Neural Sentiment Classification with User and Product Attention

Huimin Chen[1], **Maosong Sun**[1,2]*, **Cunchao Tu**[1], **Yankai Lin**[1], **Zhiyuan Liu**[1]
[1]Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, Beijing, China
[2]Beijing Advanced Innovation Center for Imaging Technology,
Capital Normal University, Beijing, China

Abstract

Document-level sentiment classification aims to predict user's overall sentiment in a document about a product. However, most of existing methods only focus on local text information and ignore the global user preference and product characteristics. Even though some works take such information into account, they usually suffer from high model complexity and only consider word-level preference rather than semantic levels. To address this issue, we propose a hierarchical neural network to incorporate global user and product information into sentiment classification. Our model first builds a hierarchical LSTM model to generate sentence and document representations. Afterwards, user and product information is considered via attentions over different semantic levels due to its ability of capturing crucial semantic components. The experimental results show that our model achieves significant and consistent improvements compared to all state-of-the-art methods. The source code of this paper can be obtained from `https://github.com/thunlp/NSC`.

1 Introduction

Sentiment analysis aims to analyze people's sentiments or opinions according to their generated texts and plays a critical role in the area of data mining and natural language processing. Recently, sentiment analysis draws increasing attention of researchers with the rapid growth of online review sites such as Amazon, Yelp and IMDB, due to its importance to personalized recommendation.

In this work, we focus on the task of document-level sentiment classification, which is a fundamental problem of sentiment analysis. Document-level sentiment classification assumes that each document expresses a sentiment on a single product and targets to determine the overall sentiment about the product.

Most existing methods take sentiment classification as a special case of text classification problem. Such methods treat annotated sentiment polarities or ratings as categories and apply machine learning algorithms to train classifiers with text features, e.g., bag-of-words vectors (Pang et al., 2002). Since the performance of text classifiers heavily depends on the extracted features, such studies usually attend to design effective features from text or additional sentiment lexicons (Ding et al., 2008; Taboada et al., 2011).

Motivated by the successful utilization of deep neural networks in computer vision (Ciresan et al., 2012), speech recognition (Dahl et al., 2012) and natural language processing (Bengio et al., 2006), some neural network based sentiment analysis models are proposed to learn low-dimensional text features without any feature engineering (Glorot et al., 2011; Socher et al., 2011; Socher et al., 2012; Socher et al., 2013; Kim, 2014). Most proposed neural network models take the text information in a sentence or a document as input and generate the semantic representations using well-designed neural networks. However, these methods only focus

*Corresponding author: M. Sun (sms@tsinghua.edu.cn) on the text content and ignore the crucial characteristics of users and products. It is a common sense that the user's preference and product's characteristics make significant influence on the ratings.

To incorporate user and product information into sentiment classification, (Tang et al., 2015b) bring in a text preference matrix and a representation vector for each user and product into CNN sentiment classifier. It modifies the word meaning in the input layer with the preference matrix and concatenates the user/product representation vectors with generated document representation before softmax layer. The proposed model achieves some improvements but suffers the following two problems: (1) The introduction of preference matrix for each user/product is insufficient and difficult to be well trained with limited reviews. For example, most users in IMDB and Yelp only have several tens of reviews, which is not enough to obtain a well-tuned preference matrix. (2) The characteristics of user and product should be reflected on the semantic level besides the word level. For example, a two star review in Yelp said "great place to grab a steak and I am a huge fan of the hawiian pizza $\cdots$ but I don't like to have to spend 100 bucks for a diner and drinks for two". It's obvious that the poor rating result mainly relies on the last sentence compared with others.

To address these issues, we propose a novel hierarchical LSTM model to introduce user and product information into sentiment classification. As illustrated in Fig. 1, our model mainly consists of two parts. Firstly, we build a hierarchical LSTM model to generate sentence-level representation and document-level representation jointly. Afterwards, we introduce user and product information as attentions over different semantic levels of a document. With the consideration of user and product information, our model can significantly improve the performance of sentiment classification in several real-world datasets.

To summarize, our effort provide the following three contributions:

(1) We propose an effective Neural Sentiment Classification model by taking global user and product information into consideration. Comparing with (Tang et al., 2015b), our model contains much less parameters and is more efficient for training.

(2) We introduce user and product based attentions over different semantic levels of a document. Traditional attention-based neural network models only take the local text information into consideration. In contrast, our model puts forward the idea of user-product attention by utilizing the global user preference and product characteristics.

(3) We conduct experiments on several real-world datasets to verify the effectiveness of our model. The experimental results demonstrate that our model significantly and consistently outperforms other state-of-the-art models.

2 Related Work

With the trends of deep learning in computer vision, speech recognition and natural language processing, neural models are introduced into sentiment classification field due to its ability of text representation learning. (Glorot et al., 2011) use Stacked Denoising Autoencoder in sentiment classification for the first time. Socher conducts a series of recursive neural network models to learn representations based on the recursive tree structure of sentences, including Recursive Autoencoder (RAE) (Socher et al., 2011), Matrix-Vector Recursive Neural Network (MV-RNN) (Socher et al., 2012) and Recursive Neural Tensor Network (RNTN) (Socher et al., 2013). Besides, (Kim, 2014) and (Johnson and Zhang, 2014) adopt convolution neural network (CNN) to learn sentence representations and achieve outstanding performance in sentiment classification.

Recurrent neural network also benefits sentiment classification becuase it is capable of capturing the sequential information. (Li et al., 2015), (Tai et al., 2015) investigate tree-structured long-short term memory (LSTM) networks on text or sentiment classification. There are also some hierarchical models proposed to deal with document-level sentiment classification (Tang et al., 2015a; Bhatia et al., 2015), which generate different levels (e.g., phrase, sentence or document) of semantic representations within a document. Moreover, attention mechanism is also introduced into sentiment classification, which aims to select important words from a sentence or sentences from a document (Yang et al.,

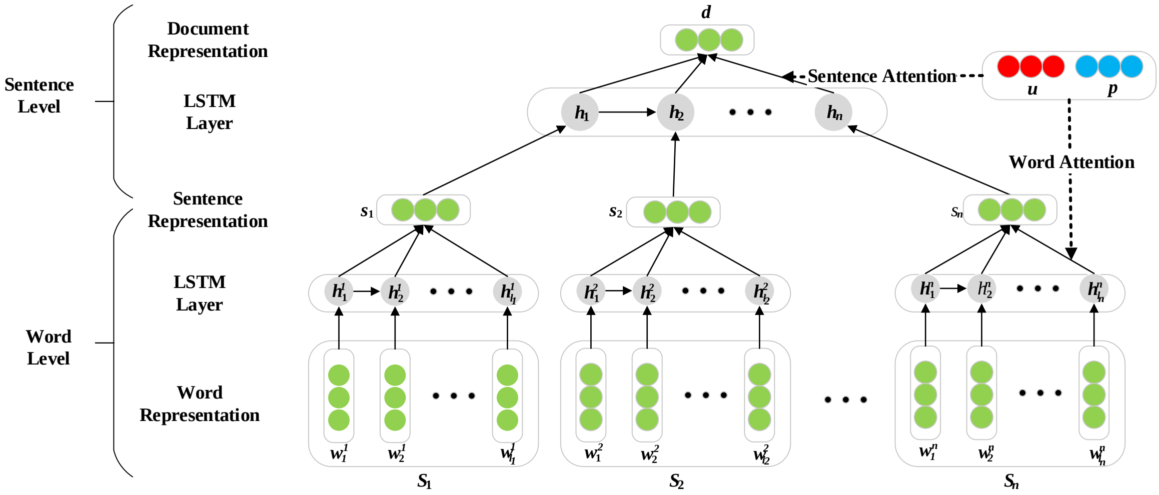


Figure 1: The architecture of User Product Attention based Neural Sentiment Classification model.

2016).

Most existing sentiment classification models ignore the global user preference and product characteristics, which have crucial effects on the sentiment polarities. To address this issue, (Tang et al., 2015b) propose to add user/product preference matrices and representation vectors into CNN models. Nevertheless, it suffers from high model complexity and only considers word-level preference rather than semantic levels. In contrast, we propose an efficient neural sentiment classification model with users and products to serve as attentions in both word and semantic levels.

3 Methods

In this section, we will introduce our User Product Attention (UPA) based Neural Sentiment Classification (NSC) model in detail. First, we give the formalizations of document-level sentiment classification. Afterwards, we discuss how to obtain document semantic representation via the Hierarchical Long Short-term Memory (HLSTM) network . At last, we present our attention mechanisms which incorporates the global information of users and products to enhance document representations. The enhanced document representation is used as features for sentiment classification. An overall illustration of UPA based NSC model is shown in Fig. 1.

3.1 Formalizations

Suppose a user $u \in U$ has a review about a product $p \in P$. We represent the review as a document d with n sentences $\{\mathbf{S}_1, \mathbf{S}_2, \cdots, \mathbf{S}_n\}$. Here, l_i is the length of i-th sentence. The i-th sentence $\mathbf{S}_i$ consists of l_i words as $\{w_1^i, w_2^i, \cdots, w_{l_i}^i\}$. Document-level sentiment classification aims to predict the sentiment distributions or ratings of these reviews according to their text information.

3.2 Neural Sentiment Classification Model

According to the principle of compositionality (Frege, 1892), we model the semantic of a document through a hierarchical structure composed of word-level, sentence-level and document-level. To model the semantic representations of sentences, we adopt Long Short-Term Memory (LSTM) network because of its excellent performance on sentiment classification, especially for long documents. Similarly, we also use LSTM to learn document representations.

In word level, we embed each word in a sentence into a low dimensional semantic space. That means, each word w_j^i is mapped to its embedding $\mathbf{w}_j^i \in \mathbb{R}^d$. At each step, given an input word $\mathbf{w}_j^i$, the current cell state $\mathbf{c}_j^i$ and hidden state $\mathbf{h}_j^i$ can be updated with the previous cell state $\mathbf{c}_{j-1}^i$ and hidden state $\mathbf{h}_{j-1}^i$ as follows:

$$\begin{bmatrix} \mathbf{i}_j^i \\ \mathbf{f}_j^i \\ \mathbf{o}_j^i \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \end{bmatrix} (\mathbf{W} \cdot [\ \mathbf{h}_{j-1}^i, \mathbf{w}_j^i\ ] + \mathbf{b}), \quad (1)$$

$$\hat{\mathbf{c}}_j^i = \tanh(\mathbf{W} \cdot [\ \mathbf{h}_{j-1}^i, \mathbf{w}_j^i\ ] + \mathbf{b}), \quad (2)$$

$$\mathbf{c}_j^i = \mathbf{f}_j^i \odot \mathbf{c}_{j-1}^i + \mathbf{i}_j^i \odot \hat{\mathbf{c}}_j^i, \quad (3)$$

$$\mathbf{h}_j^i = \mathbf{o}_j^i \odot \tanh(\mathbf{c}_j^i), \quad (4)$$

where $\mathbf{i}, \mathbf{f}, \mathbf{o}$ are gate activations, $\odot$ stands for element-wise multiplication, σ is sigmoid function, $\mathbf{W}, \mathbf{b}$ are the parameters we need to train. We then feed hidden states $[\mathbf{h}_1^i, \mathbf{h}_2^i, \cdots, \mathbf{h}_{l_i}^i]$ to an average pooling layer to obtain the sentence representation $\mathbf{s}_i$.

In sentence level, we also feed the sentence embeddings $[\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_n]$ into LSTM and then obtain the document representation $\mathbf{d}$ through an average pooling layer in a similar way.

3.3 User Product Attention

We bring in User Product Attention to capture the crucial components over different semantic levels for sentiment classification. Specifically, we employ word-level UPA to generate sentence representations and sentence-level UPA to obtain document representation. We give the detailed implementations in the following parts.

It is obvious that not all words contribute equally to the sentence meaning for different users and products. Hence, in word level, instead of feeding hidden states to an average pooling layer, we adopt a user product attention mechanism to extract user/product specific words that are important to the meaning of sentence. Finally, we aggregate the representations of those informative words to form the sentence representation. Formally, the enhanced sentence representation is a weighted sum of hidden states as:

$$\mathbf{s}_i = \sum_{j=1}^{l_i} \alpha_j^i \mathbf{h}_j^i, \quad (5)$$

where α_j^i measures the importance of the j-th word for current user and product. Here, we embed each user u and each product p as continuous and real-valued vectors $\mathbf{u} \in \mathbb{R}^{d_u}$ and $\mathbf{p} \in \mathbb{R}^{d_p}$, where d_u and d_p are the dimensions of user embeddings and product embeddings respectively. Thus, the attention weight α_j^i for each hidden state can be defined as:

$$\alpha_j^i = \frac{\exp(e(\mathbf{h}_j^i, \mathbf{u}, \mathbf{p}))}{\sum_{k=1}^{l_i} \exp(e(\mathbf{h}_k^i, \mathbf{u}, \mathbf{p}))}, \quad (6)$$

where e is a score function which scores the importance of words for composing sentence representation. The score function e is defined as:

$$\begin{aligned} & e(\mathbf{h}_j^i, \mathbf{u}, \mathbf{p}) = \\ & \mathbf{v}^{\mathrm{T}} \tanh(\mathbf{W}_H \mathbf{h}_{ij} + \mathbf{W}_U \mathbf{u} + \mathbf{W}_P \mathbf{p} + \mathbf{b}), \end{aligned} \quad (7)$$

where $\mathbf{W}_H$, $\mathbf{W}_U$ and $\mathbf{W}_P$ are weight matrices, $\mathbf{v}$ is weight vector and $\mathbf{v}^{\mathrm{T}}$ denotes its transpose.

The sentences that are clues to the meaning of the document vary in different users and products. Therefore, in sentence level, we also use a attention mechanism with user vector $\mathbf{u}$ and product vector $\mathbf{p}$ in word level to select informative sentences to compose the document representation. The document representation $\boldsymbol{d}$ is obtained via:

$$\mathbf{d} = \sum_{i=1}^{n} \beta_i \mathbf{h}_i, \quad (8)$$

where β_i is the weight of hidden state $\boldsymbol{h_i}$ in sentence level which can be calculated similar to the word attention.

3.4 Sentiment Classification

Since document representation $\boldsymbol{d}$ is hierarchically extracted from the words and sentences in the documents, it is a high level representation of the document. Hence, we regard it as features for document sentiment classification. We use a non-linear layer to project document representation $\boldsymbol{d}$ into the target space of C classes:

$$\hat{\mathbf{d}} = \tanh(\mathbf{W}_c \mathbf{d} + \mathbf{b}_c). \quad (9)$$

Afterwards, we use a softmax layer to obtain the document sentiment distribution:

$$p_c = \frac{\exp(\hat{d}_c)}{\sum_{k=1}^{C} \exp(\hat{d}_k)}, \quad (10)$$

where C is the number of sentiment classes, p_c is the predicted probability of sentiment class c. In

Datasets	#classes	#docs	#users	#products	#docs/user	#docs/product	#sens/doc	#words/sen
IMDB	10	84,919	1,310	1,635	64.82	51.94	16.08	24.54
Yelp 2014	5	231,163	4,818	4,194	47.97	55.11	11.41	17.26
Yelp 2013	5	78,966	1,631	1,633	48.42	48.36	10.89	17.38

Table 1: Statistics of IMDB, Yelp2013 and Yelp2014 datasets

our model, cross-entropy error between gold sentiment distribution and our model's sentiment distribution is defined as loss function for optimization when training:

$$L = -\sum_{d \in D} \sum_{c=1}^{C} p_c^g(d) \cdot \log(p_c(d)), \qquad (11)$$

where p_c^g is the gold probability of sentiment class c with ground truth being 1 and others being 0, D represents the training documents.

4 Experiments

In this section, we introduce the experimental settings and empirical results on the task of document-level sentiment classification.

4.1 Experimental Settings

We evaluate the effectiveness of our NSC model on three sentiment classification datasets with user and product information: IMDB, Yelp 2013 and Yelp 2014, which are built by (Tang et al., 2015b). The statistics of the datasets are summarized in Table 1. We split the datasets into training, development and testing sets in the proportion of 8:1:1, with tokenization and sentence splitting by Stanford CoreNLP (Manning et al., 2014). We use two metrics including $Accuracy$ which measures the overall sentiment classification performance and $RMSE$ which measures the divergences between predicted sentiment classes and ground truth classes. The $Accuracy$ and $RMSE$ metrics are defined as:

$$Accuracy = \frac{T}{N} \qquad (12)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(gd_i - pr_i)^2}{N}}, \qquad (13)$$

where T is the numbers of predicted sentiment ratings that are identical with gold sentiment ratings, N is the numbers of documents and gd_i, pr_i represent the gold sentiment rating and predicted sentiment rating respectively.

Word embeddings could be initialized in several ways, such as random initialization, pre-training on each dataset in (Tang et al., 2015b) or on the larger ones in (Tang et al., 2015a). We pre-train the 200-dimensional word embeddings on each dataset in (Tang et al., 2015a) with SkipGram (Mikolov et al., 2013). We set the user embedding dimension and product embedding dimension to be 200, initialized to zero. The dimensions of hidden states and cell states in our LSTM cells are set to 200. We tune the hyper parameters on the development sets and use AdaDelta (Zeiler, 2012) to update parameters when trainning. We select the best configuration based on performance on the development set, and evaluate the configuration on the test set.

4.2 Baselines

We compare our NSC model with several baseline methods for document sentiment classification:

Majority regards the majority sentiment category in training set as the sentiment category of each document in test set.

Trigram trains a SVM classifier with unigrams, bigrams and trigrams as features.

TextFeature extracts text features including word and character n-grams, sentiment lexicon features, etc, and then train a SVM classifier.

UPF extracts use-leniency features (Gao et al., 2013) and corresponding product features from training data, which is further concatenated with the features in **Trigram** an **TextFeature**.

AvgWordvec averages word embeddings in a document to obtain document representation which is fed into a SVM classifier as features.

SSWE generates features with sentiment-specific word embeddings (SSWE) (Tang et al., 2014) and then trains a SVM classifier.

RNTN + RNN represents each sentence with the

Models	IMDB		Yelp2013		Yelp2014	
	Acc.	RMSE	Acc.	RMSE	Acc.	RMSE
Models without user and product information						
Majority	0.196	2.495	0.411	1.060	0.392	1.097
Trigram	0.399	1.783	0.569	0.814	0.577	0.804
TextFeature	0.402	1.793	0.556	0.845	0.572	0.800
AvgWordvec + SVM	0.304	1.985	0.526	0.898	0.530	0.893
SSWE + SVM	0.312	1.973	0.549	0.849	0.557	0.851
Paragraph Vector	0.341	1.814	0.554	0.832	0.564	0.802
RNTN + Recurrent	0.400	1.764	0.574	0.804	0.582	0.821
UPNN (CNN and no UP)	0.405	1.629	0.577	0.812	0.585	0.808
NSC	0.443	1.465	0.627	**0.701**	**0.637**	**0.686**
NSC + LA	**0.487**	**1.381**	**0.631**	0.706	0.630	0.715
Models with user and product information						
Trigram + UPF	0.404	1.764	0.570	0.803	0.576	0.789
TextFeature + UPF	0.402	1.774	0.561	1.822	0.579	0.791
JMARS	N/A	1.773	N/A	0.985	N/A	0.999
UPNN (CNN)	0.435	1.602	0.596	0.784	0.608	0.764
UPNN (NSC)	0.471	1.443	0.631	0.702	N/A	N/A
NSC+UPA	**0.533**	**1.281**	**0.650**	**0.692**	**0.667**	**0.654**

Table 2: Document-level sentiment classification results. Acc.(Accuracy) and RMSE are the evaluation metrics. The best performances are in bold in both groups.

Recursive Neural Tensor Network (RNTN) (Socher et al., 2013) and feeds sentence representations into the Recurrent Neural Network (RNN). Afterwards, the hidden vectors of RNN are averaged to obtain document representation for sentiment classification.

Paragraph Vector implements the PVDM (Le and Mikolov, 2014) for document sentiment classification.

JMARS considers the information of users and aspects with collaborative filtering and topic modeling for document sentiment classification.

UPNN brings in a text preference matrix and a representation vector for each user and product into CNN sentiment classifier (Kim, 2014). It modifies the word meaning in the input layer with the preference matrix and concatenates the user/product representation vectors with generated document representation before softmax layer.

For all baseline methods above, we report the results in (Tang et al., 2015b) since we use the same datasets.

4.3 Model Comparisons

We list the experimental results in Table 2. As shown in this table, we manually divide the results into two parts, the first one of which only considers the local text information and the other one incorporates both local text information and the global user product information.

From the first part in Table 2, we observe that NSC, the basic implementation of our model, significantly outperforms all the other baseline methods which only considers the local text information. To be specific, NSC achieves more than 4% improvements over all datasets compared to typical well-designed neural network models. It demonstrates that NSC is effective to capture the sequential information, which can be a crucial factor to sentiment classification. Moreover, we employ the idea of local semantic attention (LA) in (Yang et al., 2016) and implement it in NSC model (denoted as NSC+LA). The results shows that the attention based NSC obtains a considerable improvements than the original one. It proves the importance of selecting more meaningful words and sentences in sentiment classification, which is also a main reason of introducing global user and product information in an attention form.

In the second part of Table 2, we show the performance of models with user product information.

Basic Model	Level		IMDB		Yelp2013		Yelp2014	
	Word	Sentence	Acc	RMSE	Acc	RMSE	Acc	RMSE
NSC	AVG	AVG	0.443	1.465	0.627	0.701	0.637	0.686
	AVG	ATT	0.498	1.336	0.632	0.701	0.653	0.672
	ATT	AVG	0.513	1.330	0.640	**0.686**	0.662	0.657
	ATT	ATT	**0.533**	**1.281**	**0.650**	0.692	**0.667**	**0.654**

Table 3: Effect of attention mechanisms in word and sentence level. AVG means an average pooling layer, and ATT represents the attention mechanism in word or sentence level.

Basic Model	Attention Type	IMDB		Yelp2013		Yelp2014	
		Acc	RMSE	Acc	RMSE	Acc	RMSE
NSC	ATT	0.487	1.381	0.631	0.706	0.630	0.715
	PA	0.485	1.456	0.630	0.704	0.644	0.676
	UA	0.525	**1.276**	0.645	0.699	0.644	0.680
	UPA	**0.533**	1.281	**0.650**	**0.692**	**0.667**	**0.654**

Table 4: Effect of user and product attention mechanisms. UA represents the user attention mechanism, and PA indicates the product attention mechanism.

From this part, we have the following observations:

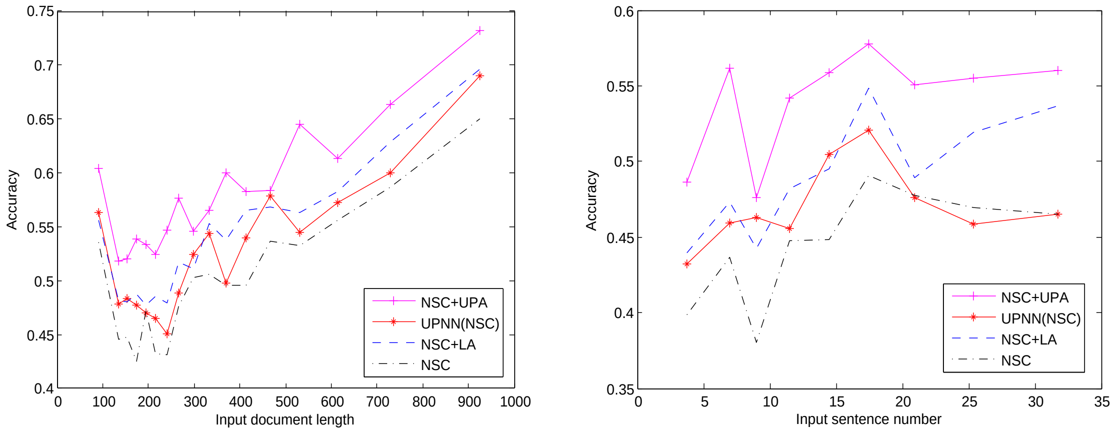
(1) The global user and product information is helpful to neural network based models for sentiment classification. With the consideration of such information in IMDB, UPNN achieves 3% improvement and our proposed NSC+UPA obtains 9% improvement in accuracy. The significant improvements state the necessity of considering these global information in sentiment classification.

(2) Our proposed NSC model with user production attention (NSC+UPA) significantly and consistently outperforms all the other baseline methods. It indicates the flexibility of our model on various real-world datasets. Note that, we also implement (Tang et al., 2015b)'s method to deal with user and product information on NSC (denoted as UPNN (NSC)). Though the employment of NSC improves the performance of UPNN, it is still not comparable to our model. More specifically, UPNN exceed the memory of our GPU (12G) when dealing with Yelp2014 dataset due to the high complexity of its parameters. Compared to UPNN which utilizes the user product information with matrices and vectors simultaneously, our model only embeds each user and product as a vector, which makes it suitable to large-scale datasets. It demonstrates that our NSC model is more effective and efficient to handle additional user and product information.

Observations above demonstrate that NSC with user product attention (NSC+UPA) is capable of capturing meanings of multiple semantic layers within a document. Comparing with other user product based models, our model incorporates global user product information in an effective and efficient way. Furthermore, the model is also robust and achieves consistent improvements than state-of-the-art methods on various real-world datasets.

4.4 Model Analysis: Effect of Attention Mechanisms in Word and Sentence Level

Table 3 shows the effect of attention mechanisms in word or sentence level respectively. From the table, we can observe that: (1) Both the attention mechanisms applied in word level and sentence level improve the performance for document sentiment classification compared with utilizing average pooling in word and sentence level; (2) The attention mechanism in word level improves more for our model as compared to sentence level. The reason is that the word attention mechanism can capture the informative words in all documents, while the sentence attention mechanism may only work in long documents with various topics. (3) The model considering both word level attention and sentence level attention outperforms the ones considering only one semantic level attention. It proves that the characteristics of users and products are reflected on multiple semantic levels, which is also a critical mo-



(a) Accuracy over document length

(b) Accuracy over sentence number

Figure 2: Accuracy over various input document lengths on IMDB test set

tivation of introducing User Product Attention into sentiment classification.

4.5 Model Analysis: Effect of User Product Attention Mechanisms

Table 4 shows the performance of attention mechanisms with the information of users or products. From the table, we can observe that:

(1) The information of both users and products contributes to our model as compared to a semantic attention. It demonstrates that our attention mechanism can catch the specific characteristic of a user or a product.

(2) The information of users is more effective than the products to enhance document representations. Hence, the discrimination of user preference is more obvious than product characteristics.

4.6 Model Analysis: Performance over Sentence Numbers and Lengths

To investigate the performance of our model over documents with various lengths, we compare the performance of different implementations of NSC under different document lengths and sentence number settings. Fig. 2 shows the accuracy of sentiment classification generated by NSC, NSC+ATT, UPNN(NSC) and NSC+UPA on the IMDB test set with respect to input document lengths and input sentence numbers in a document. From Fig. 2, we observe that our model NSC with attention mechanism of user and product information consistently outperforms other baseline methods for all input document lengths and sentence numbers. It indicates the robustness and flexibility of NSC on different datasets.

4.7 Case Study

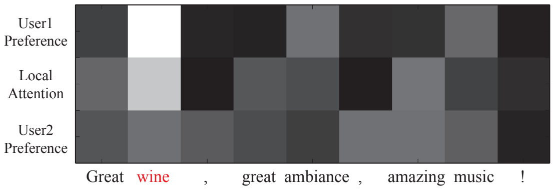


Figure 3: Visualization of attentions over words

To demonstrate the effectiveness of our global attention, we provide a review instance in Yelp2013 dataset for example. The content of this review is "Great wine, great ambiance, amazing music!". We visualize the attention weights in word-level for two distinct users and the local semantic attention (LA) in Fig 3. Here, the local semantic attention represents the implementation in (Yang et al., 2016), which calculates the attention without considering the global information of users and products. Note that, darker color means lower weight.

According to our statistics, the first user often mentions "wine" in his/her review sentences. On the contrary, the second user never talks about "wine" in his/her review sentences. Hence, we infer that the first user may has special preference to wine while the second one has no concern about wine. From the figure, we observe an interesting phenomenon which confirms to our inference. For the word "wine", the first user has the highest attention weight and the second user has the lowest attention weight. It indicates that our model can capture the global user preference via our user attention.

5 Conclusion and Future Work

In this paper, we propose a hierarchical neural network which incorporates user and product information via word and sentence level attentions. With the user and product attention, our model can take account of the global user preference and product characteristics in both word level and semantic level. In experiments, we evaluate our model on sentiment analysis task. The experimental results show that our model achieves significant and consistent improvements compared to other state-of-the-art models.

We will explore more in future as follows:

(1) In this paper, we only consider the global user preference and product characteristics according to their personal behaviors. In fact, most users and products usually have some text information such as user and product profiles. We will take advantages of those information in sentiment analysis in future.

(2) Aspect level sentiment classification is also a fundamental task in the field of sentiment analysis. The user preference and product characteristics may also implicitly influence the sentiment polarity of the aspect. We will explore the effectiveness of our model on aspect level sentiment classification.

6 Acknowledgements

This work is supported by the National Social Science Foundation of China (13&ZD190) and the National Natural Science Foundation of China (NSFC No. 61331013). We sincerely thank Shiqi Shen and Lei Xu for their insightful discussions, and thank Ayana, Yu Zhao, Ruobing Xie, Jiacheng Zhang and Meng Zhang in Tsinghua University Natural Language Processing group for their constructive comments. We also thank all anonymous reviewers for their insightful suggestions.

References

Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin, and Jean-Luc Gauvain. 2006. Neural probabilistic language models. In *Innovations in Machine Learning*, pages 137–186. Springer.

Parminder Bhatia, Yangfeng Ji, and Jacob Eisenstein. 2015. Better document-level sentiment analysis from rst discourse parsing. In *Proceedings of EMNLP*.

Dan Ciresan, Ueli Meier, and Jürgen Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *Proceedings of CVPR*, pages 3642–3649. IEEE.

George E Dahl, Dong Yu, Li Deng, and Alex Acero. 2012. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Trans. Audio, Speech, and Language Processing*, 20(1):30–42.

Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of WSDM*, pages 231–240. ACM.

Gottlob Frege. 1892. On sense and reference. In *Ludlow*.

Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Modeling user leniency and product popularity for sentiment classification. In *Proceedings of IJCNLP*, pages 1107–1111.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Domain adaptation for large-scale sentiment classification: A deep learning approach. In *Proceedings of ICML*, pages 513–520.

Rie Johnson and Tong Zhang. 2014. Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*.

Quoc V Le and Tomas Mikolov. 2014. Distributed representations of sentences and documents. In *Proceedings of ICML*.

Jiwei Li, Dan Jurafsky, and Eudard Hovy. 2015. When are tree structures necessary for deep learning of representations? *arXiv preprint arXiv:1503.00185*.

Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The stanford corenlp natural language processing toolkit. In *Proceedings of ACL*, pages 55–60.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of EMNLP*, pages 79–86.

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 EMNLP*, pages 151–161.

Richard Socher, Brody Huval, Christopher D Manning, and Andrew Y Ng. 2012. Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of EMNLP*, pages 1201–1211.

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 EMNLP*, page 1642.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *CL*, 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. In *Proceedings of ACL*.

Duyu Tang, Furu Wei, Nan Yang, Ming Zhou, Ting Liu, and Bing Qin. 2014. Learning sentiment-specific word embedding for twitter sentiment classification. In *Proceedings of ACL*, pages 1555–1565.

Duyu Tang, Bing Qin, and Ting Liu. 2015a. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of EMNLP*, pages 1422–1432.

Duyu Tang, Bing Qin, and Ting Liu. 2015b. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of ACL*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings NAACL*.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.