365, 2011.

[23] B. Pang and L. Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[24] R. Pascanu, T. Mikolov, and Y. Bengio. On the difficulty of training recurrent neural networks. In *ICML*, pages 1310–1318, 2013.

[25] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[26] S. Siersdorfer, E. Minack, F. Deng, and J. S. Hare. Analyzing and predicting sentiment of images on the social web. In *Proceedings of the 18th International Conference on Multimedia (ACM MM)*, pages 715–718, 2010.

[27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[28] R. Socher, A. Karpathy, Q. V. Le, C. D. Manning, and A. Y. Ng. Grounded compositional semantics for finding and describing images with sentences. *TACL*, 2:207–218, 2014.

[29] R. Socher, C. C. Lin, A. Y. Ng, and C. D. Manning. Parsing natural scenes and natural language with recursive neural networks. In *ICML*, pages 129–136, 2011.

[30] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. *Journal of Machine Learning Research*, 15(1):2949–2980, 2014.

[31] I. Sutskever. *Training recurrent neural networks*. PhD thesis, University of Toronto, 2013.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–9, 2015.

[33] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1556–1566, July 2015.

[34] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *CVPR*, June 2015.

[35] M. Wang, D. Cao, L. Li, S. Li, and R. Ji. Microblog sentiment analysis based on cross-media bag-of-words model. In *ICIMCS*, pages 76:76–76:80. ACM, 2014.

[36] Y. Wang, Y. Hu, S. Kambhampati, and B. Li. Inferring sentiment from web images with joint inference on visual and social cues: A regulated matrix factorization approach. In *ICWSM*, pages 473–482, 2015.

[37] Y. Wang, S. Wang, J. Tang, H. Liu, and B. Li. Unsupervised sentiment analysis for social media images. In *IJCAI*, pages 2378–2379, 2015.

[38] J. Weston, S. Bengio, and N. Usunier. WSABIE: scaling up to large vocabulary image annotation. In *IJCAI*, pages 2764–2770, 2011.

[39] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, and Y. Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *ICML*, pages 2048–2057, 2015.

[40] Y. Yang and D. Ramanan. Articulated pose estimation with flexible mixtures-of-parts. In *CVPR*, pages 1385–1392, June 2011.

[41] Q. You, J. Luo, H. Jin, and J. Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *AAAI*, pages 381–388, 2015.

[42] Q. You, J. Luo, H. Jin, and J. Yang. Cross-modality consistent regression for joint visual-textual sentiment analysis of social multimedia. In *ACM International Conference on Web Search and Data Ming (WSDM)*, pages 13–22, 2016.

[43] J. Yuan, S. Mcdonough, Q. You, and J. Luo. Sentribute: image sentiment analysis from a mid-level perspective. In *WISDOM*, page 10, 2013.

[44] X. Zhu, P. Sobhani, and H. Guo. Long short-term memory over recursive structures. In *ICML*, pages 1604–1612, 2015.