Cross-modality Consistent Regression for Joint Visual-Textual Sentiment Analysis of Social Multimedia

Quanzeng You and Jiebo Luo
University of Rochester
Rochester, NY 14623
{qyou, jluo}@cs.rochester.edu

Hailin Jin
Adobe Research
San Jose, CA 95110
hljin@adobe.com

Jianchao Yang
Snapchat Inc
Venice, CA 90291
jianchao.yang@snapchat.com

ABSTRACT

Sentiment analysis of online user generated content is important for many social media analytics tasks. Researchers have largely relied on textual sentiment analysis to develop systems to predict political elections, measure economic indicators, and so on. Recently, social media users are increasingly using additional images and videos to express their opinions and share their experiences. Sentiment analysis of such large-scale textual and visual content can help better extract user sentiments toward events or topics. Motivated by the needs to leverage large-scale social multimedia content for sentiment analysis, we propose a cross-modality consistent regression (CCR) model, which is able to utilize both the state-of-the-art visual and textual sentiment analysis techniques. We first fine-tune a convolutional neural network (CNN) for image sentiment analysis and train a paragraph vector model for textual sentiment analysis. On top of them, we train our multi-modality regression model. We use sentimental queries to obtain half a million training samples from Getty Images. We have conducted extensive experiments on both machine weakly labeled and manually labeled image tweets. The results show that the proposed model can achieve better performance than the state-of-the-art textual and visual sentiment analysis algorithms alone.

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Vision and Scene Understanding; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision*

Keywords

sentiment analysis, cross-modality regression, multimodality analysis

1. INTRODUCTION

The increasing popularity of social networks attracts more and more people to share their experiences and to express their opinions on virtually all events and subjects in online social network platforms. Each day, billions of messages and posts are generated. In this study, we focus on deriving people's opinions or sentiments towards topics and events happening in real world. In other words, we are interested in automatical detection of sentiment from online user generated content.

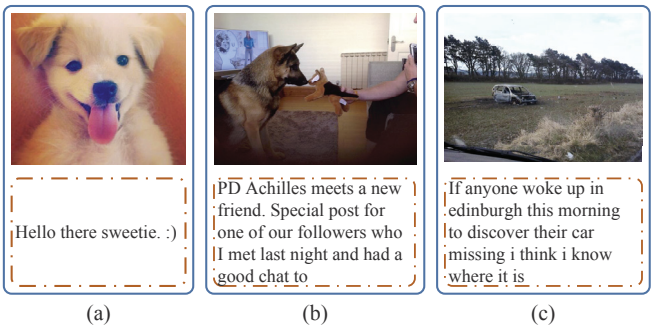


Figure 1: Examples of image tweets from Twitter.

Figure 1 shows several example image tweets from Twitter. Image tweets refer to those tweets that contain images. If we take a look at these three example image tweets, we can observe that in example (a), both image and the text indicate that this tweet carries a positive sentiment; in (b) while it is difficult to tell the sentiment from the image in the *middle* image tweet, however, we can tell that this tweet expresses positive sentiment from the text; in (c) on the contrary, it is hard to tell the sentiment from the text, however the worn-out car in the image suggest an overall negative sentiment. These examples explain the motivation for our work. We would like to learn people's overall sentiment over the same object from different modalities of the object provided by the user. In particular, we focus on inferring people's sentiment according to the available images and the short and informal text.

Many researchers have contributed to sentiment analysis. For instance, there are related works on detecting users' sentiment and applying sentiment analysis to predict box-office revenues for movies [1], political elections [23, 29] and economic indicators [3, 35]. In particular, recently published works started to focus on analyzing sentiment of informally user generated content from online social networks. However, current techniques are mostly based on the analysis of textual content to detect sentiment. On the other hand,

WSDM'16, February 22–25, 2016, San Francisco, CA, USA.

DOI: http://dx.doi.org/10.1145/2835776.2835779

visual content, including both images and videos, are becoming increasingly popular in all mainstream online social network platforms. For example, Twitter's support of image tweets and Vine as well as Facebook's Instagram are all designed to support people to share and post more visual content. More interestingly, statistics show that the usage of image in a tweet is able to increase the popularity of this tweet in terms of clicks , retweets, and favorites[1]. This can encourage Twitter users to post more visual content. We cannot ignore the prevalently available visual content in analyzing online users' sentiment.

To the best of our knowledge, little attention has been paid to the sentiment analysis of visual content as well as multi-modality sentiment analysis. Only a few recent works attempted to predict visual sentiment using features from images [25, 5, 4, 34] and videos [21]. Visual sentiment analysis is extremely challenging, as image sentiment involves a much higher level of abstraction and subjectivity in the human recognition process [16], on top of a wide variety of visual recognition tasks including object, scene, action and event recognition. However, Convolutional Neural Networks [19, 7, 17] have been proved to be very powerful in solving computer vision related tasks. Due to the challenges of visual sentiment analysis, we propose a multi-modality framework to analyze sentiment on top of state-of-the-art techniques in both visual and textual analysis.

To that end, we address in this work two major challenges as follows: 1) we propose a novel multi-modality regression model, which can integrate different modality features for sentiment analysis, and 2) we demonstrate the feasibility of using weakly labelled data for multi-modal sentiment analysis and how to easily transfer models from one domain to another domain. The contributions of this paper include

- We employ the state-of-the-art machine learning algorithms to a solve a challenging novel problem, multi-modality sentiment analysis. In particular, we adopt Convolutional Neural Networks [17, 32] to visual sentiment analysis and employ the state-of-the-art distributed representation of documents [18] for textual sentiment analysis.

- We propose a novel multi-modality regression model, CCR, which tries to impose consistent constraints across related but different modalities. The formulation of the model is simple yet generalizable and can be easily implemented. The analysis and experimental results on sentiment analysis validate the effectiveness of the proposed model.

- Our model can be trained on a large scale data set in a mini-batch mode. In particular, we show that it is possible to employ *weakly* labeled data to learn models for highly abstract tasks, such as sentiment analysis and achieve satisfying performance.

- To evaluate our model against competing algorithms, we build a manually labeled sentiment data set using Amazon Mechanical Turk. This data set will be released to the research community to promote further investigations on both textual and visual sentiment.

[1]http://tinyurl.com/lb4xkak

2. RELATED WORK

For sentiment analysis of online user generated textual content, dictionaries based approaches [29, 2, 8, 13] have been widely used due to its efficiency and simplicity. Very recently, distributed representation of words started to attract research attention due to its ability in learning robust features for words [20]. Le and Mikolov [18] further proposed an approach to learn distributed representation for documents. They applied their document representations to sentiment analysis and achieve the best performance over existing competing algorithms.

There are also several recent works on visual sentiment analysis. The work in [25] is a machine learning algorithm to predict the sentiment of images using pixel-level color histogram and SIFT bag of words visual features. Motivated by the fact that sentiment involves high-level abstraction, which may be easier to explain by objects or attributes in images, both [4] and [34] proposed to employ visual entities or attributes as features for visual sentiment analysis. In [4], 1200 adjective noun pairs (ANP) are extracted to crawl images from Flickr. The responses of the trained 1200 ANP classifiers can be considered as mid-level features for visual sentiment analysis. The work in [34] employed a similar mechanism but using 102 scene attributes instead. More recently, You *et al.* [32] proposed a progressively trained Convolutional Neural Network for visual sentiment analysis. Compared with other approaches that employ low-level or mid-level features, CNN achieved the state-of-the-art performance in predicting image sentiments. A bench-marking analysis of CNN on emotion analysis is proposed in [33].

There are only a few publications on analyzing sentiment using multi-modalities, such as text and images. Both [30] and [6] 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 [4]. In addition, researchers have investigated cross-modal issues in other multimedia retrieval related tasks. Rasiwasia *et al.* [24] employed canonical correlation analysis (CCA) to learn the correlations between visual features and textual features for image retrieval. Besides that, Feng *et al.* [10] further developed so-called correspondence autoencoder for cross-modal retrieval, where a code layer is shared between the visual and textual autoencoder for *unsupervised* learning of parameters. Ngiam *et al.* [22] also proposed a multimodal *unsupervised* deep learning model, which achieved the best published results in visual speech classification. Meanwhile, Nitish and Ruslan [28] employed multimodal deep Boltzmann machine to learn joint representation of images and text by the sharing of high-level abstract representation. On the other hand, learning semantic mappings between visual and textual feature spaces has become popular due to the success of deep learning. Socher *et al.* [26] learned the semantic mappings in order to classify unseen visual classes. Frome *et al.* [11] employed hinge loss instead of using squared loss in their objective function to learn the semantic mapping between words and deep visual features for image annotation. Socher *et al.* [27] tried to even learn the transformation between sentence descriptions and images by margin based loss function on Recursive Neural Network and Convolutional Neural Network. More recently, Gong *et al.* [12] employed largely weakly annotated images to help learn the visual and textual embedding in their proposed stacked auxiliary embedding.

Inspired by these works on learning joint visual and textual models, we also rely on the recently successful deep learning techniques to extract features from images and text. However, different from the previously mentioned works, all of which are intended for unsupervised learning of shared feature embedding space between images and short text for image annotation or retrieval, our work intends to use supervised learning to enforce the consistency between the prediction labels of different features for sentiment analysis.

3. CROSS-MODALITY CONSISTENT REGRESSION (CCR)

In this section, we describe the details of our proposed model. Our main motivation is that different modalities should be consistent in terms of depicting the same subject. In sentiment analysis, given multiple modalities, we expect the utilization of features extracted from different modalities, such as images and text, to achieve more accurate sentiment analysis results.

3.1 Cross-modality consistent regression

Our model accepts input from different modalities of the same subject. The penalties between the predicted label distributions of different modality features need to be taken into consideration. To measure the penalty between any two different predicted label distributions, we employ KL divergence. In particular, let p and q denote two probability distributions of the same length. We define $D(p \parallel q)$ as the sum of KL divergence between them.

$$D(p \parallel q) = D_{KL}(p \parallel q) + D_{KL}(q \parallel p) \tag{1}$$

Assume we have a total of M different modalities and a total of N training instances. If we consider a pair-wise penalty of the given M modalities, we have to solve a total of $\binom{M}{2}$ penalty terms between any pair of modalities, which may be too complicated for a relatively large M. Instead, we first concatenate all the features from the M modalities and add penalty terms between the M individual modality features and the concatenated features. In this way, only M penalty terms need to be added. Motivated by these observations, the objective function is formulated in Eqn. (2). We denote by x_i^m (for $m \in \{1, \cdots, M\}$) the m-th modality features of the i-th instance and by x_i^c the concatenated features from all the M modality features of the i-th instance. $\Theta = \{\theta^c, \theta^1, \cdots, \theta^M\}$ are the parameters that needs to be learned. λ and γs are the hyper parameters to control the weights of different components in the proposed model.

$$\begin{aligned} \min_{\Theta, \lambda, \gamma_1, \cdots, \gamma_M} J(\Theta) &= \frac{1}{N} \sum_{i=1}^{N} D(y_i \parallel p_{\theta^c}(x_i^c)) + \frac{\lambda}{2} \theta^{cT} \theta^c \\ &+ \frac{\lambda}{2} \sum_{m=1}^{M} \theta^{mT} \theta^m + \sum_{m=1}^{M} \frac{\gamma_m}{N} \sum_{i=1}^{N} D(p_{\theta^c}(x_i^c) \parallel p_{\theta^m}(x_i^m)) \end{aligned} \tag{2}$$

Let $p_\theta(x_i)$ be the prediction function for the label distribution of x_i given the parameter vector θ. We use softmax function to evaluate the probability distribution, which is defined as

$$p_\theta(x_i) = \frac{1}{\sum_{k=1}^{K} e^{\theta_k^T x_i}} [e^{(\theta_1^T x_i)}, e^{(\theta_2^T x_i)}, \cdots, e^{(\theta_K^T x_i)}]^T, \tag{3}$$

where K is the number of classes and θ_k is the parameter vector for the k-th class, *i.e.* θ_k is a sub-vector of θ and $\theta_k = \theta_{[(k-1)|x|, k|x|)}$ (we use $|x|$ to denote the length of the feature vector).

The first component of the objective function is the consistency constraint between the ground-truth label and the predicted label distribution using concatenated features. Next, regularization terms are added to the framework to prevent overfitting. The last component considers the predicted distribution consistency between each single modality features and the concatenated features. In this way, we hope that it is possible to propagate knowledge learned from different modalities to each other to improve the overall performance of the system.

3.2 Relation to softmax regression

The proposed model is closely related to softmax regression, where the objective is to minimize the loss function in Eqn. (4). Similarly, $\theta = [\theta_1; \theta_2; \ldots; \theta_K]$ is the parameter vector, $\delta(\cdot)$ is the indicator function and x_i is the feature vector for the i-th instance.

$$\min_\theta -\frac{1}{N} \left(\sum_{i=1}^{N} \sum_{k=1}^{K} \delta(y_i = k) \ln \frac{\exp(\theta_k^T x_i)}{\sum_{j=1}^{K} \exp(\theta_j^T x_i)} \right) \tag{4}$$

Indeed, softmax regression is a special case of the first component $\frac{1}{N} \sum_{i=1}^{N} D(y_i \parallel p_{\theta^c}(x_i^c))$ of our proposed model. Eqn. (5) shows more details of this loss term, where C_1 and C_2 are constants determined by y_i. We use $p_\theta(k|x_i)$ to represent the probability of assigning x_i to class k, *i.e.* the k-th element in Eqn. (3). If we are given a *hard* label for each instance, *i.e.* y_{ik} only has one and only one non-zero entry, then the term $-y_{ik} \ln p_{\theta^c}(k|x_i)$ in Eqn. (5) is the same with the loss objective of softmax regression.

$$\begin{aligned} D(y_i \parallel p_{\theta^c}(x_i^c)) &= \sum_{k=1}^{K} p_{\theta^c}(k|x_i^c) \ln p_{\theta^c}(k|x_i^c) \\ &- y_{ik} \ln p_{\theta^c}(k|x_i^c) + C_1 p_{\theta^c}(k|x_i^c) + C_2 \end{aligned} \tag{5}$$

In addition, there are two other terms related to $p_{\theta^c}(x_i)_k$ in Eqn. (5), which are related to cross entropy. This is the main reason to use KL divergence instead of softmax regression for the first component of our model, which is favorable for tasks with noisy and uncertain labels, such as sentiment analysis.

3.3 Parameter learning

Even though KL divergence is convex, the proposed objective function is not convex on its parameters Θ. In this section, we explain in detail how to learn the parameters in our model.

3.3.1 Gradient descent

We resort to gradient descent to optimize the objective function $J(\Theta)$ in Eqn. (2). The gradient of the $J(\Theta)$ with respect to θ^c is

$$\begin{aligned} \frac{\partial J(\Theta)}{\partial \theta^c} &= \frac{1}{N} \sum_{i=1}^{N} \frac{\partial D(y_i \parallel p_{\theta^c}(x_i^c))}{\partial \theta^c} + \lambda \theta^c \\ &+ \sum_{m=1}^{M} \frac{\gamma_m}{N} \sum_{i=1}^{N} \frac{\partial D(p_{\theta^c}(x_i^c) \parallel p_{\theta^m}(x_i^m))}{\partial \theta^c}. \end{aligned} \tag{6}$$

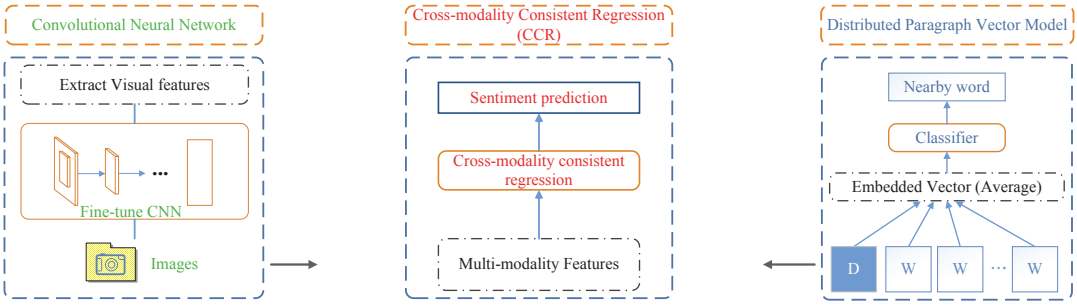


Figure 2: The framework for multi-modality sentiment analysis. *Left:* We fine-tune a CNN visual sentiment analysis model, which is employed to extract visual features for testing images. *Right:* We train a distributed paragraph vector model on the related titles and descriptions of the images to learn textual features. *Middle:* The proposed cross-modality consistent regression model is trained on the visual and textual features to learn the final sentiment classifier.

For last derivative term $\frac{\partial D(p_{\theta^c}(x_i^c)\|p_{\theta^m}(x_i^m))}{\partial \theta^c}$, we have[2]

$$\frac{\partial D_{KL}(p_{\theta^c}(x_i^c) \parallel p_{\theta^m}(x_i^m)) + D_{KL}(p_{\theta^m}(x_i^m) \parallel p_{\theta^c}(x_i^c))}{\partial \theta_{jl}^c} = \sum_{k=1}^{K} \left(1 - \frac{p_{\theta^m}(k|x_i^m)}{p_{\theta^c}(k|x_i^c)} - \ln \frac{p_{\theta^m}(k|x_i^m)}{p_{\theta^c}(k|x_i^c)}\right) \frac{\partial p_{\theta^c}(k|x_i^c)}{\partial \theta_{jl}^c}. \quad (7)$$

where

$$\frac{\partial p_{\theta^c}(k|x_i^c)}{\partial \theta_{jl}^c} = \frac{\partial \frac{\exp(\theta_k^{c\,T} x_i^c)}{\sum_{k=1}^{K} \exp(\theta_k^{c\,T} x_i^c)}}{\partial \theta_{jl}^c} = (\delta(k=j) - p_{\theta^c}(k|x_i^c))\, p_{\theta^c}(j|x_i^c) x_{il}^c \quad (8)$$

We can also calculate the first derivative term $\frac{\partial D(y_i\|p_{\theta^c}(x_i^c))}{\partial \theta^c}$, which is similar to Eqn. (7).

The gradient of the objective function $J(\Theta)$ with respect to θ^m for $m \in \{1, 2, \ldots, M\}$ is

$$\frac{\partial J(\Theta)}{\partial \theta^m} = \lambda \theta^m + \sum_{m=1}^{M} \frac{\gamma_m}{N} \sum_{i=1}^{N} \frac{\partial D(p_{\theta^c}(x_i^c) \parallel p_{\theta^m}(x_i^m))}{\partial \theta^m}. \quad (9)$$

Since $D(p_{\theta^c}(x_i^c) \parallel p_{\theta^m}(x_i^m))$ is symmetric in terms of θ^m and θ^c, we can apply Eqn. (7) to calculate the derivatives of Eqn. (9).

3.3.2 Learning algorithm and convergence analysis

There are two groups of parameters in our model, namely θ^c and $\{\theta^1, \theta^2, \cdots, \theta^M\}$. In our implementation, we learn those two groups of parameters iteratively. Specifically, in each iteration, the learning algorithm will try to update the two groups of parameters sequentially. Since, we built a large data set for our experiments, we employ mini-batch L-BFGS to learn the parameters[3].

Algorithm 1 summarizes the main steps for **C**ross-modality **C**onsistent **R**egression (CCR). The proposed objective function is differentiable, thus the function is smooth in terms of Θ. Meanwhile, it is easy to prove that $J(\Theta) \geq 0$. The objective function is lower-bounded. During each iteration of the learning algorithm, we are trying to find a smaller $J(\Theta)$ using L-BFGS. It is possible that mini-batch training may lead to oscillations of the objective function between different batches. However, overall we are still able to conclude that the iterative learning algorithm in Algorithm 1 converges when there is enough number of iterations.

We will discuss in the experimental section on how to select the hyper-parameters of our model, *i.e.* λ and γs.

[2]Recall that θ_j^c is a sub-vector of θ^c, we use θ_{jl}^c to represent the l-th element of θ_j^c.

[3]When the whole data set can fit into the machine's memory, it is also possible to employ full-batch L-BFGS.

Algorithm 1 Cross-modality consistent regression (CCR)

Require: $X^1, X^2, \cdots, X^M$ a total of M different modality features on a collection of objects X.
$Y = \{y_1, y_2, \ldots, y_N\}$ sentiment labels of X

```
Randomly split the objects into mini-batches
Concatenate the M modality features to get X^c
Randomly initialize Θ = {θ^c, θ^1, ..., θ^M}
repeat
    Randomly select one mini-batch X_b
    Apply L-BFGS to update θ^c on X_b with derivative in Eqn. (6) and objective function in Eqn. (2)
    for m from 1 to M do
        Randomly select one mini-batch X_b
        Apply L-BFGS to update θ^m on X_b with derivative in Eqn. (9) and objective function in Eqn. (2)
    end for
until Convergent or reach the maximum numbers of iterations
return Θ
```

3.3.3 Prediction algorithm

We can employ the learned model for prediction of testing instances given their M modality features. Recall that our training objective is to enforce the consistency of prediction results using different sets of modality features. Similarly, in the prediction stage, we also intend to obtain the same objective. Let p denote the desired label distribution of the testing instances, then we have

$$\min_p J(p|p_{\theta^1}, p_{\theta^2}, \ldots, p_{\theta^M}) = \sum_{k=1}^{M} \sum_i p(i) \ln \frac{p(i)}{p_{\theta^k}(i)} \quad s.t. \sum_i p(i) = 1. \quad (10)$$

THEOREM 1. *The optimal solution to* $J(p|p_{\theta^1}, p_{\theta^2}, \ldots, p_{\theta^M})$ *is that* $p(i) = \frac{\sqrt[M]{\Pi_k p_{\theta^k}(i)}}{\sum_j \sqrt[M]{\Pi_k p_{\theta^k}(j)}}$.

PROOF. The Lagrange function for the above problem is

$$\Lambda(p, \lambda) = \sum_{k=1}^{M} \sum_i p(i) \ln \frac{p(i)}{p_{\theta^k}(i)} + \lambda(\sum_i p(i) - 1).$$

Derivative of Λ with respect to $p(i)$ is

$$\frac{\partial \Lambda}{\partial p(i)} = M \ln p(i) - \sum_{k=1}^{M} \ln p_{\theta^k}(i) + M + \lambda.$$

Let the derivative equal 0, we have

$$M \ln p(i) = \sum_{k=1}^{M} \ln p_{\theta^k}(i) - M - \lambda.$$

From the constraint $\sum_i p(i) = 1$, we conclude that

$$p(i) = \frac{\sqrt[M]{\Pi_k p_{\theta^k}(i)}}{\sum_j \sqrt[M]{\Pi_k p_{\theta^k}(j)}}$$

□

4. MULTI-MODALITY SENTIMENT ANALYSIS

In this section, we describe in details on how to apply the proposed model to multi-modality sentiment analysis. In particular, we focus on how to extract the state-of-the-art visual and textual features and apply them to the proposed model. Figure 2 shows the framework for multi-modality sentiment analysis. The recent developed Convolutional Neural Network (CNN) [17] has achieved the state-of-the-art performance on a wide range of vision tasks. You *et al.* [32] conducted experiments on deploying CNN for visual sentiment analysis and achieved better performance than both low-level [25] features and mid-level [4, 34] features. Inspired by their conclusion, we propose to use CNN for the extraction of visual features. In particular, we employ the pre-trained CNN model on imagenet [15] to fine-tune a CNN model for visual sentiment analysis. The details on fine-tuning the CNN model will be discussed in the experimental section. Next, the fine-tuned CNN model is employed to extract visual features from the second to the last layer of the neural network.

For textual features, Le and Mikolov [18] developed an unsupervised language model to learn distributed representations for documents. They applied the learned representations to analyze textual sentiment, which achieved the best performance compared with other existing state-of-the-art textual sentiment analysis models. We employed the proposed model to learn distributed representations for related text of each image. In particular, We use descriptions and titles of each image as the body of a document to learn the textual features of each image.

Given the visual and textual features, we are able to train a cross-modality consistent regression model for sentiment analysis. Meanwhile, the trained visual and textual model can extract visual and textual features for testing images and text individually, which next can be used to predict the sentiment distribution for the image and related text respectively.

5. EXPERIMENTS

In this section, we conduct experiments to evaluate the performance of the proposed cross-modality consistent regression model on sentiment analysis. To train the visual and textual model in Figure 2, we choose to crawl data from Getty Images[4]. The main reasons to use Getty Images are its relatively formal descriptions of images and its convenient and powerful query based searching system.

5.1 Training visual and textual models

To fine-tune the pre-trained CNN model for sentiment analysis, we need a relatively large labeled data set, which can cost huge human efforts. Meanwhile, different people may have somewhat different opinions on the sentiment of the same object, which makes it harder to have a well labeled training data set. In our implementation, we propose to use *weakly* labeled data to train our neural network. To be more specific, we use a list of keywords for both positive and negative sentiment.[5] We query Getty Images with these keywords and all the returned images are labeled using the sentiment labels of these keywords. In this way, we are able to collect a large weakly labeled data set consisting of both images and text, which is employed to fine-tune the CNN model and learn the paragraph vector for related text of each image. Table 1 summarizes the statistics of our collected data set from Getty Images. In total, there are 101 keywords. We collect a total of over half million weakly labeled images as well as their titles and descriptions.

Table 1: Summary of the dataset from Getty Image.

Sentiment	Num of Keywords	Num of Images
Positive	37	311,940
Negative	64	276,281
Sum	101	588,221

Given the above collected data set, we randomly split them into 80% for training and 20% for testing. We fine-tune the CNN model on the publicly available implementation Caffe [15]. We run the GPU accelerated version of Caffe implementation with a total iteration of 200,000 on a Linux X86_64 machine with 32G RAM and two NVIDIA GTX Titan GPUs. The fine-tuned model is then employed to extract features for both training and testing images.

For textual model, the title and description of each image are concatenated as a single *document*. We use the algorithm in [31] to pre-process the textual data. First, numbers and special characters are removed. Then, we tokenize the text using the tokenizer model from NLTK (http://www.nltk.org). We also remove those words that appear less than 5 times in all the documents. The size of the paragraph vectors is 400 and the size of the nearby word window is 5, which are the default settings in [18].

We compare the performance of the proposed model with several baseline algorithms, including the following several different approaches. We also tried to use canonical correlation analysis (CCA) on this task. However, due to the scalability issue of CCA, we cannot fit all the training data into memory to learn the correlation using CCA. Table 2

[4]http://www.gettyimages.com

[5]http://www.sci.sdsu.edu/CAL/wordlist/origwordlist.html

Table 2: Performance of CCA on different testing data (see following sections for detailed description of the data).

Testing Data	Precision	Recall	F1	Accuracy
Getty	0.697	0.718	0.708	0.687
Twitter	0.769	0.698	0.731	0.727
AMT Twitter	0.66	0.52	0.559	0.526

summarizes the results of CCA on different sets of testing data using the same visual and textual features with other approaches using a single randomly selected mini-batch (10,240 instances). The results suggest poor performance compared with other approaches (see following details for details of the testing data and results of other approaches). Thus, we do not further compare the results of CCA with other approaches in the following experimental sections.

For all the following results of CCR, we run the algorithm 10 times with randomized initialization of the parameters. The averaged results are reported. The results of the following baselines are also reported and analyzed.

- **Single visual model.** We only use the visual features to build a logistic regression model, which outperforms models on both low-level and mid-level visual features [32].
- **Single textual model.** The paragraph feature vectors are also fed to a logistic classifier to predict the sentiment [18].
- **Early fusion.** We concatenate both visual and textual features and build a logistic regression model on these concatenated features.
- **Late fusion.** The average of the prediction sentiment score of visual and textual models is used as the prediction score of the late fusion model [30, 6].

5.2 Performance on Getty Images testing data set

We extract visual features for the 20% testing images given the fine-tuned CNN model. In this paper, we use the second to the last layer to extract features, which has a total of 4096 features for each image. For textual features, since the training is unsupervised, all documents are given to the model to learn their features [18].

Following the steps in Algorithm 1, we split the training data into mini-batches and train all visual, textual logistic regression model and CCR model on the same collection of mini-batches. In our implementation, we use a batch size of 10240, which is a trade-off between memory load and convergence rate.

Figure 3 shows the changes of the objective loss functions with the increase of mini-batch iteration numbers. The results show that the loss function value changes on some randomly chosen validation data set and training data set are comparable. Meanwhile, since we employ L-BFGS, the loss function converges after about 10 iterations, *i.e.* running on 10 mini-batches.

Since we have about $4,000$ visual features and $4,00$ textual features for each image, we try to balance the two modalities in selecting the hyper-parameters. In all of our experiments,

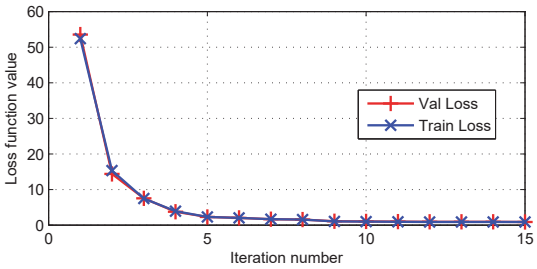


Figure 3: Changes of the objective loss function on both training and validating data set.

Table 3: Performance on the testing data set by different approaches.

Algorithm	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**

we set λ to be 1, γ_v for visual features also to be 1 and γ_t for textual features to be 5[6]. Table 3 shows the performance of different approaches on the 20% weakly labeled testing data from Getty Images. The results show that visual features may have comparable precision and recall on these data. Textual features can achieve higher precision but lower recall. Both early fusion and late fusion can produce improved results over single modalities. However, both fail to improve the performance of precision than the single textual model. On the other hand, the proposed CCR model can improve the performance of both precision and recall than the two single models. Meanwhile, CCR performs best among all the methods in terms of both F1 and accuracy score.

5.3 Performance on Twitter data set

We also build a new data set from image tweets. In particular, we employ the Twitter streaming API to collect a large number of Tweets. In total, we collected about 15 million Tweets. Next, we keep Tweets that contain both images and English text. In total, we collect $220,000$ image tweets. In our implementation, we employ the recent proposed VADER [14] to weakly label these tweets, which is a rule based textual sentiment analysis and attuned to Twitter contexts. Next, we select the top ranked positive and negative image tweets according to the VADER score. We manually filter out duplicates, low-quality, porn and all-text images. In this way, we collect a total of $31,584$ *weakly* labeled image tweets, $16,844$ of them are positive tweets and the rest are negative.

Since images from Twitter are much more diverse and different from Getty Images and tweets are also much more informal, we could not directly apply the trained model from Getty Images to these image tweets. Instead, we randomly

[6]Indeed, there is no significant different when we set $\gamma_t \in [5, 10]$, which is close to the proportion of feature size $|x_v|$ and $|x_t|$.

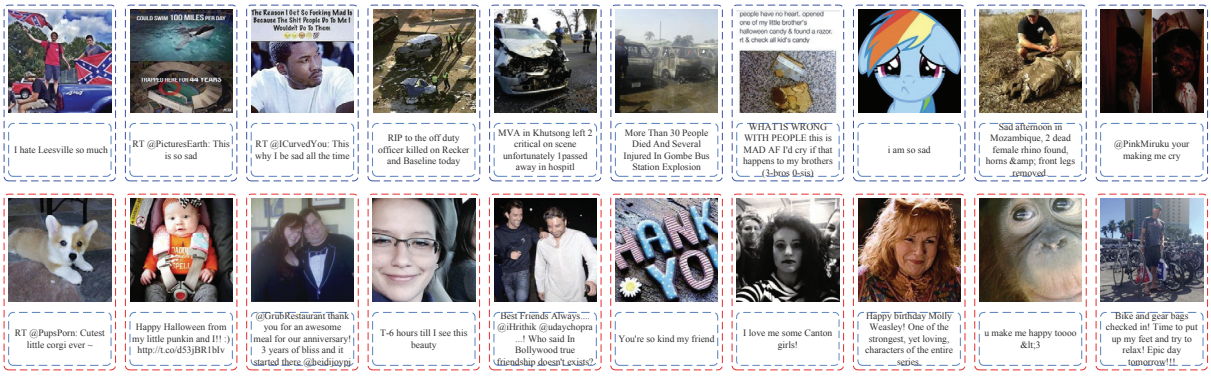


Figure 4: Top confidently ranked examples by AMT workers. Top row (in blue rectangle) shows positive examples and bottom row (in red rectangle) shows negative examples.

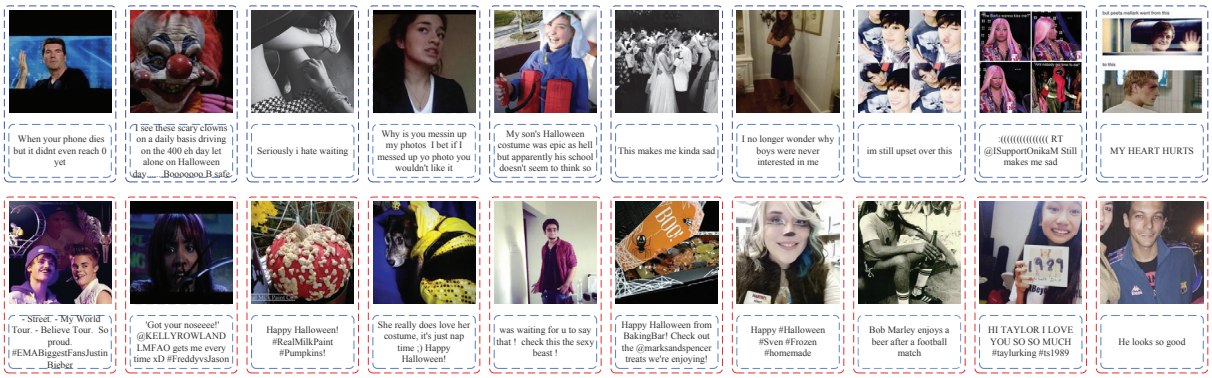


Figure 5: Ambiguously ranked examples by AMT workers. Top row (in blue rectangle) shows positive examples and bottom row (in red rectangle) shows negative examples. See text for explanation.

split this data set into batches with the same size of 10240. We use the first batch the testing data set and the rest the training data set to fine-tune both CNN and paragraph vector model. In particular, we slightly fine-tune the CNN model previously trained on the Getty Images with $2,000$ iterations with the learning rate set to 0.001. For paragraph vector model, we feed both the descriptions from Getty Images and the tweet text of the $31,584$ image tweets to this model to learn the vector representation for each tweet. For the tweets, we preprocessed them by further replacing hashtags, url links and user ids with special string sequences. Table 4 shows the results on the randomly selected 10240 testing image tweets. It is interesting to find that the textual features works better than the visual features. This may be due to the fact that we obtain the weak labels from text based system VADER and an insufficient number of images for CNN to find a relatively good local optima. However, the proposed CCR are able to improve the performance on the same set of visual and textual features.

5.3.1 *Performance on manually labeled tweets*

Meanwhile, in order to have more accurate labels for these image tweets, we employed crowd intelligence, Amazon Mechanical Turk (AMT), to generate sentiment labels for selected image tweets, in a similar fashion to [5]. We recruited 5 AMT workers for each of the candidate image tweet. We

Table 4: Performance on the Twitter testing data set by different approaches.

Algorithm	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**

test the performance of different models on this manually labeled data using the previously fine-tuned models on the *weakly* labeled image tweets. We randomly select $2,000$ image tweets and post them in AMT for sentiment annotation. After receiving the batch results from AMT, we keep those that have at least 4 agreements on the sentiment label and also exclude those that appears in the previously weakly labeled image tweets for fine-tuning. Eventually, we have 613 image tweets, of which 389 are labelled positive and 224 are labelled negative by 5 AMT workers.

Table 5 gives the performance of different approaches. CCR performs best in terms of precision, F1 and accuracy. However, it has a slightly lower recall. Compared with the results in Table 4, visual features show significant improve-

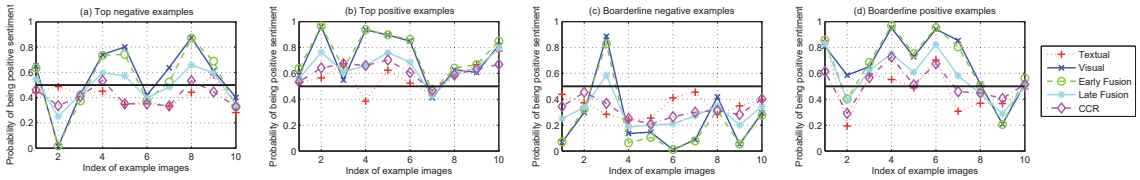


Figure 6: Machine performance on confident and uncertain examples labelled by AMT workers.

Table 5: Performance on the AMT manually labeled data set by different approaches.

Algorithm	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	**0.886**	0.730	**0.800**	**0.769**

ment, which may be due to the fact that AMT workers take both text and image to label the sentiment. Meanwhile, it is possible that the labels by AMT workers are biased compared with the weak labels given by VADER, causing relatively poor performance of both CCR and Early Fusion compared with the results in Table 4.

5.3.2 *Analysis of top ranked examples*

We also compare and analyze the top ranked examples of both AMT workers and machines. Since each image is labeled by 5 AMT workers into one of *strongly negative (-2), negative (-1), positive (1)* and *strongly positive (2)*, we rank images according to the sum of their scores by these 5 workers. Next, we select the top ranked positive and negative examples as well as some borderline examples. Figure 4 shows the top 10 ranked negative and positive examples by AMT workers. For negative examples, most of them are related to some bad experienced topics, such as car accident, environmental change and so on. Most of the positive examples are kind of cute, happy images along with some funny short descriptions. For comparison, ambiguously ranked examples are also selected and shown in Figure 5, where most of these examples also seem reasonable. It seems that the disagreement of these borderline examples may come from the inconsistency between the text and the visual content of an image tweet. Meanwhile, some of these image tweets may have celebrity related topic, which may also cause different opinions among different groups of fans.

Next, we conduct experiments on the performance of different machine approaches on these selected human labeled examples. Figure 6 shows the predicted results of different approaches on the two selected groups of example images in Figure 4 and Figure 5. It is interesting to note that for negative examples, machines seem to be uncertain on the confident examples given by AMT workers. However, they are confident on those uncertain examples. For positive examples, it seems that machine is kind of having similar recognition ability on both the confident and uncertain examples. These results demonstrate that the trained machine model and human beings may have different recognition ability

Table 6: Top 100 most confident sentiment prediction distribution of different algorithms.

Senti	Alg	5 Agree	4 Agree	4 Obj	5 Obj
Neg	Textual	22	52	7	19
	Visual	23	45	13	19
	Early	23	50	9	18
	Late	28	52	7	13
	CCR	**29**	**58**	**4**	**9**
Pos	Textual	61	**24**	12	3
	Visual	62	20	13	5
	Early	66	20	11	3
	Late	69	**24**	**6**	1
	CCR	**71**	22	7	**0**

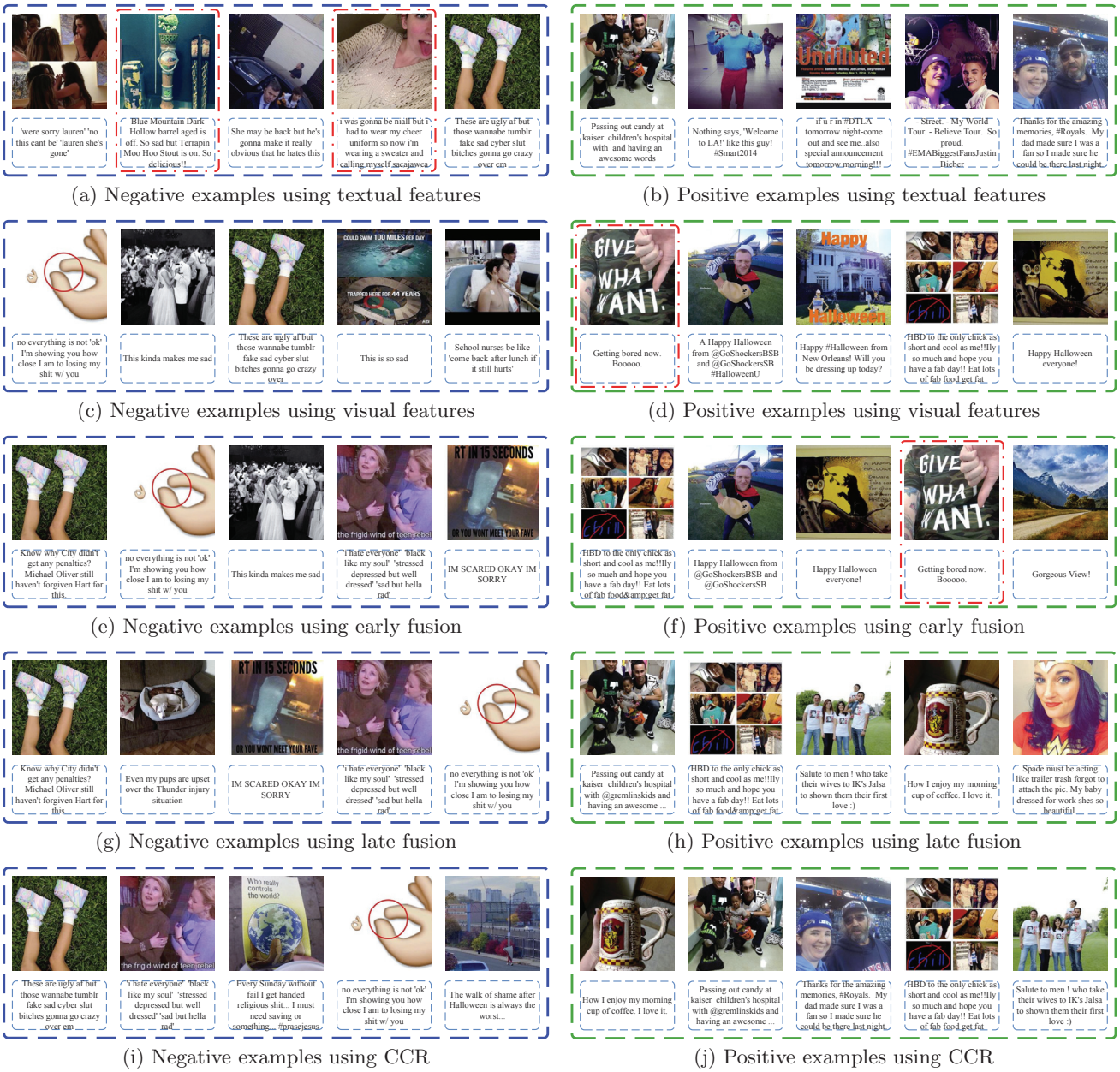
towards the sentiment of the same group of images, which may be due to difference between the limited training samples for machines and the constantly learning process for human beings.

We also extract the top 100 positive and top 100 negative image tweets by AMT workers. The prediction results of different approaches are given in Table 6. Overall, CCR outperforms other approaches in both negative and positive categories in terms of accuracy. Meanwhile, all approaches seem to be more likely to agree with AMT workers on the positive category. This may be due to the biased nature of social networks, where users are more likely to post positive content than negative content.

In addition, we extract the most *confident* prediction examples of different approaches on the manually labeled image tweets by AMT workers. We rank the images by the prediction score of each model. Figure 7 shows the top ranked 5 images of each model on both positive and negative categories respectively (red circles indicate wrongly predicted samples). All the image tweets are ranked from left to right in a decreasing order. There are many common highly ranked examples between different approaches. However, different approaches have different ranking orders. In particular, highly ranked examples using textual features seems to have strong discriminative words than those using visual features, which explains the main reason of the two wrongly predicted examples. Similarly, only using visual features may also lead to wrong confident examples due to the lack of knowledge from the text data. Meanwhile, we note that there are no shared top ranked examples with that given by human beings in Figure 4. Again, these differences may come from different learning scenarios for both human beings and machines. Meanwhile, this also suggests the challenging nature of visual sentiment analysis.

6. CONCLUSIONS

Sentiment analysis, in particular visual sentiment analysis, is a challenging and interesting problem. In this work,



(a) Negative examples using textual features

(b) Positive examples using textual features

(c) Negative examples using visual features

(d) Positive examples using visual features

(e) Negative examples using early fusion

(f) Positive examples using early fusion

(g) Negative examples using late fusion

(h) Positive examples using late fusion

(i) Negative examples using CCR

(j) Positive examples using CCR

Figure 7: Examples of most confident image tweets of different approaches. *Left* column shows the most confident negative examples. *Right* column shows the most confident positive examples.

we aim to analyze sentiment via both visual and textual content. The recently developed machine learning algorithms lead to the availability of robust visual and textual features for abstract tasks, such as sentiment analysis. Due to the largely easily accessible weakly labeled data, we can train both visual and textual models to extract robust features for sentiment analysis. We develop a cross modality consistency regression model, which tries to enforce the agreement between sentiment labels predicted by different modality features. The experimental results suggest that the proposed multi-modality regression model outperforms both the state-of-the-art single textual and visual sentiment analysis models and two fusion models.

Meanwhile, the main advantage of using convolutional neural networks and unsupervised paragraph vector model is that we can transfer the knowledge to other domains using a much simpler fine-tuning technique than those in the literature *i.e.*, [9]. We also hope our sentiment analysis results can encourage further research on online user generated multimedia content.

Acknowledgements

This work was generously supported in part by Adobe Research, and New York State CoE IDS. Jianchao Yang performed related work while he was with Adobe Research.

7. REFERENCES

[1] S. Asur and B. A. Huberman. Predicting the future with social media. In *WI-IAT*, volume 1, pages 492–499. IEEE, 2010.

[2] J. Bollen, H. Mao, and A. Pepe. Modeling public mood and emotion: Twitter sentiment and socio-economic phenomena. In *ICWSM*, 2011.

[3] J. Bollen, H. Mao, and X. Zeng. Twitter mood predicts the stock market. *Journal of Computational Science*, 2(1):1–8, 2011.

[4] D. Borth, T. Chen, R. Ji, and S.-F. Chang. Sentibank: large-scale ontology and classifiers for detecting sentiment and emotions in visual content. In *ACM MM*, pages 459–460. ACM, 2013.

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

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

[7] D. C. Cireşan, U. Meier, J. Masci, L. M. Gambardella, and J. Schmidhuber. Flexible, high performance convolutional neural networks for image classification. In *IJCAI*, pages 1237–1242, 2011.

[8] D. Davidov, O. Tsur, and A. Rappoport. Enhanced sentiment learning using twitter hashtags and smileys. In *ICL*, pages 241–249, 2010.

[9] L. Duan, D. Xu, I.-H. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *IEEE PAMI*, 34(9):1667–1680, 2012.

[10] F. Feng, X. Wang, and R. Li. Cross-modal retrieval with correspondence autoencoder. In *ACM MM*, pages 7–16. ACM, 2014.

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

[12] Y. Gong, L. Wang, M. Hodosh, J. Hockenmaier, and S. Lazebnik. Improving image-sentence embeddings using large weakly annotated photo collections. In *ECCV*, pages 529–545. Springer, 2014.

[13] X. Hu, J. Tang, H. Gao, and H. Liu. Unsupervised sentiment analysis with emotional signals. In *WWW*, pages 607–618, 2013.

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

[15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv preprint arXiv:1408.5093*, 2014.

[16] D. Joshi, R. Datta, E. Fedorovskaya, Q.-T. Luong, J. Z. Wang, J. Li, and J. Luo. Aesthetics and emotions in images. *IEEE Signal Processing Magazine*, 28(5):94–115, 2011.

[17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[18] Q. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, 2014.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[20] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.

[21] L.-P. Morency, R. Mihalcea, and P. Doshi. Towards multimodal sentiment analysis: Harvesting opinions from the web. In *ICMI*, pages 169–176, 2011.

[22] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, pages 689–696, 2011.

[23] B. O'Connor, R. Balasubramanyan, B. R. Routledge, and N. A. Smith. From tweets to polls: Linking text sentiment to public opinion time series. *ICWSM*, 11:122–129, 2010.

[24] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos. A new approach to cross-modal multimedia retrieval. In *ACM MM*, pages 251–260. ACM, 2010.

[25] S. Siersdorfer, E. Minack, F. Deng, and J. Hare. Analyzing and predicting sentiment of images on the social web. In *ACM MM*, pages 715–718. ACM, 2010.

[26] R. Socher, M. Ganjoo, C. D. Manning, and A. Ng. Zero-shot learning through cross-modal transfer. In *NIPS*, pages 935–943, 2013.

[27] R. Socher, Q. Le, C. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. In *NIPS Workshop*, 2013.

[28] N. Srivastava and R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, pages 2222–2230, 2012.

[29] A. Tumasjan, T. O. Sprenger, P. G. Sandner, and I. M. Welpe. Predicting elections with twitter: What 140 characters reveal about political sentiment. *ICWSM*, 10:178–185, 2010.

[30] 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.

[31] S. Wang and C. D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, pages 90–94, 2012.

[32] Q. You, J. Luo, H. Jin, and J. Yang. Robust image sentiment analysis using progressively trained and domain transferred deep networks. In *The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[33] Q. You, J. Luo, H. Jin, and J. Yang. Building a large scale dataset for image emotion recognition: The fine print and the benchmark. In *The Thirtieth AAAI Conference on Artificial Intelligence (AAAI)*, 2016.

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

[35] X. Zhang, H. Fuehres, and P. A. Gloor. Predicting stock market indicators through twitter "i hope it is not as bad as i fear". *Procedia-Social and Behavioral Sciences*, 26:55–62, 2011.