Cascading Multiway Attention for Document-level Sentiment Classification

Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, Xu Sun

MOE Key Lab of Computational Linguistics, Peking University, Beijing 100871, China {madehong, lisujian, zxdcs, wanghf, xusun}@pku.edu.cn

Abstract

Document-level sentiment classification aims to assign the user reviews a sentiment polarity. Previous methods either just utilized the document content without consideration of user and product information, or did not comprehensively consider what roles the three kinds of information play in text modeling. In this paper, to reasonably use all the information, we present the idea that user, product and their combination can all influence the generation of attention to words and sentences, when judging the sentiment of a document. With this idea, we propose a cascading multiway attention (CMA) model, where multiple ways of using user and product information are cascaded to influence the generation of attention on the word and sentence layers. Then, sentences and documents are well modeled by multiple representation vectors, which provide rich information for sentiment classification. Experiments on IMDB and Yelp datasets demonstrate the effectiveness of our model.

1 Introduction

Document-level sentiment classification aims to predict an overall sentiment polarity (e.g., 1-5 stars or 1-10 stars) for a user review document. This task recently draws increasing research concerns and is helpful to many downstream applications, such as user and product recommendation.

Early work focuses on traditional machine learning associated with handcraft text features for sentiment classification (Pang et al., 2002; Ding et al., 2008; Taboada et al., 2011). With the development of deep learning techniques, some researchers design neural networks to automatically learn features from document content and achieve comparable performance (Glorot et al., 2011; Kalchbrenner et al., 2014; Yang et al., 2016), though they ignore the use of user and product information. Almost at the same time, deep learning techniques exhibit another advantage that product and user information can be flexibly modeled with document content for sentiment classification (Tang et al., 2015a; Chen et al., 2016). Tang et al. (2015a) design user and product preference matrices to tune word representations, based on which convolutional neural networks (CNNs) are used to model the whole document. To avoid the high-cost preference matrix, Chen et al. (2016) develop the two-layer (i.e., word and sentence layers) model, where the combination of user and product information is used to generate attention to words and sentences respectively on each layer.

Though previous studies achieve improvements in synthesizing text, user and product for sentiment classification, they are somewhat limited to either of the following two aspects. First, when considering user and product information, previous work mostly models their combination which may cause adverse effects on sentiment classification. As we can see, user and product information also have their own influence on sentence and document representations. For example, a lenient user tends to focus on the good aspects, while a picky user is always concerned about unsatisfactory clues, no matter which product is reviewed. In another view, different products have their own concerns. For example, a car is closely related with fuel consumption while a hotel with bed or food, which are relatively independent of users. Second, in previous hierarchical modeling method, different layers are usually treated with similar means. In our opinion, different linguistic units (e.g., words and sentences) should be given different treatment to achieve good performance. As we can see, words are closely related with each other and sentences are relatively independent in reflecting their sentimental polarity. Take a restaurant review for example, the review "*Its dinner environment is good. But the pizza is not worth the money!*" is comprised of two sentences. For this review, a user who values environment may mark 5 stars, while a user who thinks highly of food may mark 1 star. Thus, we can not simply combine the two sentences together in judging its sentiment polarity. It is better to model each sentence independently before knowing the users' or products' attention.

To address the two issues above, we propose to comprehensively make use of the information of user, product, and their combination to generate attention on the word and sentence layers, and further cascade the multiple ways of attention on the generation of sentence and document representations. Here, we name our proposed model the cascading multiway attention (CMA) model. Specifically, each sentence is first represented with multiple representation vectors through using different ways of attention to focus on the important part. Next, the similar multiple ways of generating attention are applied on the sentence level and used to tune the generation of document representations. In such a case, document representation is not simply treated as a sequence of sentences, but considers more about the influence of user and product information. Finally, we can get a well modeled document representation for sentiment classification. Experiments on three real-world datasets demonstrate that CMA achieves state-ofthe-art performance and fully considers the contributions from user and product information.

2 Cascading Multiway Attention Model for Sentiment Classification

In this section, we first give an overview of the cascading multiway attention (CMA) model for sentiment classification. Then, we focus on introducing how to design multiway attention for modeling sentences and documents through using user and product information. Next, we show the training details of CMA.

2.1 Model Overview

Figure 1(a) shows the overall architecture of our CMA model, which is mainly composed of four

levels including word level, sentence level, document level, classification level. With word embeddings as input, we can employ convolutional neural networks or recurrent neural networks to obtain deeper semantic of words on the word level. On the sentence level, we design the multiway attention networks to generate attention to words from three different views and get three representation vectors to represent a sentence. On the document level, we keep generating multiple kinds of attention to sentences and get the document representation, which is fed into a softmax function for classification.

Specifically, let us first formalize the notation. We suppose that a document contains msentences $\{S_1, S_2, ..., S_m\}$ whose lengths are set $n_1, n_2, ..., n_m$ respectively. For Sentence $S_t(1 \le t \le m)$, it is composed of a sequence of words $w_t^1, w_t^2, ..., w_t^{n_t}$ where w_t^j denotes a specific word. To represent a word, we embed each word into a low dimensional real-value vector, called word embedding (Bengio et al., 2003).

Then, we can get $w_t^j \in \mathbb{R}^d$ from $M^{v \times d}$, where t is the sentence index in a document, j denotes the word index in sentence t, d means the embedding dimension and v gives the vocabulary size. Word embeddings can be regarded as parameters of neural networks or pre-trained from proper corpus via language model (Collobert and Weston, 2008; Mnih and Hinton, 2007; Mikolov et al., 2010; Huang et al., 2012). In our model, we choose the second strategy.

Next, deeper word semantics representations can be learned by using the neural network models, such as convolutional neural networks (CNN) or recurrent neural networks (RNN). In this paper, the LSTM model is employed to obtain the word representation, since it has the good performance of learning the long-term dependencies and can well model the dependence between words. Formally, for sentence S_t , we input its word embeddings $w_t^1, w_t^2, ..., w_t^{n_t}$ to the LSTM networks and get the final word representations $r_t^1, r_t^2, ..., r_t^{n_t}$.

With deeper word representations as input, we adopt the attention mechanism to model sentences. As Section 1 stated, we consider the influence from three views: user (δ^u) , product (δ^p) , and combination of user and product (δ^{up}) , which can generate three kinds of attention to words. Then, we impose the three attention vectors on word representations respectively and get three represen-



Figure 1: Overall Architecture of CMA.

tation vectors for each sentence. Formally, sentence S_t is represented by the concatenation of three vectors $\mathbf{s}_t^u, \mathbf{s}_t^p, \mathbf{s}_t^{up}$. On the document level, we still use δ^u , δ^p and δ^{up} respectively to generate attention to sentences and get three vectors $\mathbf{d}^u, \mathbf{d}^p$, \mathbf{d}^{up} for representing the document. The concrete design is introduced in the next subsection.

Finally, the three vectors \mathbf{d}^u , \mathbf{d}^p , \mathbf{d}^{up} are concatenated to a vector \mathbf{d} for a classifier. Here, we use a non-linear layer to project \mathbf{d} into the space of the targeted C classes. That is,

$$x = tanh(W_l \cdot \mathbf{d} + b_l) \tag{1}$$

where W_l and b_l are the weight matrix and bias respectively. The probability of labeling document with sentiment polarity $i(i \in [1, C])$ is computed by:

$$y_i = \frac{exp(x_i)}{\sum_{i=1}^{C} exp(x_i)}$$
(2)

The label with highest probability is set as the final result.

2.2 Cascading Multiway Attention

Now, we detail how to model user and product information and cascade their influence on representing sentences and documents.

As stated in Section 1, the meaning of a sentence has different interpretations from different views including user, product and their combination, under which words may be paid different attention on. With consideration of each view, only some specific parts are focused on. And different views have their own focuses which contribute to sentiment classification. Thus, we propose the multiway attention mechanism to capture sentence focuses with repsect to different views.

Specifically, in our model, we represent each user and each product with a representation vector which is learned from the model, notated with \mathbf{u} and \mathbf{p} here. To consider the influence from user and product, we formalize user view, product view and their combination view respectively as:

$$\delta^u = W_u \cdot \mathbf{u} \tag{3}$$

$$\delta^p = W_p \cdot \mathbf{p} \tag{4}$$

$$\delta^{up} = W_u' \cdot \mathbf{u} + W_p' \cdot \mathbf{p} \tag{5}$$

where W_u , W_p , W_u' , and W_p' are weight matrices for tuning the influence of **u** and **p**.

We take sentence S_t for example to describe the sentence modeling process, as shown in Figure 1(b). With word representations $r_t^1, r_t^2, ..., r_t^{n_t}$, we can generate three attention vectors using δ^u , δ^p and δ^{up} respectively. For convenience we use δ^* to denote δ^u , δ^p or δ^{up} . Then, the attention mechanism generates the attention vector α_t^* with respect to δ^* . The j^{th} dimension $[\alpha_t^*]^j$ denotes the attention weight of the j^{th} word r_t^j .

$$[\alpha_t^*]^j = \frac{exp(\gamma(r_t^j, \delta^*))}{\sum_{j=1}^{n_t} exp(\gamma(r_t^j, \delta^*))}$$
(6)

where γ is a score function that measures the importance of r_t^j in the sentence. The score function γ is defined as:

$$\gamma(r_t^j, \delta^*) = tanh(W_H \cdot r_t^j + \delta^* + b_s) \cdot v^T \quad (7)$$

where W_H and b_s are weight matrix and bias respectively, and v^T is the transpose of the weight vector v.

After computing the word attention weights, we can get sentence representations \mathbf{s}_t^* using different attention weights:

$$\mathbf{s}_{t}^{*} = \sum_{j=1}^{n_{t}} [\alpha_{t}^{*}]^{j} r_{t}^{j}.$$
(8)

Then, we get a multi-vector sentence representation: $[\mathbf{s}_t^u, \mathbf{s}_t^p, \mathbf{s}_t^{up}]$ for sentence S_t from three views.

On the sentence level, we apply the similar multiway attention strategy to get the document representation. Unlike the word level, each sentence is relatively independent of other sentences and takes its own part in determining the document sentiment. Thus, we ignore modeling the dependence between sentences and directly use the sentence representations derived from the word level. As we see, from different views each sentence plays a different role, and from a specific view all the sentences should be paid different attention, in document-level sentiment classification. Here, we still consider the influence from user, product and their combination, and get the corresponding document representation $\mathbf{d}^{u}, \mathbf{d}^{p}, \mathbf{d}^{up}$, where \mathbf{d}^{*} $(* \in \{u, p, up\})$ is computed as follows:

$$\mathbf{d}^* = \sum_{t=1}^m \beta_t^* \mathbf{s}_t^* \tag{9}$$

$$\beta_t^* = \frac{exp(\gamma(\mathbf{s}_t^*, \delta^*))}{\sum_{t=1}^m exp(\gamma(\mathbf{s}_t^*, \delta^*))}$$
(10)

where \mathbf{s}_t^* is the sentence representation of sentence S_t with respect to δ^* . β_t^* denotes the corresponding attention weight of S_t on the view of δ^* , and the γ function is the same as in Eq. (7).

2.3 Model Training

In CMA, we need to optimize all the parameters notated as Θ which are from all the user and product embedding vectors, LSTM networks, two multiway attention layers and the softmax layer.

We use cross entropy with L_2 regularization as the loss function, which is defined as:

$$J = -\sum_{i=1}^{C} g_i \log(y_i) + \lambda_r (\sum_{\theta \in \Theta} \theta^2)$$
(11)

where $g_i \in R^C$ denotes the ground truth, represented by one-hot vector. $y_i \in R^C$ is the estimated

probability for each class, computed as in Eq. (2). λ_r is the coefficient for L_2 regularization.

After learning Θ , we test the instances by feeding their text with user and product information into CMA, and the label with the highest probability stands for the predicted sentiment polarity.

3 Experiments

In this section, we first introduce the datasets and metrics used in our experiments. Then, we show the hyperparameter setting of our model. Next, we present the results of our model and compare our model with other state-of-the-art methods. Finally, we use a case study to examine the experimental results of CMA.

Datasets

To validate the effectiveness of our model, we use three real-world datasets: IMDB, Yelp 2013 and Yelp 2014 collected by Tang et al. (2015b). For these data, we preprocess the text including using Stanford CoreNLP (Manning et al., 2014) to split the review documents into sentences and tokenizing all words. Table 1 shows the details of the three datasets including number of documents (#docs), average number of documents per user posts(#docs/user) etc. It is also noted that IMDB is rated with 10 sentiment labels (i.e., 1-10 stars) while Yelp has 5 labels (i.e., 1-5 stars). We also adopt the same data partition used in (Tang et al., 2015b) and (Chen et al., 2016) for training, developing and test.

Evaluation Metrics

To evaluate the performance of sentiment classification, we adopt the metrics of Accuracy and RMSE. Accuracy measures the overall performance. Given the number of correctly predicted samples T and the total number of samples N, Accuracy is defined as:

$$Acc = \frac{T}{N}.$$
 (12)

RMSE evaluates the divergence between predicted labels and ground truth labels and is defined as:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{N} (l_k - l'_k)^2}{N}}$$
(13)

where l_k and l'_k denote the ground truth and predicted label of sample k respectively. Generally, a good system has a high accuracy and a low RMSE.

Dataset	#docs	#users	#products	#docs/user	#docs/product	#sents/doc	#words/doc	#labels
IMDB	84919	1310	1635	64.82	51.94	16.08	24.54	10
Yelp2013	78966	1631	1633	48.42	48.36	10.89	17.38	5
Yelp2014	231163	4818	4194	47.97	55.11	11.41	17.26	5

Table 1: Data Statistics of IMDB, Yelp2013 and Yelp 2014.

Hyperparameter Setting

Following (Chen et al., 2016) and (Tang et al., 2015b), we set the dimensions of word, user and product embeddings as 200 in our experiments. We train the word embeddings through using the training and developing sets of each dataset with word2vec tool (Mikolov et al., 2013). The user and product embeddings are initialized randomly. All the weight matrices are initialized by uniform distribution, and all biases are assigned as zero. We also set the dimensions of the LSTM hidden states and attention vectors as 200.

To train our model, we utilize AdaDelta (Zeiler, 2012), which adopts a self-adaptive learning rate, to optimize the parameters. The coefficient of penalty in the objective function is set to 10^{-5} .

3.1 Method Comparison

To comprehensively evaluate the performance of *CMA*, we list some baseline methods for comparison. The baselines are introduced as follows.

• **Majority** assigns the largest sentiment polarity occurred in the training set to each sample in the test set.

• **Trigram** uses the unigrams, bigrams and trigrams features to train a SVM classifier for sentiment classification.

• **TextFeature** extracts word/character ngrams, sentiment lexicon features, negation features, etc. for a SVM classifier.

• UPF extracts user-leniency features and product features from training data (Gao et al., 2013). There features can be concatenated with the features of *Trigram* and *TextFeature*.

• AvgWordvec averages the word embeddings in a document to generate the document representation as features for a SVM classifier.

• **SSWE** first learns the sentiment-specific word embeddings and then utilizes three kinds of pooling (i.e., max, min and average) to generate the document representation for a SVM classifier (Tang et al., 2014).

• **PV**(Paragraph Vector) is an unsupervised framework to learn distributed representations for text of any length (Le and Mikolov, 2014). (Tang

et al., 2015a) implements the distributed memory model of paragraph vectors (PV-DM) to get document representations for sentiment classification.

• **RNTN+RNN** models sentences using recursive neural tensor networks (RNTN) (Socher et al., 2013). Then sentence representations are fed into the recurrent neural networks (RNN) and their hidden states are averaged to get the document representation.

• UPNN designs preference matrices for each user and product to modify word representations (Tang et al., 2015b). Word representations are then fed into the convolution neural networks (CNNs) and concatenated with the user/product representation to generate document representation before a softmax layer. Without considering user and product information, the UPNN(noUP) method just uses CNN to model the documents.

• NSC+UPA proposes the hierarchical neural networks which are composed of two LSTM models to generate word and sentence representations (Chen et al., 2016). The combination of user and product information is used to generate attention to words and sentences on the sentence and document levels. The document representation is fed into a softmax layer. The *NSC* model is similar to *NSC+UPA*, but ignores user and product information and directly applies two layers of LSTM and attention mechanism to model sentences and documents using only document content.

When comparing with our model, we directly use the results of the above baselines reported in (Chen et al., 2016), as we conduct the experiments on the same datasets.

• **CA-null** is a simplified version of our CMA model without consideration of user and product information. We use one LSTM with attention to model sentences. Based on the sentence representations, we directly use the attention mechanism to model documents for sentiment classification.

The performance comparison of CMA and all baselines are shown in Table 2. All the baseline methods are divided into two categories: the first category only uses document content and the second comprehensive considers document, user and

Model	IMDB	Yelp 2013	Yelp 2014	
Model	Acc. RMSE	Acc. RMSE	Acc. RMSE	
Majority	0.196 2.495	0.411 1.060	0.392 1.907	
Trigram	0.399 1.783	0.569 0.841	0.577 0.804	
TextFeature	0.402 1.793	0.556 0.814	0.572 0.800	
AvgWordvec	0.304 1.985	0.526 0.898	0.530 0.893	
SSWE	0.312 1.973	0.549 0.849	0.557 0.851	
PV	0.341 1.814	0.554 0.832	0.564 0.802	
RNTN+RNN	0.400 1.764	0.574 0.804	0.582 0.821	
UPNN(noUP)	0.405 1.629	0.577 0.812	0.585 0.808	
NSC+LA	0.487 1.381	0.631 0.706	0.630 0.715	
CA-null	0.491 1.408	0.635 0.689	0.640 0.690	
Trigram+UPF	0.404 1.764	0.570 0.803	0.579 0.789	
TextFeature +UPF	0.402 1.774	0.561 0.822	0.579 0.791	
UPNN	0.435 1.602	0.596 0.748	0.608 0.764	
NSC+UPA	0.533 1.281	0.650 0.692	0.667 0.654	
CMA	0.540 1.191	0.664 0.677	0.676 0.637	

Table 2:Methods Comparison on IMDB, Yelp2013 and Yelp 2014 Datasets.

product information.

In the first category of methods, the *Majority* method is the worst, meaning the majority sentiment polarity occupies 19.2%, 42.1% and 39.2% of all samples. All the other methods are better than the Majority method, showing that manual or automatic features can both bring performance improvement for sentiment classification. Though techniques of deep neural networks are widely used and have made a success in document modeling, its simple application (e.g., Ave-Wordvec, PV) does not achieve better results than using manually selected features (e.g., Trigram, TextFeature) with respect to the accuracy metric. As for the three kinds of carefully designed neural networks, the CA-null and NSC+LA methods exhibit obvious advantages with high accuracy and low RMSR, compared to the RNTN+RNN and UPNN(noUP) methods. The main reason may be that CA-null and NSC+LA can learn the longdistance dependencies with LSTM networks and make full use of the important words and sentences with attention. We can also see that CAnull outperforms NSC+LA regarding the accuracy on the three datasets. This verifies that we may not consider sentence dependence when modeling documents for sentiment classification, as the main difference between CA-null and NSC+LA is that CA-null does not apply LSTM to model sentences.

From Table 2, we observe that user and product information can promote the performance of sentiment classification. Even the worst method (TextFeature+UPF) in the second cate-

Model	IMDB		Yelp 2013		Yelp 2014	
Model	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
CA-null	0.491	1.408	0.635	0.689	0.640	0.690
$CA-\delta^u$	0.513	1.397	0.641	0.688	0.653	0.687
$\operatorname{CA-}\delta^p$	0.508	1.423	0.640	0.694	0.652	0.685
$CA-\delta^{up}$	0.520	1.281	0.645	0.684	0.658	0.670
MA-max	0.528	1.314	0.654	0.680	0.663	0.667
MA-avg	0.521	1.341	0.651	0.678	0.662	0.672
СМА	0.540	1.191	0.664	0.677	0.676	0.637

Table 3: Analysis of Cascading and Multiway Attention.

gory is comparable to the state-of-the-art system (RNTN+RNN) in the first category. For the methods with manually designed features, their performance improvement is smaller than those methods with deep learning methods. Both Trigram-UPF and TextFeature-UPF concatenate more user and product features, but have little improvement compared with Trigram and TextFeature. The reason may be that the text features are huge (1M trigram) enough and the newly added user and product features can not work well. Adding user and product information, CMA outperforms CA-null by about 5 percent with respect to the accuracy metric on IMDB dataset. At the same time, CMA performs stably better than NSC+UPA with higher accuracy and lower RMSE on three datasets, with cascading multiway attention on word and sentence levels and keeping independency between the sentences. This validates that only a combination of user and product information can not boost performance much and more ways of attention should be explored in text modeling. It is also noted that words need to be modeled with consideration of their dependence while sentences should be independently modeled.

3.2 Model Analysis

In this section, we design a series of models to verify the effectiveness of our model *CMA*. On one hand, we design three simplified models, i.e., $CA-\delta^u$, $CA-\delta^p$ and $CA-\delta^{up}$, which only consider one-way attention from user, product, combination of user and product respectively when modeling sentences and documents. On the other hand, we just keep the multiway attention on modeling sentences and directly adopt the maximization or average operation on sentence representations to model a document, which are named *MA-max* and *MA-avg* respectively. Table 3 shows the results of all the models.

From Table 3, we can see that CA-null gets the

worst result compared with other models utilizing user and product information, which indicates that both information is useful for document sentiment. The performance of $CA-\delta^u$ and $CA-\delta^p$ are worse than $CA-\delta^{up}$, this is because $CA-\delta^u$ and $CA-\delta^p$ just uses one kind of information (user or product), but $CA-\delta^{up}$ uses both information. CMA achieves a much better result than the 'CA' models, which can verify the effectiveness of introducing multiway attention. We also observe that $CA-\delta^u$ is slightly better than $CA-\delta^p$, verifying user information plays a more important role than product information as in (Chen et al., 2016).

As for *MA-max* and *MA-avg*, multiway attention on modeling sentences bring much performance improvement compared to *CA-null*, though the cascading structure is not adopted. *MA-avg* seems to treat each sentence equally while *MAmax* tries to seek the most represented sentences contributing to sentiment classification. We can see that *MA-max* performs better than *MA-avg*, verifying that sentences should not be treated equally and multiway attention are a good mechanism for computing the contributions of each sentence to representing a document.

We can also see that *CMA* achieves the best performance among all the models, since *CMA* fully considers the word importance to a sentence and sentence importance to a document via sentence-level and document-level multiway attention. Next, we give an example to illustrate the cascading multiway attention captured by CMA.

Case Study

Here, we use a five-star review document from Yelp as a case study, and the document is "Biscuits made from scratch, not frozen. Homemade pancakes with real buttermilk, not from a mix or box. Homemade blueberry jam, no jam or can. True dinner coffee. A group of friendly, engaging staff. Five stars... need I say more.". We apply CMA on the document and achieve the correct sentiment polarity. Figure 2 visualizes the multiway attention weights on the sentence and document levels computed by CMA. We represent three ways of attention, i.e., user(δ^u), product(δ^p), and combination of user and product (δ^{up}), with blue, red and green color series respectively. The deeper color means higher weight.

Figure $2(a)\sim(f)$ display the three ways of attention to words for sentence1 ~ sentence6 respectively. We can see that different words are assigned high weight values when using different ways of attention. For example, for sentence1, the word "biscuits" is paid more attention from the user view while "frozen" is focused on from the product view and "made" is emphasized considering the combination of user and product. At the same time, an interesting phenomenon is that all the three views pay little attention to some common words like "," and function words "a" "of". This conforms to our intuition: some function words and punctuation marks contribute little to sentiment classification, but different words are focused on from different views. Thus, multiple ways of using user and product information can well gather information which is helpful to representing a sentence.

Figure 2(g) shows the three ways of attention to sentences for the example document. We can see that the three ways are consistent with focusing on the sixth sentence, which gives high attention weights to the words "five" and "stars" reflecting the sentiment polarity. From the user view, high attention is also paid to sentence5, which includes the obvious sentiment word "friendly". Hence, we infer that the user may like to express his/her sentiment to a product with sentiment word. From the product view, we can see that the document representation is also closely related to many other sentences such as sentence2 (w.r.t food) and sentence5 (w.r.t. staff), which can show different attributes of a product.

This case study shows that cascading two layers of multiway attention are necessary to modeling a document in the sentiment classification task.

4 Related work

Document-level sentiment classification methods can be divided into two kinds of research lines, i.e., traditional machine learning methods and neural networks methods.

For the first kind of research line, Pang et al. (2002) validate the effectiveness of various machine learning methods with bag-of-words features on sentiment classification. Goldberg and Zhu (2006) use a graph-based semi-supervised learning algorithm with unlabeled data to predict the sentiment of reviews. There are also some work which focus on extracting effective features. Ganu et al. (2009) identify user experience information from free text. Qu et al. (2010) introduce a kind of bag-of-opinion representation.



Figure 2: Case Study: Illustration of Attention Weights.

Kiritchenko et al. (2014) extract surface, semantic, and sentiment features from text. User information is also used in traditional methods. Tan et al. (2011) employ social relationships to improve user-level Twitter sentiment analysis. Gao et al. (2013) estimate the sentiment polarity by referring to the user leniency and product popularity computed during testing. Li et al. (2014) incorporate the user and product information into a topic model for sentiment analysis.

Recently, neural network approaches have achieved a comparable performance on documentlevel sentiment classification. Glorot et al. (2011) first propose a deep learning approach which learns to extract a meaningful representation for each review in an unsupervised fashion. Then, Socher et al. (2011, 2012, 2013) introduce recursive neural networks to document-level sentiment classification. Kim (2014) employ convolutional neural networks to model sentences with two kinds of embeddings for sentiment classification. Le and Mikolov (2014) introduce an unsupervised algorithm that learns fixed-length feature representations from variable-length pieces of texts. Tai et al. (2015) utilize tree-structured longshort memory networks to learn semantic representation for sentiment classification. In addition, user and product information are flexibly modeled for sentiment classification in the neural network methods (Tang et al., 2015b; Chen et al., 2016). Tang et al. (2015a) design preference matrices for each user and each product to tune word representations, based on which convolutional neural networks (CNNs) are used to model the whole document. Chen et al. (2016) employ two layers of long-short term memory (LSTM) with attention to model sentences and documents.

5 Conclusion

In this paper, we propose the cascading multiway attention networks for document-level sentiment classification. The main idea of CMA is to use multiway attention rather than one-way attention to learn representations for sentences and documents, as user, product and their combination can provide different views for modeling text and weaken the dependencies between sentences for better depicting the influence from user preference for document sentiment. On the sentence level, CMA uses three ways of exploiting user and product information to generate attention to words and get sentence representations. On the document level, CMA keeps using the multiway attention mechanism to generate attention to sentences. Experimental results on IMDB, Yelp 2013 and Yelp 2014 verify that CMA can learn efficient representations for sentences and documents and provide rich information for judging the document-level sentiment polarity.

Acknowledgments

We would like to thank the anonymous reviwers for thier insightful suggestions. Our work is supported by National High Technology Research and Development Program of China (863 Program) (No.2015AA015402) and National Natural Science Foundation of China (No.61370117 & No.61572049). The corresponding authors of

this paper are Houfeng Wang & Sujian Li.

References

- Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Jauvin. 2003. A neural probabilistic language model. *journal of machine learning research*, 3(Feb):1137–1155.
- Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings* of *EMNLP*.
- Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.
- Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 international conference on web search and data mining*, pages 231–240. ACM.
- Gayatree Ganu, Noemie Elhadad, and Amélie Marian. 2009. Beyond the stars: Improving rating predictions using review text content. In *WebDB*, volume 9, pages 1–6. Citeseer.
- Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Modeling user leniency and product popularity for sentiment classification. In *IJCNLP*, pages 1107–1111.
- Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale ssentiment classification: A deep learning approach. In Proceedings of the 28th International Conference on Machine Learning (ICML-11), pages 513–520.
- Andrew B Goldberg and Xiaojin Zhu. 2006. Seeing stars when there aren't many stars: graph-based semi-supervised learning for sentiment categorization. In *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing*, pages 45–52. Association for Computational Linguistics.
- Eric H Huang, Richard Socher, Christopher D Manning, and Andrew Y Ng. 2012. Improving word representations via global context and multiple word prototypes. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 873–882. Association for Computational Linguistics.
- Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. *arXiv preprint arXiv:1404.2188*.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

- Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. volume 50, pages 723–762.
- Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *ICML*, volume 14, pages 1188–1196.
- Fangtao Li, Sheng Wang, Shenghua Liu, and Ming Zhang. 2014. Suit: A supervised user-item based topic model for sentiment analysis. In *AAAI*, pages 1636–1642.
- Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David Mc-Closky. 2014. The stanford corenlp natural language processing toolkit. In ACL (System Demonstrations), pages 55–60.
- Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. 2010. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference* on Machine learning, pages 641–648. ACM.
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.
- Lizhen Qu, Georgiana Ifrim, and Gerhard Weikum. 2010. The bag-of-opinions method for review rating prediction from sparse text patterns. In *Proceedings* of the 23rd International Conference on Computational Linguistics, pages 913–921. Association for Computational Linguistics.
- Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1201–1211. Association for Computational Linguistics.
- Richard Socher, Jeffrey Pennington, Eric H Huang, Andrew Y Ng, and Christopher D Manning. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 151–161. Association for Computational Linguistics.

- Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, page 1642. Citeseer.
- Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.
- Kai Sheng Tai, Richard Socher, and Christopher D Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075*.
- Chenhao Tan, Lillian Lee, Jie Tang, Long Jiang, Ming Zhou, and Ping Li. 2011. User-level sentiment analysis incorporating social networks. In *Proceedings* of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1397–1405. ACM.
- Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1422–1432.
- Duyu Tang, Bing Qin, and Ting Liu. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *Proc. ACL*.
- Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Qin Bing. 2014. Learning sentiment specific word embedding for twitter sentiment classification. In *Proc. ACL*, pages 1555–1565. Association for Computational Linguistics.
- Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.
- Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. arXiv preprint arXiv:1212.5701.