Incorporating Multi-Level User Preference into Document-Level Sentiment Classification

JUNJIE LI, HAORAN LI, and XIAOMIAN KANG, National Laboratory of Pattern Recognition, Institute of Automation, University of Chinese Academy of Sciences, Chinese Academy of Sciences, China
HAITONG YANG, School of Computer, Central China Normal University, China
CHENGQING ZONG, National Laboratory of Pattern Recognition, Institute of Automation, University of Chinese Academy of Sciences, CAS Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, China

Document-level sentiment classification aims to predict a user's sentiment polarity in a document about a product. Most existing methods only focus on review contents and ignore users who post reviews. In fact, when reviewing a product, different users have different word-using habits to express opinions (i.e., word-level user preference), care about different attributes of the product (i.e., aspect-level user preference), and have different characteristics to score the review (i.e., polarity-level user preference). These preferences have great influence on interpreting the sentiment of text. To address this issue, we propose a model called Hierarchical User Attention Network (HUAN), which incorporates multi-level user preference into a hierarchical neural network to perform document-level sentiment classification. Specifically, HUAN encodes different kinds of information (word, sentence, aspect, and document) in a hierarchical structure and imports user embedding and user attention mechanism to model these preferences. Empirical results on two real-world datasets show that HUAN achieves state-of-the-art performance. Furthermore, HUAN can also mine important attributes of products for different users.

CCS Concepts: • **Information systems → Sentiment analysis**;

Additional Key Words and Phrases: Sentiment classification, deep learning, user preference, hierarchical attention network

ACM Reference format:
Junjie Li, Haoran Li, Xiaomian Kang, Haitong Yang, and Chengqing Zong. 2018. Incorporating Multi-Level User Preference into Document-Level Sentiment Classification. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.* 18, 1, Article 7 (November 2018), 17 pages.
https://doi.org/10.1145/3234512

The research work described in this article has been supported by the National Key Research and Development Program of China under Grant No. 2017YFB1002103.

Authors' addresses: J. Li, H. Li, and X. Kang, Intelligence Building, No. 95, Zhongguancun East Road, Haidian District, Beijing, 100190, China; emails: {junjie.li, haoran.li, xiaomian.kang}@nlpr.ia.ac.cn; H. Yang, NO.152 Luoyu Road, Wuhan, HuBei, 430079, China; email: htyang@mail.ccnu.edu.cn; C. Zong (corresponding author), Intelligence Building, No. 95, Zhongguancun East Road, Haidian District, Beijing, 100190, China; email: cqzong@nlpr.ia.ac.cn.

2375-4699/2018/11-ART7 $15.00
https://doi.org/10.1145/3234512

1 INTRODUCTION

The emergence of online consumer review platforms, such as Tripadvisor[1] and Yelp[2], allow users to express their opinions on a wide variety of products and services. The popularity of such platforms has resulted in large amounts of online reviews created by different users. These reviews are useful to customers for getting a better understanding of products and to merchants for improving their products and services. However, the volume of reviews grows so rapidly that it is difficult to mine needed information from these reviews manually. Much work in sentiment analysis has been done to alleviate this problem, including sentiment classification [16–20, 35, 41, 42, 44], opinion summarization [8, 39], and automatic extraction of aspects [9, 40].

This article focuses on the task of document-level sentiment classification, which is a fundamental problem of sentiment analysis. This task is to predict a user's overall sentiment polarity of a document about a product [21, 24].

Motivated by successful applications of deep neural networks in computer vision [3], speech recognition [5], and natural language processing [4], many models [13, 30, 37] based on neural networks are proposed to perform sentiment classification. These models take a review as input, generate its semantic representation using well-designed neural networks, and classify it based on the representation. Even though these methods obtain good performance, they only focus on the text content and ignore users who post these reviews. Actually, users are very important factors in determining the sentiment polarity of reviews, which contains word-level user preference (WrdUP), aspect-level user preference (AspUP), and polarity-level user preference (PolUP). Table 1 presents reviews, with respect to 1–5 rating scales, posted by two users (*User1* and *User2*) in our dataset to show these preferences:

- *WrdUP*: Different users have different word-using habits to express opinions. "Good" is a positive word and should often appear in high-rating (such as 4-star or 5-star) reviews and *User2* frequently accords with the habit, while *User1* often violates the habit. For example, although "good" appears in *User1*'s first sample, the overall score is only 2-star. Actually, "good" in this case is in a sarcasm style.
- *AspUP*: When scoring a product (such as "hotel"), different users care about different aspects, where aspects refers to a product's or service's properties (or attributes), such as "service" and "price." Identifying important aspects for each user is beneficial to score reviews posted by them. From Table 1, we can find *User1* cares about service more than *User2*, because *User1* often comments "service," and service has a strong correlation with the overall score of *User1*'s review, while *User2* rarely comments "service" and the correlation is negligible for him/her.
- *PolUP*: Different users have different characteristics in scoring reviews. Table 1 shows that *User1* is a critical user and often writes reviews in low-rating intervals (such as 1-star or 2-star), while *User2* is a lenient one and always posts reviews with high ratings (such as 4-star or 5-star). The average score of *User1* and *User2* in training sets are 1.96 and 4.32, respectively.

A model that is agnostic to user differences will lose these preferences and performance suffers. Recently, some models [2, 6, 35, 36] have incorporated user information into sentiment classification; however, they only consider such information partially (Table 2). First, they all obtain review representation directly from words or sentences and ignore aspects in modeling reviews. Considering aspects in modeling reviews can get better review representation and boost document-level

[1]http://www.tripadvisor.com/.
[2]https://www.yelp.com/.

Table 1. Samples in Tripadvisor, a Dataset Used in This Article (Section 3.1), Show Multi-Level User Preference

User	Score	Text
User1	2	... The place is really **good** with nothing in the area. The <u>service</u> is also <u>terrible</u>. ...
	2	... That is all of the **good** things about this hotel. <u>Bad service</u>: unfriendly staff and only one person at the reception. ...
	5	... Cake tastes delicious, very <u>friendly staff</u> and **good** happy hour ....
User2	5	... The food is very **good** and decently priced for here. ...
	4	... Massive bed, excellent shower, and **good** view. ...
	5	... Although the <u>service is not that perfect</u>, this hotel is also very **good**, it contains excellent location and reasonable price! ...

The score range of these reviews are 1-5. Bold **words** in review text, underlined <u>words</u> in review text and review score show **WrdUP**, **AspUP**, and **PolUP** respectively.

Table 2. Comparison of Various Approaches for Incorporating User Information into Sentiment Classification

	WrdUP	**AspUP**	**PolUP**
Tang et al. (35)	✓	–	✓
Tang et al. (36)	✓	–	–
Chen et al. (2)	✓	–	–
Dou (6)	✓	–	–
HUAN	✓	✓	✓

"✓" denotes the specific preference is considered in model, while "–" denotes not.

sentiment classification (Section 3). Second, Tang et al. [35, 36] use matrix and vector to represent users to consider WrdUP and PolUP, and integrate them into a convolution neural network for review rating prediction. However, it is hard to train with limited reviews, especially for user matrix [2]. Third, although Chen et al. (2) and Dou (6) represent users as a vector to consider WrdUP and merge them into a hierarchical LSTM model or deep memory network to perform sentiment classification, they ignore AspUP and PopUP.

To fully take user information into consideration, we propose a model called Hierarchical User Attention Network (HUAN), which has three characteristics: (1) Inspired by other hierarchical models [2, 45], HUAN also utilizes a hierarchical structure to encode different kinds of information from word-level, sentence-level, and aspect-level to document-level. The main difference between HUAN and other hierarchical models is that HUAN contains an aspect-level representation layer. In classical hierarchical structures, they model review as a combination of sentences and ignore aspects. In fact, review contains a user's attitudes to aspects, therefore we model review as a combination of aspects. (2) HUAN introduces user information as attentions over word-level representation and aspect-level representation to consider WrdUP and AspUP. (3) To consider PolUP, our model generates document representation by combining user and document information, and utilizes this representation for classification.

In summary, our main contributions are as follows:

— We propose a model (HUAN) to fully incorporate user information into document-level sentiment classification, and consider WrdUP, AspUP, and PolUP jointly (Section 2).

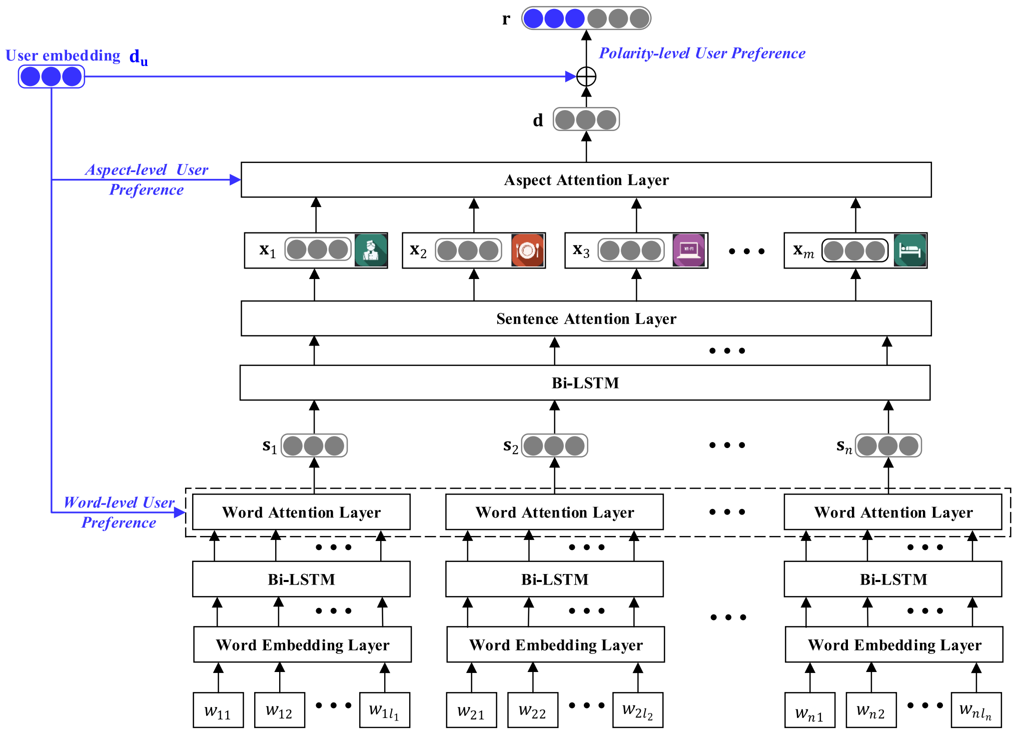


Fig. 1. The architecture of HUAN. Given a review, HUAN gets word-level, sentence-level, aspect-level, and document-level representation in turn. Sample aspects in the figure are service, food, facility, and room. To consider *WrdUP* and *AspUP*, HUAN introduces user information as attentions over word-level representation and aspect-level representation. To incorporate *PolUP*, HUAN gets review representation by concatenating user embedding and document-level representation.

- Different from other hierarchical structures [2, 45], HUAN imports an aspect-level representation layer and experiments on two datasets demonstrate this layer is useful for document-level sentiment classification (Section 3).
- We conduct experiments on two real-world datasets to verify the effectiveness of HUAN. The experimental results show that HUAN outperforms state-of-the-art methods significantly (Section 3). Furthermore, HUAN can also mine the important aspects for different users (Section 4).

2 HIERARCHICAL USER ATTENTION NETWORK

The overall architecture of HUAN is shown in Figure 1. It consists of six parts: a word sequence encoder, a word-level attention layer, a sentence sequence encoder, a sentence-level attention layer, an aspect-level attention layer, and a review representation layer. Table 3 provides a summary of notations used in this article.

Suppose we have a corpus D about a specific domain (such as hotel) and m pre-defined aspects $\{a_1, a_2, \ldots, a_m\}$, such as service and location. Detailed information about the pre-defined aspects is presented in Section 3.2. Review d is a sample of D and its author is d_u. There are n sentences in d and each sentence s_i is labeled by a set of aspects using *Aspect Segmentation* algorithm [38], which is shown in Section 3.2. The primary goal of HUAN is to correctly classify document d. In the following sections, we describe the details of different components.

Table 3. Summary of Notations Used in This Article

Symbol	Description
D	review corpus.
C	the number of sentiment labels in D.
d	a sample review.
g_d	the ground truth label for review d.
m	the number of all aspects in D.
n	the number of sentence in d.
a_i	an aspect $i \in \{1, 2, \ldots, m\}$.
$d_u, \mathbf{d}_u$	the author of d and its embedding.
$s_i, \mathbf{s}_i$	a sentence in d and its representation, $i \in \{1, 2, \ldots, n\}$.
l_i	the number of words in s_i, $i \in \{1, 2, \ldots, n\}$.
$w_{ij}, \mathbf{w}_{ij}$	a word in s_i and its embedding, $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, l_i\}$.
A	sentence-aspect matrix $\in \mathbb{R}^{n \times m}$, if s_i is assigned to a_j $A_{ij} = 1$, otherwise $A_{ij} = 0$.
$\mathbf{h}_{ij}$	hidden representation of w_{ij} in d, $i \in \{1, 2, \ldots, n\}$ and $j \in \{1, 2, \ldots, l_i\}$.
$\mathbf{h}_i$	hidden representation of s_i in d, $i \in \{1, 2, \ldots, n\}$.
$\mathbf{x}_i$	aspect representation of a_i in d, $i \in \{1, 2, \ldots, m\}$.
$\mathbf{d}$	document representation of d.
$\mathbf{r}_d$	review representation of d.
α_{ij}, β_{ij}	attention weights at word-level and sentence-level.
γ_i	attention weight at aspect-level in d.

2.1 LSTM-Based Sequence Encoder

Long short-term memory network (LSTM) [10] is a special form of recurrent neural networks (RNNs), which processes sequence data and alleviates the problem of gradient diffusion and explosion. LSTM can capture the long dependencies in a sequence by introducing a memory unit and a gate mechanism.

Formally, the update of each LSTM component can be formalized as

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \mathbf{x}_t + \mathbf{U}_i \mathbf{h}_{t-1}), \tag{1}$$
$$\mathbf{f}_t = \sigma(\mathbf{W}_f \mathbf{x}_t + \mathbf{U}_f \mathbf{h}_{t-1}), \tag{2}$$
$$\mathbf{o}_t = \sigma(\mathbf{W}_o \mathbf{x}_t + \mathbf{U}_o \mathbf{h}_{t-1}), \tag{3}$$
$$\tilde{\mathbf{c}}_t = \tanh(\mathbf{W}_c \mathbf{x}_t + \mathbf{U}_c \mathbf{h}_{t-1}), \tag{4}$$
$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t, \tag{5}$$
$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{6}$$

where σ is the logistic sigmoid function. Operator $\odot$ is the element-wise multiplication operation. $\mathbf{i}_t$, $\mathbf{f}_t$, $\mathbf{o}_t$, and $\mathbf{c}_t$ are the input gate, forget gate, output gate, and memory cell activation vector at timestep t, respectively, all of which have the same size as the hidden vector $\mathbf{h}_t$. $\mathbf{W}_i$, $\mathbf{W}_f$, $\mathbf{W}_o$, and $\mathbf{U}_i$, $\mathbf{U}_f$, $\mathbf{U}_o$ are trainable parameters.

2.2 Hierarchical User Attention

Word Encoder. Given sentence s_i, we embed each word w_{ij} to vector $\mathbf{w}_{ij} \in \mathbb{R}^{e_w}$, where e_w is the dimension of word embeddings. Then we use a bidirectional LSTM to encode contextual information of word w_{ij} into its hidden representation $\mathbf{h}_{ij}$ as follows:

$$\overrightarrow{\mathbf{h}}_{ij} = \overrightarrow{\mathrm{LSTM}}(\mathbf{w}_{ij}), \tag{7}$$

$$\overleftarrow{\mathbf{h}}_{ij} = \overleftarrow{\mathrm{LSTM}}(\mathbf{w}_{ij}), \tag{8}$$

$$\mathbf{h}_{ij} = \overrightarrow{\mathbf{h}}_{ij} \oplus \overleftarrow{\mathbf{h}}_{ij}, \tag{9}$$

where $\overrightarrow{\mathrm{LSTM}}(\cdot)$ and $\overrightarrow{\mathrm{LSTM}}(\cdot)$ indicates the forward and backward process of LSTM, and $\oplus$ is the concatenating operator.

Word-Level Attention. It is obvious that not all words contribute equally to sentence meaning, especially for different users. To model WrdUP, we introduce a user attention mechanism to treat words differently in a sentence and get the sentence representation using Equation (10):

$$\mathbf{s}_i = \sum_j \alpha_{ij}\mathbf{h}_{ij}, \tag{10}$$

where α_{ij} measures the importance of the jth word for the current user. We embed user d_u as continuous and real-valued vector $\mathbf{d}_u \in \mathbb{R}^{e_u}$, where e_u is the dimensions of user embeddings. Then we compute α_{ij} as follows:

$$\mathbf{m}_{ij} = \tanh(\mathbf{W}_{wh}\mathbf{h}_{ij} + \mathbf{W}_{wu}\mathbf{d}_u + \mathbf{b}_w), \tag{11}$$

$$\alpha_{ij} = \frac{\exp(\mathbf{v}_w^T\mathbf{m}_{ij})}{\sum_j \exp(\mathbf{v}_w^T\mathbf{m}_{ij})}, \tag{12}$$

where $\mathbf{W}_{wh}$, $\mathbf{W}_{wu}$, $\mathbf{b}_w$, and $\mathbf{v}_w$ are parameters in the attention layer.

Sentence Encoder. After obtaining sentence vector $\mathbf{s}_i$, we also use a bidirectional LSTM to encode the sentences:

$$\overrightarrow{\mathbf{h}}_{i} = \overrightarrow{\mathrm{LSTM}}(\mathbf{s}_{i}), \tag{13}$$

$$\overleftarrow{\mathbf{h}}_{i} = \overleftarrow{\mathrm{LSTM}}(\mathbf{s}_{j}), \tag{14}$$

$$\mathbf{h}_{i} = \overrightarrow{\mathbf{h}}_{i} \oplus \overleftarrow{\mathbf{h}}_{i}, \tag{15}$$

where $\overrightarrow{\mathrm{LSTM}}(\cdot)$ and $\overrightarrow{\mathrm{LSTM}}(\cdot)$ indicates the forward and backward process of LSTM, and $\mathbf{h}_i$ summarizes the neighbor sentences around sentence i but still focuses on sentence i.

Sentence-Level Attention. Here, we get aspect representation from sentence encoder representation. After assigning each sentence with one or more aspects using *Aspect Segmentation* algorithm [38], we use a matrix $A \in \mathbb{R}^{n \times m}$ to record sentence-aspect information. If sentence s_i is assigned to aspect a_j, A_{ij} equals to 1, and otherwise A_{ij} equals to 0. Then we get aspect representation $\mathbf{x}_k$ for aspect a_k by merging sentences that are assigned to aspect a_k: as follows:

For arbitrary aspect a_k, there are three situations: (1) no sentence is assigned to a_k ($\sum_i A_{ik} = 0$), (2) only one sentence is assigned to a_k ($\sum_i A_{ik} = 1$), and (3) more than one sentence is assigned to a_k ($\sum_i A_{ik} > 1$). For situation (1), we use zero vector to represent $\mathbf{x}_k$. For situation (2), we use the representation of the sentence which is assigned to aspect a_k to represent $\mathbf{x}_k$. For situation (3), different sentences may have different effects on the aspect representation $\mathbf{x}_k$, so we use sentence attention to treat sentences differently. Specifically,

$$\mathbf{z}_{ik} = \tanh(\mathbf{W}_{sh}\mathbf{h}_i + \mathbf{b}_s), \tag{16}$$

$$\beta_{ik} = \begin{cases} 0 & \sum_i A_{ik} = 0 \\ 1 & \sum_i A_{ik} = 1 \\ \frac{A_{ik}\exp(\mathbf{v}_s^T \mathbf{z}_{ik})}{\sum_i A_{ik}\exp(\mathbf{v}_s^T \mathbf{z}_{ik})} & \sum_i A_{ik} > 1 \end{cases}, \tag{17}$$

$$\mathbf{x}_k = \sum_i \beta_{ik}\mathbf{h}_i, \tag{18}$$

$$\tag{19}$$

where β_{ik} measures the importance of sentence s_i for aspect a_k. $\mathbf{W}_{sh}$, $\mathbf{b}_s$, and $\mathbf{v}_s$ are parameters in the attention layer.

Aspect-Level Attention. After obtaining aspect representation, we obtain document representation. Different users care about different aspects. To get better document representation, we need to get customized aspect weights for each user. Therefore, we introduce user attention mechanism to treat aspects differently based on different users. Formally, the document representation $\mathbf{d}$ can be computed as follows:

$$\mathbf{t}_i = \tanh(\mathbf{W}_{ah}\mathbf{x}_i + \mathbf{W}_{au}\mathbf{d}_u + \mathbf{b}_a), \tag{20}$$

$$\gamma_i = \frac{\exp(\mathbf{v}_a^T \mathbf{t}_i)}{\sum_i \exp(\mathbf{v}_a^T \mathbf{t}_i)}, \tag{21}$$

$$\mathbf{d} = \sum_i \gamma_i \mathbf{x}_i, \tag{22}$$

where γ_i measures the importance of jth aspect for user d_u, which shows d_u's preference about aspect a_i. $\mathbf{d}_u$ is d_u's embedding vector. $\mathbf{W}_{ah}$, $\mathbf{W}_{au}$, $\mathbf{b}_a$, and $\mathbf{v}_a$ are parameters in the attention layer.

Review Representation. To consider **PolUP**, we get review representation $\mathbf{r}_d$ by concatenating user embedding $\mathbf{d}_u$ and document representation $\mathbf{d}$ using Equation (23):

$$\mathbf{r}_d = \mathbf{d}_u \oplus \mathbf{d}. \tag{23}$$

2.3 Document-Level Sentiment Classification

The review representation $\mathbf{r}_d$ is a high level representation of the combination of user information and document information and can be used as features for document classification. We use a softmax layer to project $\mathbf{r}_d$ into sentiment distribution $\mathbf{p}(d)$ over C classes:

$$\mathbf{p}(d) = \text{softmax}(\mathbf{W}_c\mathbf{r}_d + \mathbf{b}). \tag{24}$$

$p_c(d)$ is used to represent the predicted probability of sentiment class c for review d. Then we define the cross-entropy error between gold sentiment distribution and our model's sentiment distribution as our loss function:

$$L = -\sum_{d\in D}\sum_{c=1}^{C} \mathbb{1}\{g_d = c\} \cdot \log(p_c(d)), \tag{25}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function and g_d represents the ground truth label for review d.

3 EXPERIMENTS

In this section, we present the datasets used in our experiments and data preprocessing, aspect segmentation algorithm, training, and evaluation details, all the classification methods we compare in experiments and the empirical results on the task of document-level sentiment classification.

Table 4. Statistics of Different Datasets

Datasets	#docs	#users	#docs/user	#sens/doc	#words/sen	#words/doc
Tripadvisor	387,805	9,653	40.17	9.51	16.81	159.84
Yelp2014	231,163	4,818	47.97	11.41	17.26	196.91

The rating scale of Tripadvisor and Yelp2014 are 1–5. #users is the number of users, #docs/user indicates the average number of documents per user posts in the corpus. #words/sen (doc) indicates the average number of words in sentence (document). #sens/doc indicates the average number of sentences in document.

Table 5. Aspect Categories and Keywords for Different Datasets

Aspect	Keywords	Tripadvisor	Yelp2014
Facility	pool, parking, internet, wifi	✓	–
Value	value, price, quality, worth	✓	✓
Service	server, service, welcome, staff	✓	✓
Location	location, traffic, minute, walk	✓	✓
Food	delicious, breakfast, coffee, cheese	✓	✓
Room	room, bed, clean, dirty	✓	–
Environment	atmosphere, music, internet, quiet	–	✓
Others		✓	✓

"✓" denotes the dataset contains the specific aspect category, while "–" denotes not. "Others" is a default aspect category.

3.1 Dataset and Data Preprocessing

We evaluate HUAN on two datasets: Tripadvisor and Yelp2014. The first dataset is created by ourself, which belongs to the hotel domain. And the second one is built by Reference [35], which belongs to the restaurant domain. Various statistics of these datasets are summarized in Table 4.

We perform simple pre-processing on reviews in our datasets: (1) converting words into lower cases, (2) stemming words with Porter Stemmer [26], and (3) splitting sentences by Stanford CoreNLP [22].

3.2 Aspect Segmentation

We apply *Aspect Segmentation* algorithm [38] to mine aspect information from reviews. *Aspect Segmentation* is a boot-strapping algorithm that assigns sentences in our review corpus into different pre-defined aspects. The input for *Aspect Segmentation* is a collection of review sentences as well as a few keywords describing aspects, and the output is review sentences with aspect assignments. It assigns each sentence to the aspect that shares the maximum word overlapping with this sentence and expands the words with high dependencies into the corresponding aspect keyword list [38]. We manually define different aspects for our datasets as well as their keywords in Table 5.

If there is no word in a sentence matching aspect keywords, we will set a default category 'others" to the sentence. Other parameters in *Aspect Segmentation*, such as selection threshold and iteration times, are set as the same with Reference [38].

3.3 Training and Evaluation Details

We split the datasets into training, development, and testing sets in the proportion of 8:1:1 and use standard *Accuracy* to measure the overall sentiment classification performance and use *RMSE* to measure the divergences between predicted sentiment ratings and ground truth ratings. The

Accuracy and *RMSE* are defined as

$$Accuracy = \frac{T}{N}, \tag{26}$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N}(gd_i - pr_i)^2}{N}}, \tag{27}$$

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 randomly initialized or pre-trained. For Tripadvisor, we pre-train the 200-dimensional word-embeddings with SkipGram [23]. For Yelp2014, we use trained word embeddings by Reference [2]. We also initialize user embeddings randomly and set the user embedding dimension to 200. The dimensions of hidden states and cell states in LSTM cells are also set to 200. We tune the hyperparameters on the development sets and use Adadelta [46] to update parameters when training. We select the best model based on performance on the development set and then evaluate the model on the test set.

3.4 Comparison Methods

We compare HUAN with the following baseline methods for document-level sentiment classification:

(1) *Majority* is a heuristic baseline method that assigns the majority sentiment category in training set to each review in the test dataset.
(2) *Trigram* trains a SVM classifier with unigrams, bigrams, and trigrams as features.
(3) *AvgWordvec* averages word embeddings in a document to obtain document representation that is fed into a SVM classifier as features.
(4) *HAN* [45] models review in a hierarchical structure (from word-level, sentence-level to document-level) and utilizes an attention mechanism to capture important words and sentences, which is only based on text information and achieves state-of-the-art result in document-level sentiment classification.
(5) *BiLSTM* uses bidirectional LSTM to model reviews from word-level to document-level representation without hierarchical structure.
(6) *BiLSTM+UA* is a variant of BiLSTM that adds user attention to model reviews.
(7) *BiLSTM+UAl* is another variant of BiLSTM, which not only adds user attention to model reviews but also concatenates user embedding and document vector to predict sentiment.
(8) *NSC+UPA* [2] is the state-of-the-art system considering user and product information to improve document-level sentiment classification.[3]
(9) *HUAN-usr* is a variant of HUAN that abandons user information from HUAN and predicts sentiment ratings only based on text.
(10) *HUAN-asp* is also a variant of HUAN, which abandons aspect information from HUAN and obtains document representation directly from sentence representation.
(11) *HUAN-usr-asp* is another variant of HUAN, which abandons user information as well as aspect information.

[3] As the performance reported in Reference [2] is better than other related work [6, 34, 36] on public dataset Yelp2014, we only compare HUAN with Reference [2].

Table 6. Sentiment Classification on Tripadvisor and Yelp2014 Datasets

Models	Tripadvisor		Yelp2014	
	Acc↑	RMSE↓	Acc↑	RMSE↓
User-agnostic models				
Majority	0.413	0.910	0.392	1.097
Trigram	0.578	0.748	0.577	0.804
AvgWordvec	0.610	0.708	0.530	0.893
BiLSTM	0.660	0.668	0.628	0.712
HAN	0.666	0.609	0.638	0.690
HUAN-usr-asp	0.681	0.593	0.642	0.680
HUAN-usr	**0.684**	**0.590**	**0.644**	**0.675**
User-aware models				
BiLSTM+UA	0.693	0.587	0.653	0.673
BiLSTM+UAl	0.703	0.581	0.664	0.663
NSC+UPA	0.710	0.562	0.667	0.654
HUAN-asp	0.712	0.558	0.670	0.652
HUAN	**0.715***	**0.556**	**0.672***	**0.651**

Our full model is HUAN. The best performance in each group is in **bold**. "*" indicates that the model significantly outperforms NSC+UPA. Statistical significance testing has been performed using paired t-test with $p < 0.05$.

3.5 Results

Experimental results are given in Table 6. The results are separated into two groups: user-agnostic models and user-aware models.

For the first group, we can see that *Majority* performs very poorly because it does not capture any text information. SVM classifier with unigrams, bigrams, and trigrams Trigram) are powerful for document-level sentiment classification, which is also better than SVM classifier with average word embedding (AvgWordvec). When applying bidirectional LSTM to model reviews from word-level to document-level representation directly, BiLSTM obtains better results compared with AvgWordvec. However, BiLSTM performs worse than HAN, which indicates that the hierarchical structure is useful for the document-level sentiment classification task. Our text-only-based model (HUAN-usr) performs better than HAN, Trigram, and AvgWordvec. When we remove aspect information from HUAN-usr, the performance is descending (HUAN-usr versus HUAN-usr-asp), which shows that modeling review from aspects is better than modeling review directly from sentences for document-level sentiment classification.

For the second group, we can see that the user information is helpful to neural-network-based models for sentiment classification. Adding such information into BiLSTM, BiLSTM+UA, and BiLSTM+UAl achieves 3.3% (4.3%) and 2.5% (3.6%) improvements on Tripadvisor and Yelp2014, respectively. With the consideration of such information into HUAN-usr, HUAN also achieves 3.1% and 2.8% improvements in Tripadvisor and Yelp2014. At last, our model obtains better results than the state-of-the-art system NSC+UPA significantly. It's worth mentioning our model only considers user information and NSC+UPA considers user information as well as product information, but even so, our model outperforms NSC+UPA, which shows that our model can better incorporate user information for document-level sentiment classification than NSC+UPA. When we remove aspect information from HUAN, the performance is also descending (HUAN versus HUAN-asp),

Table 7. Effects of User Preference in Different Levels on Document-Level Sentiment Classification

No.	Different levels			Tripadvisor		Yelp2014	
	WrdUP	**AspUP**	**PolUP**	Acc↑	RMSE↓	Acc↑	RMSE↓
1	–	–	–	0.684	0.590	0.644	0.675
2	✓	–	–	0.703	0.566	0.654	0.668
3	–	✓	–	0.698	0.580	0.650	0.670
4	–	–	✓	0.705	0.565	0.660	0.662
5	✓	✓	–	0.705	0.563	0.660	0.660
6	✓	–	✓	0.712	0.556	0.670	0.652
7	–	✓	✓	0.710	0.561	0.668	0.654
8	✓	✓	✓	0.715	0.556	0.672	0.651

"✓" denotes a model considers the specific preference, while "–" denotes not.

which once again indicates that aspect information is also useful for document-level sentiment classification.

4 DISCUSSIONS

In this section, we first give some discussions about the effects of different levels' user preference on document-level sentiment classification and then visualize multi-level user preference.

4.1 Effects of User Preference in Different Levels

Table 7 shows that the effect of user preference in different levels on sentiment classification. Here we identify which level user preference is the most important for document-level sentiment classification. From the table, we can observe that

(1) When our model is user-agnostic (line 1), HUAN gets the worst performance. Even so, it also performs better than other text-only-based models (Table 6).
(2) When there is only one kind of user preference considered in our model (lines 2–4), HUAN can obtain at least 1.4% and 0.6% improvements in accuracy compared with the user-agnostic model. Compared with WrdUP and AspUP, PolUP has the greatest impact on boosting the performance. The main reason is that PolUP directly model the relationship between user and sentiment rating, while others only model such information through words and aspects.
(3) When considering two kinds of user preference (lines 5–7), HUAN can obtain better performance and achieve at least 2.1% and 1.6% improvements in accuracy compared with the user-agnostic model. The results reconfirm that PolUP is the most important factor to perform the task.
(4) After WrdUP, AspUP, and PolUP being considered jointly, our model gets the best performance.

4.2 Visualization of Multi-Level User Preference

4.2.1 Visualization of Word-Level User Preference. To show the ability that HUAN can capture WrdUP for different users, we take two sentences with "good" posted by different users in Tripadvisor for example. The content of these two sentences are "The place is really *good* with nothing in the area" and "The food is very *good* and decently priced for here." These two sentences are in different circumstances, the former is in a 2-star review while the latter is in a 5-star review. We visualize the attention weights in word-level for these two users (*User1* and *User2*) and the local semantic attention (Local Attention) in Figure 2. Here, the local semantic attention indicates the

Fig. 2. Visualization of attention weights over words for different users.

implementation in Reference [45], which calculates attention over words without considering user information.

According to our statistics, *User1* often gives negative reviews, no matter if "good" appears in *User1*'s reviews or not. The word "good" is used 19 times in *User1*'s reviews and the times it appears in high-rating or low-rating reviews are almost equal. However, *User2* often gives high-score reviews and always uses "good" to express his attitude to products. Most instances of "good" appear in high-score reviews for *User2*. Therefore, "good" means different things for *User1* and *User2*. HUAN can treat it differently and capture "good" as an unimportant (or important) factor for *User1* (or *User2*) to determine review score, which is reflected in different attention weights of "good." As Local Attention is user-agnostic, it cannot capture the difference and treat "good" important for all users.

4.2.2 Visualization of Aspect-Level User Preference. To show the ability that HUAN can capture AspUP for different users, we take three reviews posted by three different users (*User3*, *User4*, and *User5*) in Tripadvisor for example. Figure 3 visualizes attention weights over aspect-level representation for these users. *User3* posts a negative review and comments three aspects, room, location," and other. From the review contents, we can find the main reason why he gives 1 star is that he is dissatisfied with the room and the location. Therefore we can find he prefers room and location, and HUAN can also capture the information. Although *User4* dislikes the size of hsi room, he likes the service and also gives 5 stars, which shows that he prefers service more. *User5* gives service and food very positive comments, feels disappointed with location, and finally he only gives 2 stars to the review, which shows that he cares about location most. At last we can find attention weights over aspect-level representation for different users can truly show user preferences on different aspects.

However, these weights only show user preference about aspects in a specific review. To mine user preference about aspects for all reviews, we should add them up. In our model, γ_i represents attention weight at aspect-level in review d and shows user d_u's preference about aspect a_i. We can concatenate all γ_i to obtain $\boldsymbol{\gamma}_{d_u}$ by Equation (28), then use $\boldsymbol{\gamma}_{d_u}$ to represent d_u's preference about all aspects in d. Finally, given an arbitrary user u in corpus D, we use $\boldsymbol{\Gamma}_u$ to represent his preference about all aspects and calculate it through Equation (29):

$$\boldsymbol{\gamma}_{d_u} = \gamma_1 \oplus \cdots \oplus \gamma_m, \tag{28}$$

$$\boldsymbol{\Gamma}_u = \sum_{d \in D} \mathbb{1}\{d_u = u\} \times \boldsymbol{\gamma}_{d_u}, \tag{29}$$

where $\mathbb{1}\{\cdot\}$ is the indicator function.

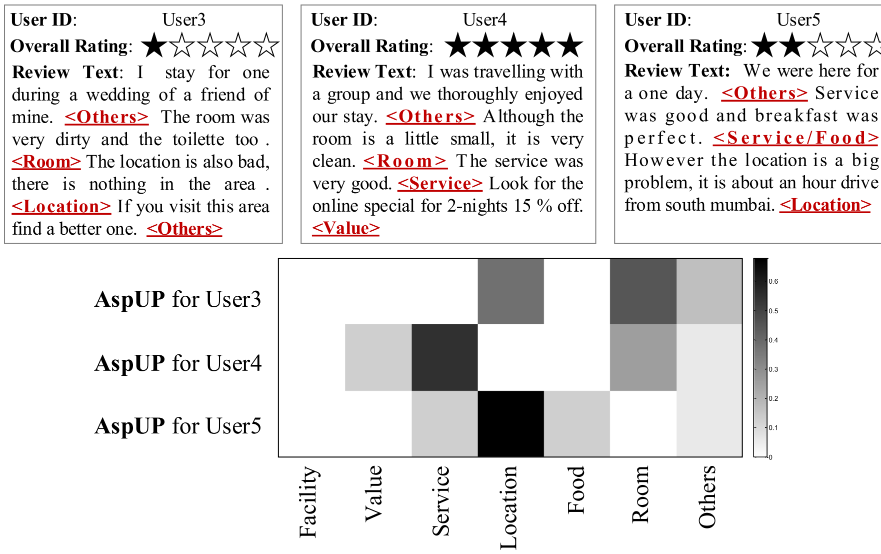


Fig. 3. Visualization of attention weights over aspects for different users. The upper part shows reviews posted by three different users. **Words** with red color, bold, and underlined are aspects for the sentence before the **Words**. The below part shows results of attention weights over aspects for these users.

Table 8. Aspects Win the Highest Priority for Customers in Different Datasets

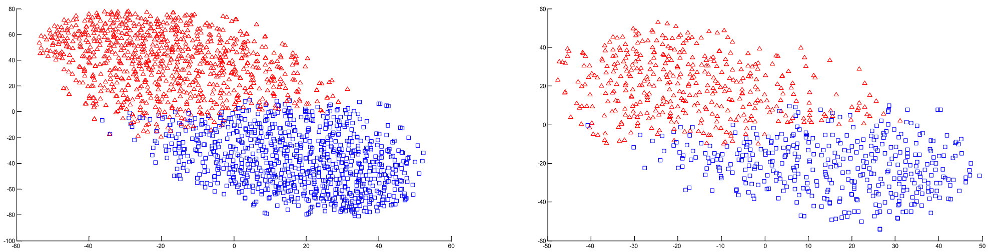
Tripadvisor		Yelp2014	
Aspects	Percentage	Aspects	Percentage
Room	56.80%	Food	55.92%
Service	27.54%	Service	18.70%
Food	12.65%	Value	12.27%
Facility	1.60%	Location	11.77%

Percentage shows how many users agree with the top aspects.

After computing $\mathbf{\Gamma}_u$, we can get aspect-level preference for all users in our datasets. Then we show the top aspect[4] that a user cares about in different datasets in Table 8 and identify the most important aspects for choosing hotels and restaurants. And we find that room and service are the most important factors for customers to compare different hotels. When choosing restaurants, 55.92% users primarily care about food and 18.70% users care about service as the first candidate.

4.2.3 Visualization of Polarity-Level User Preference. As different users have different polarity-level preferences and HUAN imports user embedding to consider users, we identify whether such personalized information is encoded in user embedding. To perform the task, we first rank all users according to their average score in training set. Then the top 10% of users are labeled as high-score users and the bottom 10% users are labeled as low-score users. Finally, we visualize user embedding of these users in Figure 4. We find high-schore users and low score users are separated, apparently.

[4]Here we focus on aspect categories with specific meanings such as service and food, and ignore "other".



(a) User embedding visualization in Tripadvisor (b) User embedding visualization in Yelp2014

Fig. 4. t-SNE Visualization for user embeddings. Blue squares and red triangles represent high- score users and low-score users, respectively.

The visualization shows that user embedding learned by HUAN can encode personalized traits in scoring reviews.

5 RELATED WORK

Document-level sentiment classification is a typical task in sentiment analysis [21, 24], which infers the sentiment polarity of a whole document. Pang et al. [25] regard this problem as a special case of text classification, and use a machine learning method in a supervised learning framework. Since the performance of supervised learning methods is heavily dependent on the representation of data, most studies follow [25] and focus on designing effective features, such as bag-of-opinion features [27], sentiment lexicon features [14] and word relation features [43].

User information is also used for sentiment classification. Gao et al. [7] design user-specific features to capture user leniency. Many studies [11, 12, 33] utilize user interactions (such as following, retweeting, etc.) to improve sentiment classification performance. Their main idea is that sentiments of two messages posted by friends are more likely to be similar than those of two randomly selected messages. Incorporating this information into a graph-based model [12] or a supervised method [11] obtains good performance. Other studies [1, 31, 35] also incorporate user information to perform personalized sentiment classification. Unlike most previous studies that design hand-crafted features to consider user information, we use neural network approach and learn discriminative features from data.

Neural-network-based methods are prevalent for sentiment classification due to their ability to learn discriminative features from data. Socher conducts a series of recursive neural network models to perform sentiment classification, including recursive autoencoder [29], matrix-vector recursive neural network [28], and recursive neural tensor network [30]. Other studies [32, 34] adopt recurrent neural network in sentiment classification due to its capacity to capture sequential information. Li et al. [15] compare the effectiveness of recursive neural network and recurrent neural network on five NLP tasks including sentiment classification. Besides, Kim [13] also applies convolution neural network to learn sentence representations and obtains outstanding performance in sentiment classification.

Most existing neural-based sentiment classification models ignore the effect of user information on determining sentiment polarities. To address this issue, Tang et al. [36] represent each user as a matrix and introduce a user-word composition vector model (UWCVM) to effectively consider WrdUP. Finally, they integrate UWCVM into a convolution neural network for review rating prediction. Tang et al. [35] extend Reference [36] by concatenating user embedding and document representation to consider PolUP. However, it is hard to train with limited reviews, especially for user matrix [2]. To make training more efficient, Chen et al. [2] and Dou [6] embed user as vectors and merge them into a hierarchical LSTM model or deep memory network to perform sentiment classification. Nevertheless, they only focus on WrdUP and ignore AspUP and PolUP. In conclusion, these related studies either consider user preference in a manner that is difficult to train or ignore important user preferences. Our model, HUAN, cannot only fully take all these preferences into consideration, but also is easy to train. Furthermore, HUAN can also mine important aspects for different users.

6 CONCLUSION AND FUTURE WORK

In this article, we present HUAN, a Hierarchical User Attention Network model, to consider user information for classifying reviews. To thoroughly analyze the effect of user information on determining sentiment polarity, we present three kinds of user preference, which are word-level user preference, aspect-level user preference, and polarity-level user preference. To fully consider these preferences, HUAN introduces user information as attentions over word-level representation and aspect-level representation, and generates review representation by combining user and document information. When modeling review contents, HUAN imports an aspect layer and encodes different kinds of information (word, sentence, aspect, and document) in a hierarchical structure.

The proposed model is evaluated on two real-world datasets (Tripadvisor and Yelp14). Experiments show that (1) aspect layer is helpful for document representation, (2) considering user information can boost sentiment classification performance by a large margin, and (3) HUAN outperforms state-of-the-art methods significantly. Furthermore, HUAN can also mine important aspects for different users.

In the future, we will first expand user information and explore the effect of user attributes (such as age and sex) on sentiment classification. Second, as HUAN can mine important aspects for different users, we will apply HUAN to other personalized tasks, such as personalized recommendation.

REFERENCES

[1] Mohammad Al Boni, Keira Qi Zhou, Hongning Wang, and Matthew S. Gerber. 2015. Model adaptation for personalized opinion analysis. In *Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 769–774.

[2] Huimin Chen, Maosong Sun, Cunchao Tu, Yankai Lin, and Zhiyuan Liu. 2016. Neural sentiment classification with user and product attention. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1650–1659.

[3] Dan C. Ciresan, Ueli Meier, and Jürgen Schmidhuber. 2012. Multi-column deep neural networks for image classification. In *Proceedings of the 2012 Conference on Computer Vision and Pattern Recognition*. IEEE Computer Society, 3642–3649.

[4] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel P. Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12 (2011), 2493–2537.

[5] George E. Dahl, Dong Yu, Li Deng, and Alex Acero. 2011. Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *IEEE Transactions on Audio Speech and Language Processing* 20, 1 (2011), 30–42.

[6] Zi-Yi Dou. 2017. Capturing user and product information for document level sentiment analysis with deep memory network. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 532–537.

[7] Wenliang Gao, Naoki Yoshinaga, Nobuhiro Kaji, and Masaru Kitsuregawa. 2013. Modeling user leniency and product popularity for sentiment classification. In *Proceedings of the 2016 International Joint Conference on Natural Language Processing*. Asian Federation of Natural Language Processing, 1107–1111.

[8] Shima Gerani, Yashar Mehdad, Giuseppe Carenini, Raymond T. Ng, and Bita Nejat. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1602–1613.

[9] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *Proceedings of the 2017 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 388–397.

[10] Sepp Hochreiter and JÃijrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.

[11] Xia Hu, Lei Tang, Jiliang Tang, and Huan Liu. 2013. Exploiting social relations for sentiment analysis in microblogging. In *Proceedings of the 2013 International Conference on Web Search and Data Mining*. ACM, 537–546.

[12] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter sentiment classification. In *Proceedings of the 2011 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 151–160.

[13] Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1746–1751.

[14] Svetlana Kiritchenko, Xiaodan Zhu, and Saif M Mohammad. 2014. Sentiment analysis of short informal texts. *Journal of Artificial Intelligence Research* 50 (2014), 723–762.

[15] Jiwei Li, Thang Luong, Dan Jurafsky, and Eduard Hovy. 2015. When are tree structures necessary for deep learning of representations. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2304–2314.

[16] Junjie Li, Haitong Yang, and Chengqing Zong. 2016. Sentiment classification of social media text considering user attributes. In *Proceedings of the 2016 Conference on Natural Language Processing and Chinese Computing*. Springer, 583–594.

[17] Shoushan Li, Chu-Ren Huang, and Chengqing Zong. 2011. Multi-domain sentiment classification with classifier combination. *Journal of Computer Science and Technology* 26, 1 (2011), 25–33.

[18] Shoushan Li, Shengfeng Ju, Guodong Zhou, and Xiaojun Li. 2012. Active learning for imbalanced sentiment classification. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics, 139–148.

[19] Shoushan Li, Sophia Yat Mei Lee, Ying Chen, Chu-Ren Huang, and Guodong Zhou. 2010. Sentiment classification and polarity shifting. In *Proceedings of the 2010 International Conference on Computational Linguistics*. COLING 2010 Organizing Committee, 635–643.

[20] Shoushan Li, Yunxia Xue, Zhongqing Wang, and Guodong Zhou. 2013. Active learning for cross-domain sentiment classification. In *Proceedings of the 2013 International Joint Conference on Artificial Intelligence*. AAAI Press, 2127–2133.

[21] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies* 5, 1 (2012), 1–167.

[22] 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 the 2014 Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, 55–60.

[23] Tomas Mikolov, Greg Corrado, Kai Chen, Jeffrey Dean, Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. *Efficient Estimation of Word Representations in Vector Space*. arXiv preprint arXiv:1301.3781 (2013).

[24] Bo Pang and Lillian Lee. 2008. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval* 2, 1–2 (2008), 1–135.

[25] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 79–86.

[26] Martin F. Porter. 1980. An algorithm for suffix stripping. *Program* 14, 3 (1980), 130–137.

[27] 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 2010 International Conference on Computational Linguistics*. COLING 2010 Organizing Committee, 913–921.

[28] 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*. Association for Computational Linguistics, 1201–1211.

[29] 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 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 151–161.

[30] 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 2013 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1631–1642.

[31] Kaisong Song, Shi Feng, Wei Gao, Daling Wang, Ge Yu, and Kam Fai Wong. 2015. Personalized sentiment classification based on latent individuality of microblog users. In *Proceedings of the 2015 International Joint Conference on Artificial Intelligence*. AAAI Press, 2277–2283.

[32] Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1556–1566.

[33] 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 2011 International Conference on Knowledge Discovery and Data Mining*. ACM, 1397–1405.

[34] Duyu Tang, Bing Qin, and Ting Liu. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 1422–1432.

[35] Duyu Tang, Bing Qin, and Ting Liu. 2015. Learning semantic representations of users and products for document level sentiment classification. In *Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1014–1023.

[36] Duyu Tang, Bing Qin, Yuekui Yang, and Yuekui Yang. 2015. User modeling with neural network for review rating prediction. In *Proceedings of the 2015 International Joint Conference on Artificial Intelligence*. AAAI Press, 1340–1346.

[37] 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 the 2014 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 1555–1565.

[38] Hongning Wang, Yue Lu, and Chengxiang Zhai. 2010. Latent aspect rating analysis on review text data: A rating regression approach. In *Proceedings of the 2010 International Conference on Knowledge Discovery and Data Mining*. ACM, 783–792.

[39] Lu Wang and Wang Ling. 2016. Neural network-based abstract generation for opinions and arguments. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 47–57.

[40] Linlin Wang, Kang Liu, Zhu Cao, Jun Zhao, and Gerard de Melo. 2015. Sentiment-aspect extraction based on restricted Boltzmann machines. In *Proceedings of the 2015 Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 616–625.

[41] Rui Xia, Feng Xu, Jianfei Yu, Yong Qi, and Erik Cambria. 2016. Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis. *Information Processing and Management* 52, 1 (2016), 36–45.

[42] Rui Xia, Feng Xu, Chengqing Zong, Qianmu Li, Yong Qi, and Tao Li. 2015. Dual sentiment analysis: Considering two sides of one review. *IEEE Transactions on Knowledge and Data Engineering* 27, 8 (2015), 2120–2133.

[43] Rui Xia and Chengqing Zong. 2010. Exploring the use of word relation features for sentiment classification. In *Proceedings of the 2010 International Conference on Computational Linguistics*. Association for Computational Linguistics, 1336–1344.

[44] Rui Xia, Chengqing Zong, and Shoushan Li. 2011. Ensemble of feature sets and classification algorithms for sentiment classification. *Information Sciences* 181, 6 (2011), 1138–1152.

[45] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alexander J. Smola, and Eduard H. Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, 1480–1489.

[46] Matthew D. Zeiler. 2012. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012).

Received November 2017; revised March 2018; accepted June 2018