# AELA-DLSTMs: Attention-Enabled and Location-Aware Double LSTMs for Aspect-level Sentiment Classification

Kai Shuang[a], Xintao Ren[a,*], Qianqian Yang[a], Jonathan Loo[b], Rui Li[a]

[a]*State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China*
[b]*School of Computing and Engineering, University of West London, London, UK*

**Abstract**

Aspect-level sentiment classification, as a fine-grained task in sentiment classification, aiming to extract sentiment polarity from opinions towards a specific aspect word, has been made tremendous improvements in recent years. There are three key factors for aspect-level sentiment classification: **contextual semantic information towards aspect words, correlations between aspect words and their context words, and location information of context words with regard to aspect words**. In this paper, two models named AE-DLSTMs (Attention-Enabled Double LSTMs) and AELA-DLSTMs (Attention-Enabled and Location-Aware Double LSTMs) are proposed for aspect-level sentiment classification. AE-DLSTMs take full advantage of the DLSTMs (Double LSTMs) which can capture **the contextual semantic information** in both forward and backward directions **towards aspect words**. Meanwhile, a novel attention weights generating method that combines aspect words with their contextual semantic information is designed so that those weights can make better use of **the correlations between aspect words and their context words**. Besides, we observe that context words with different distances or different directions towards aspect words have different contributions in sentiment polarity. Based on AE-DLSTMs, **the location information of context words** by assigning different weights is incorporated in AELA-DLSTMs to improve the accuracy. Experiments are conducted on two English datasets and one

*Corresponding author
*Email addresses:* shuangk@bupt.edu.cn (Kai Shuang), rxt2012kc@bupt.edu.cn (Xintao Ren), echo_yang@bupt.edu.cn (Qianqian Yang), jonathan.loo@uwl.ac.uk (Jonathan Loo), lirui@bupt.edu.cn (Rui Li)

Chinese dataset. The experimental results have confirmed that our models can make remarkable improvements and outperform all the baseline models in all datasets, improving the accuracy of 1.67 percent to 4.77 percent in different datasets compared with baseline models[1].

*Keywords:* Neural Network; Long Short-Term Memory; Attention Mechanism; Aspect-Level Sentiment Classification

## 1. Introduction

Sentiment classification, as a core part in Natural Language Processing (NLP) [1, 2], has attracted great attention in recent years. Early work in sentiment classification mainly aimed to detect the overall polarity (e.g., positive, negative or neutral) of given texts [3, 4]. As a more fine-grained approach, aspect-level sentiment classification is also a fundamental task aiming to extract aspect polarity from opinions towards a specific aspect word [2, 5, 6]. For example, "The ambience was nice, but the service was awful.", for aspect word *ambience*, the sentiment polarity is *positive* while for *service* it is *negative*. And in the sentence "质量很好, 但是做工很差", the sentiment polarity for "质量" is *positive* while for "做工" is *negative*. We observe that a sentence may contain multiple aspect words, in which case the sentiment polarity corresponding to each aspect word may be different.

Neural network models have been demonstrated to be capable of achieving state-of-the-art performance in many NLP tasks. Long Short-term Memory Network (LSTM) is widely used in tasks such as sentiment classification [7, 8], question answering [9], automatic summarization [10] and machine translation [11]. LSTMs [12], which can tackle the problem of gradient exploding or vanishing, are superior to standard RNNs and they are also applied to aspect-level sentiment classification such as Target-Dependent LSTM (TD-LSTM), Target-Connection LSTM (TC-LSTM) [13] and Attention-based LSTM with Aspect Embedding (ATAE-LSTM) [14]. DLSTMs (Double LSTMs) that we applied in our models take the both past and future information towards aspect words

[1]Our code is open-source and available at https://github.com/rxt2012kc/AELA-DLSTMs.

into consideration [15, 16] while LSTMs only consider the future information, thus DLSTMs can gain better results than LSTMs.

Attention mechanism has a long history in the field of neural networks, especially in the field of image recognition [17, 18, 19]. Recently, it has been commonly used in the NLP domain, such as speech recognition task [20, 21] and neural machine translation [22, 23]. There are also applications of attention mechanism in aspect-level sentiment classification. Tang et al. [24] proposed a deep memory network with attention. Wang et al. [14] designed Attention-based LSTM (AT-LSTM) and ATAE-LSTM. Yang et al. [25] improved the methods for assigning attention scores. Despite the effectiveness of above approaches, there still remains a challenge that how to model the semantic correlations of an aspect word with its context words more effectively in a sentence and to assign the attention weights for hidden state more precisely at each time step.

Encoding a sequence of word vectors into a sentence vector, emphasizing the aspect words information and extracting the correlations between aspect words and their context words are of great significance for aspect-level sentiment classification task. The aspect-level sentiment classification task has achieved excellent developments these years [14, 24, 26]. The key of it can be summarized as the following three factors: contextual semantic information towards aspect words, correlations between aspect words and their context words, and location information of context words with regard to aspect words. Better results can be obtained by taking into consideration all the three key factors. However, there has not been a certain model that fully considered the above three factors in aspect-level sentiment classification area.

In this paper, we propose two models named AE-DLSTMs (Attention-Enabled Double LSTMs) and AELA-DLSTMs (Attention-Enabled and Location-Aware Double LSTMs) considering all the three key factors for aspect-level sentiment classification. Given that DLSTMs are able to obtain better remembering and memory accesses and capture the contextual semantic information from forward and backward orders of the contexts, our models based on DLSTMs structure can gain more contextual semantic information towards aspect words. AE-DLSTMs are capable of capturing the correlations between aspect words and their context words more accurately by considering different attention factors successively to generate attention weights for hidden states.

Then we notice that context words in different locations relating to one specific aspect word have different contributions to express the sentiment polarity and the key words always locate in one side of the aspect word. From this perspective, AELA-DLSTMs are proposed with generating different weights for the context words by capturing the location information. The main contributions of our work can be summarized as follows:

- AE-DLSTMs are designed which is capable of emphasizing the aspect words information more effectively and capturing the correlations between aspect words and their context words more precisely.
- AELA-DLSTMs are proposed which can take full advantage of the location information of contexts words related to one aspect word to obtain more accurate results.
- Experimental results conducted on both English and Chinese datasets confirm that our models can obtain better results and outperform all the baseline models for aspect-level sentiment classification.

## 2. Related Work

Sentiment classification, also known as opinion mining, is a fundamental area in NLP [3, 4, 27, 28, 29, 30, 31, 32]. Deep learning based on neural network models has achieved a great success in sentiment classification [13, 33, 34, 35, 36, 37]. CNN and RNN are two mainstream models in sentiment classification, where word embedding is always taken as the model input to implement sentence classification [38, 39, 40, 41]. RNN models are capable of dealing with input sentences of variable lengths, thus obtaining long-term dependencies in a sentence [26, 42, 43, 44, 45]. Due to the gradient exploring and gradient vanishing of RNN, standard LSTM is often applied to take the place of traditional RNN model for better remembering and memory accesses [8, 12, 13, 46, 47, 48].We use double LSTMs structure called DLSTMs that process the input sentence in both forward and backward directions towards aspect words [13, 44, 49, 50, 51, 52].

Aspect-level sentiment classification, as a fine-grained sentiment classification task, has also attracted much attention over these years. A huge number of work in this area have been conducted [5, 53, 54, 55, 56, 57, 58, 59]. However, previous work always brings about a lot of labor work and extra lexicon to exact features, which could be an enormous project. With the development of neural network in NLP areas [60, 61, 62], neural network models for aspect-level sentiment classification is emerging [14, 24, 25, 30, 63, 64, 65]. Tang et al. [26] developed two target dependent long short-term memory (LSTM) models named Target-Dependent LSTM (TD-LSTM) and Target-Connection LSTM (TC-LSTM). Tang et al. [24] introduced a deep memory network that calculated with multiple computational layers named MemNet. Wang et al. [14] proposed an Attention-based Long Short-Term Memory Network named ATAE-LSTM. Yang et al. [25] presented two attention methods to improve the target-dependent sentiment classification. Xue et al. [66] proposed a model called Gated Convolutional network with Aspect Embedding (GCAE) based on convolutional neural networks and gating mechanisms. However, ATAE-LSTM based on Single LSTM only considered the forward semantic information of context words while DLSTMs can consider both forward and backward semantic information of context words towards aspect words. MemNet failed to make good use of the correlations between aspect words and their context words. TD-LSTM, TC-LSTM, ATAE-LSTM and GCAE ignored the location information of context words with regard to aspect words.

Attention mechanism has been a significant part in NLP tasks in recent years [18, 22, 23, 67, 68, 69, 70]. Attention mechanism structure for aspect-level sentiment classification also achieves excellent results. AE-LSTM integrated the Aspect Embedding information in the input of LSTM, AT-LSTM integrated the Attention Mechanism in the hidden state of LSTM to capture the key part of sentence in response to a given aspect and ATAE-LSTM was formed by the combination of AE-LSTM and AT-LSTM [14]. Yang et al. [25] designed models to assign attention scores to different word locations according to their relevance to the task. Ma et al. [71] proposed a model named interactive attention networks (IAN), which used two attention networks to model the target and content interactively. Our models refer to the relatedness towards aspect words and assign more reasonable weights by attention mechanism.

Our work aims to deal with the challenge of how to model the semantic relatedness of aspect words with their context words more accurately in a sentence and our models are designed considering all the three key factors for aspect-level sentiment classification to further improve the results.

## 3. The Proposed Models: AE-DLSTMs and AELA-DLSTMs

In this section, we introduce AE-DLSTMs and AELA-DLSTMs, aiming to improve the performances for aspect-level sentiment classification. The two models use attention weights generating method to compute more accurate weights for sentence representation towards the aspect words, thus can capture the contextual semantic correlations more precisely with aspect words. AELA-DLSTMs assign location weights for input words to enhance the key words towards aspect words, which make properer use of the location information to improve the results.

### *3.1. Task Definition and Notion*

$S = \{w_1, w_2, \ldots, w_a, \ldots, w_n\}$ is defined as a sentence consisting of $n$ words and the aspect word $w_a$ is included in $S$. The task is to determine the sentiment polarity towards a specific aspect word in a sentence. For example, in the sentence "lots of extra space but the keyboard is ridiculously small.", the sentiment polarity towards aspect word *space* is *positive* while the sentiment polarity towards aspect word *keyboard* is *negative*. And in the sentence "面料太粗糙, 快递很快", the sentiment polarity towards "面料" is *negative* while the sentiment polarity towards "快递" is *positive*.

Each input word is mapped into its embedding vector [35, 38, 39, 41]. All the word vectors are stacked in a word embedding matrix $V_W \in \mathbb{R}^{d \times |V|}$, where $d$ is the dimension of word vector and $|V|$ is the vocabulary size. The word embedding of input word $w_i$ is notated as $e_i \in \mathbb{R}^{d \times 1}$, which is a column of the embedding matrix $V_W$. Thus, the sentence is expressed as $E_s = \{e_1, e_2, \ldots, e_{a-1}, e_a, e_{a+1}, \ldots, e_n\}$, where $E_s \in \mathbb{R}^{d \times n}$, $n$ is the sentence length and $a_k$ is the index of aspect word. In cases where aspect is a multi-word phrase like "battery life", aspect representation is taken as an average of their constituting word vectors [28, 72]. The hidden states of BiLSTM

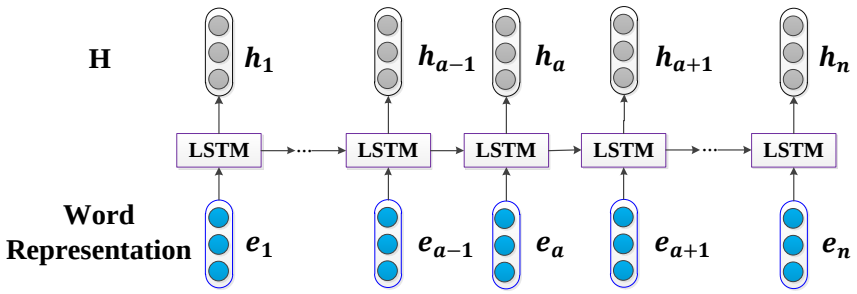


Figure 1: The architecture of LSTM. Word representation $\{e_1, \ldots, e_{a-1}, e_a, e_{a+1}, \ldots, e_n\}$ are the word embeddings of input words in a sentence whose length is $n$ and the index of aspect word is $a$. $H = \{h_1, \ldots, h_{a-1}, h_a, h_{a+1}, \ldots, h_n\}$ are the hidden states of input words.

structure for inputs $E_s$ is defined as $H = \{h_1, h_2, \ldots, h_{a-1}, h_a, h_{a+1}, \ldots, h_n\}$, where $h_a$ is the hidden state of aspect word $w_a$.

### *3.2. Long Short-Term Memory (LSTM)*

In order to overcome the gradient vanishing or exploding problems of Standard RNN, Long Short-term Memory network (LSTM) was developed [12]. LSTM has three gates (input $i$, forget $f$ and output $o$) and a cell memory state $c$. Generally, the hidden state $h_t$ at the time step $t$ is updated as follows:

$$i_t = \sigma\left(W_i\left[h_{t-1}; e_t\right] + b_i\right) \tag{1}$$

$$f_t = \sigma\left(W_f\left[h_{t-1}; e_t\right] + b_f\right) \tag{2}$$

$$o_t = \sigma\left(W_o\left[h_{t-1}; e_t\right] + b_o\right) \tag{3}$$

$$g_t = tanh\left(W_r\left[h_{t-1}; e_t\right] + b_r\right) \tag{4}$$

$$c_t = i_t \odot g_t + f_t \odot c_{t-1} \tag{5}$$

$$h_t = o_t \odot \tanh\left(c_t\right) \tag{6}$$

where $\sigma$ is the sigmoid function, $e_t$ is the input word embedding, $W_i$, $W_f$, $W_o$, $W_r \in \mathbb{R}^{d \times 2\mathrm{d}}$, $b_i$, $b_f$, $b_o$, $b_r \in \mathbb{R}^d$. The architecture of LSTM is shown as Figure 1.

Single direction LSTM suffers a weakness of not utilizing the contextual semantic information from the future tokens while DLSTMs applied in our models utilize both

the backward and forward contextual semantic information by processing the sequence in two directions, where $LSTM_L$ processes the left part input sequence of aspect word in the forward direction while $LSTM_R$ processes the right part input sequence of aspect word in the reverse direction. DLSTMs generates two independent sequences of LSTM output vectors and hidden states, and $LSTM_L$ and $LSTM_R$ share a set of parameters when training in our models.

### *3.3. Attention-Enabled Double LSTMs (AE-DLSTMs)*

It is important for aspect-level sentiment classification to encode a sequence of word vectors into a sentence vector with aspect word information and extract the contextual semantic correlations with aspect words. Attention mechanism is applied to make use of hidden state of each input word and attention weights are generated. The structure of AE-DLSTMs are introduced in Section 3.3.2.

#### *3.3.1. The Word Representaion of Aspect Word*

Let $S = \{w_1, w_2, \ldots, w_{l_1}, w_{l_2}, \ldots, w_{l_m}, \ldots, w_n\}$ be the sentence consisting of $n$ words and the aspect word consisting of $m$ words that are $\{w_{l_1}, w_{l_2}, \ldots, w_{l_m}\}$. Let $E = \{e_1, e_2, \ldots, e_{l_1}, e_{l_2}, \ldots, e_{l_m}, \ldots, e_n\}$ be the word embedding of the sentence $S$. In general, the aspect word itself does not contain emotional information. With the processing of the aspect word with multi-words by LSTM, the previous sentiment information about the aspect word will be lost. In order to avoid the loss of information, we compress the words by average pooling them into one word. The word representation of aspect word $\{e_{l_1}, e_{l_2}, \ldots, e_{l_m}\}$ are averaged into $e_a$ as follows:

$$e_a = {}^1\!/_m \times \sum_{i=0}^{m} e_{l_m} \tag{7}$$

Thus, the word representation of the sentence $S$ are $E = \{e_1, \ldots, e_a, \ldots, e_n\}$.

#### *3.3.2. Structure of AE-DLSTMs*

In order to extract the contextual semantic information towards aspect words, DLSTMs are used to obtain the preceding and following sentiment information with regard to the aspect word. Thus, the sentiment features of both sides towards aspect word are considered more comprehensively in our model to achieve better results. In addition,

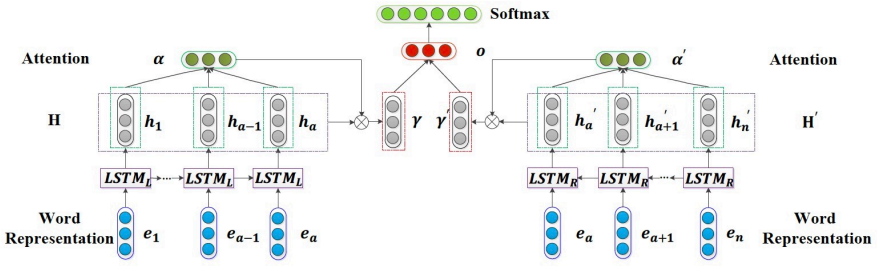


Figure 2: The structure of AE-DLSTMs for aspect-level sentiment classification. Word representation $\{e_1, e_2, \ldots, e_{a-1}, e_a, e_{a+1}, \ldots, e_n\}$ are the word embeddings of input words in a sentence whose length is $n$. $e_a$ denotes the aspect embedding. $H = \{h_1, \ldots, h_{a-1}, h_a\}$ and $H^{'} = \{h_n{}^{'}, \ldots, h_{a+1}{}^{'}, h_a{}^{'}\}$ are the hidden states of input words in double LSTM: $LSTM_L$ and $LSTM_R$ respectively. $\alpha$ is the attention weight. $h_a, h_a{}^{'}$ denote the hidden state of aspect word in two LSTMs, respectively.

we believe that taking the aspect word as the last hidden state can put more emphasis on the aspect words and make better use of the semantic information of the aspect word. It is difficult to find a common context range for each aspect word. Therefore, we set this range to the length of the entire sentence and models can correctly assign weights through learning. The greater the relationship between context words and aspect words, the greater the weight is. In this way, we can prevent the loss of useful sentimental information as much as possible.

Let $E_s = \{e_1, e_2, \ldots, e_{a-1}, e_a, e_{a+1}, \ldots, e_n\}$ be the word embedding of sentence $S$, $n$ be the length of the given sentence and $e_a$ be the embedding of the aspect word. AE-DLSTMs includes double LSTMs. $LSTM_L$ deals with the left part input sequence of aspect word in the forward direction while $LSTM_R$ processes the right part input sequence of aspect word in the reverse direction. Moreover, let $H = \{h_1, \ldots, h_{a-1}, h_a\}$, $H^{'} = \{h_n, \ldots, h_{a+1}, h_a{}^{'}\} \in \mathbb{R}^{n \times \mathrm{d}}$ be two matrix of hidden state vectors which denote the contextual semantic information produced in our models, where $d$ is the size of hidden layers, $n$ is the length of the given sentence and $h_a, h_a{}^{'}$ are the hidden states of the aspect word. The attention weights $\alpha, \alpha^{'}$ are produced by our attention mechanism which is introduced in Section 3.3.3 and weighted hidden states representation $\gamma$ is generated. The structure of AE-DLSTMs are illustrated in Figure 2.

$\alpha, \alpha^{'} \in \mathbb{R}^n$ are generated in Section 3.3.3, and hidden state representation $\gamma, \gamma^{'} \in$

$\mathbb{R}^n$ are computed as follows:

$$\gamma = \alpha^T H \tag{8}$$

$$\gamma^{'} = \alpha^{T^{'}} H^{'} \tag{9}$$

the final sentence representation is given by:

$$O = tanh\left(W_p\gamma + W_p^{'}\gamma'\right) \tag{10}$$

where $W_p, W_p^{'} \in \mathbb{R}^{d\times d}$ are parameters to be learned during training, $O$ is considered as the representation of the sentence features [14, 73].

Finally, a softmax layer is followed to classify the result:

$$y = softmax\left(W_s O + b_s\right) \tag{11}$$

where $W_s \in \mathbb{R}^{c\times d}$ and $b_s \in \mathbb{R}^c$ are the parameters for softmax layer, and $c$ is the number of categories.

### *3.3.3. The Attention Mechanism*

We use the DLSTMs structure to exact the preceding and following sentiment information of aspect word, but we can not fully obtain the contextual semantic information of the aspect word using only the hidden state of the last time step. The more important the word is for the aspect word, the bigger weight it occupies. Thus, to make better use of the hidden state at each time step, attention mechanism is designed and attention weights are assigned to different hidden states according to their contributions.

Let $\gamma$, $\gamma^{'} \in \mathbb{R}^n$ be the weighted vector computed by summing the weighted hidden states in double LSTMs which are $LSTM_L$ and $LSTM_R$, respectively. $H = \{h_1, \ldots, h_{a-1}, h_a\}$, $H^{'} = \{h_n, \ldots, h_{a+1}, {h_a}^{'}\} \in \mathbb{R}^{n\times \mathrm{d}}$ are two matrix of hidden state vectors produced in the double LSTMs, respectively, representing the contextual semantic information in the two diverse direction of the aspect word.

For the purpose of incorporating aspect words information, $h_a$ and ${h_a}^{'}$, which contain the semantic information of the aspect words in diverse direction, are concatenated into $[{h_a}^T, {h'}_a^T]^T$, which is called Aspect Word Semantic Vector. The attention weights for each hidden states are computed according to the semantic similarity with the Aspect

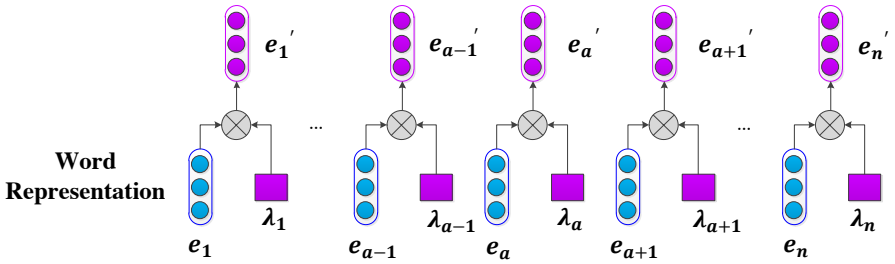


Figure 3: The input structure of AELA-DLSTMs for aspect-level sentiment classification. Word representation $\{e_1, e_2, \ldots, e_{a-1}, e_a, e_{a+1}, \ldots, e_n\}$ are the word embeddings of input words in a sentence whose length is $n$. $\left\{e_1{}', e_2{}', \ldots, e_{a-1}{}', e_a{}', e_{a+1}{}', \ldots, e_n{}'\right\}$ denote the new aspect embedding weighted with location weights.

Word Semantic Vector. We map the hidden states $H$ and $H'$ into the semantic space by multiplying the matrix $W_m \in \mathbb{R}^{d\times 2d}$, $W_m{}' \in \mathbb{R}^{d\times 2d}$. And we obtain the semantic similarity by inner product with the Aspect Word Semantic Vector. The attention weights $\alpha, \alpha'$ are designed as follows:

$$\alpha = soft\max(H \times W_m \times \begin{bmatrix} h_a \\ h_a{}' \end{bmatrix}) \tag{12}$$

$$\alpha' = soft\max(H' \times W_m \times \begin{bmatrix} h_a \\ h_a{}' \end{bmatrix}) \tag{13}$$

where $W_m \in \mathbb{R}^{d\times 2d}$, $W_m{}' \in \mathbb{R}^{d\times 2d}$ are trained with models.

### *3.4. Attention-Enabled and Location-Aware Double LSTMs (AELA-DLSTMs)*

We observe that the key words which express the sentiment for aspect word to a maximum degree are always close to the aspect word itself [24, 74]. The closer to the aspect words, the more correlation information they may contain. For example, in the sentence "The ambience was nice, but the service was awful.", the context word "nice" is the key word for the aspect word "ambience" while "awful" is the key word for the aspect word "service". As the context word "nice" is closer to aspect word "ambience" than the context word "awful", "nice" should have more contributions to the sentiment

polarity for the aspect word "ambience" while "awful" have more contributions to "service".

We further notice that the key words for the aspect word always both only locate in one side of aspect word. For the above example, the context word "nice" and "awful" both locate on the right of the aspect word "ambience" and "service" respectively, thus, in this example, the context words on the right of aspect word should have greater influence than context words on the left. To this end, two parameters are designed for both sides according to their respective importance, which are also to be learned through training.

Directly inputting the word vector into the model and generating the representations of sentence vector will contain information that is not related to the aspect word. Therefore, we use the location weight to emphasize the related sentiment information towards aspect words in the process of generating sentence vector. And in the meantime, the words unrelated to the aspect are also suppressed accordingly. Given the fact that location information is beneficial to obtain a better result, input word embedding vectors are weighted by location weights. The settings of the location weights are inspired by Luong et al. [23], who used the probability density function of the Gaussian distribution. Since we expect the weights on words far away from the aspect word to fall faster, which helps to prevent the interference of information that is not related to aspects, we use the Laplacian probability density function to obtain the location weights. The Laplacian probability density function falls faster for the area away from the aspect word compared to the Gaussian probability density function, which can prevent involving irrelevant information.

The location weights $\lambda$ corresponding to input word embedding vectors $E_s = \{e_1, e_2, \ldots, e_{a-1}, e_a, e_{a+1}, \ldots, e_n\}$ are defined as follows:

$$\lambda = [\lambda_1, \lambda_2, \ldots, \lambda_{a-1}, \lambda_a, \lambda_{a+1}, \ldots, \lambda_n] \tag{14}$$

$$\lambda = \begin{cases} \alpha_l exp(-\beta_l |i-a|), & i < a \\ \alpha_m, & i = a \\ \alpha_r exp(-\beta_r |i-a|), & i > a \end{cases} \tag{15}$$

where $i$ is the index of words in the sentence, $a$ is the index of the aspect word, different input location weights are generated for different locations of the words. $\alpha_l$ and $\beta_l$ are the parameters for context words on the left of the aspect word while $\alpha_r$ and $\beta_r$ are the parameters for context words on the right, $\alpha_m$ is for the aspect word, all of which are trained with the models. For example, if input location weights for the left context words are larger than the right when training, it indicates that the left context words are of greater importance to the aspect word. Then, the input location weights are incorporated to the input word embedding vector as described below:

$$e_i^{'} = \lambda_i \times e_i \tag{16}$$

where $e_i^{'}$ is the $i_{th}$ new word embedding vector for the $i_{th}$ input words weighted by input location weights. The input structure of AELA-DLSTMs is shown as Figure 3.

AELA-DLSTMs are designed to make the input word weighted with the location weights and further improve the results with location information.

### *3.5. Model Training*

AE-DLSTMs, AELA-DLSTMs and the compared baseline models are trained in an end-to-end way by back propagation in a supervised learning. The loss function is the cross-entropy loss [75]. Meanwhile, dropout [76] and the L2-regularization penalty [77] are incorporated. Models are trained by minimizing the loss between the target distribution and the predicted distribution. Let $y$ be the target distribution and $\hat{y}$ be the distribution results. The loss function can be defined as follows:

$$loss = -\sum_i \sum_j y_i^j log\hat{y}_i^j + \sum_{\theta \in \Phi} \lambda \|\theta\|^2 \tag{17}$$

where $i$ is the index of sentence, $j$ is the index of the classification, $\lambda$ is the L2-regularization coefficient and $\Phi$ denotes the all parameters.

## 4. Performance Evaluations

The performance of the proposed models are evaluated in this section, following with statements about the experimental datasets, setting details and result analysis.

### *4.1. Datasets*

Experiment are conducted on both English and Chinese datasets. English datasets include Restaurant and Laptops datasets and Chinese dataset includes Taobao[2] dataset.

**Restaurant and Laptops Datasets:** The Restaurant and Laptops datasets are from SemEval 2014 Task 4[3] [65] obtained from Restaurant domain reviews and Laptops domain reviews. The total numbers of samples in training data and test data in Restaurant dataset are 3608 and 1120 while that in Laptops dataset are 2328 and 638, respectively.

**Taobao Dataset[4]:** We collected reviews from 10 kinds of goods (Shoes, food, Kids, woman's clothing, jewelry, outdoor products, menswear, building materials, office supplies, luggage) reviews on Taobao website. These reviews with multiple aspect words were reviewed by the users who got goods and tagged three sentiment polarities (positive, negative and neutral) by our annotators. We selected 10,030 positive reviews, 1,980 neural reviews and 2,812 negative reviews from a large number of annotation samples. Each review of the dataset was extracted by crawling from Taobao website, and was preprocessed such as word segmentation and removing stop words. The percentages of training sets and test sets are 90 percent and 10 percent. The training data and test data in Taobao Dataset are 13339 and 1483 samples. The percentages of positive, negative and neutral polarities in training and test sets are both 68 percent, 19 percent, 13 percent, respectively.

The above datasets all have three sentiment polarities for the aspect word: positive, negative and neutral. The models are trained in three-way and two-way sentiment classification. The samples of three-way classification includes positive, negative and neutral polarities while the two-way remove the samples with neutral polarity. Models are trained on training dataset and the accuracy is evaluated on test dataset. The statistics of the above datasets are shown as Table 1.

[2]Taobao, as China's dominant online trading platform with over 400 million users, is the largest online retail platform in the world. The website of it is https://www.taobao.com/.

[3]The introduction about SemEval 2014 can be obtained from http://alt.qcri.org/semeval2014/.

[4]The dataset can be accessed from `https://github.com/rxt2012kc/Taobao-Dataset`.

| | Restaurant | | Laptops | | Taobao | |
|---|---|---|---|---|---|---|
| | Train | Test | Train | Test | Train | Test |
| Pos | 2164 | 728 | 994 | 341 | 9027 | 1003 |
| Neu | 637 | 196 | 464 | 169 | 1782 | 198 |
| Neg | 807 | 196 | 870 | 128 | 2530 | 282 |
| Total | 3608 | 1120 | 2328 | 638 | 13339 | 1483 |

Table 1: The Statistics of Datasets

### *4.2. Experimental Settings*

In the experiments, pre-trained word vectors trained by Glove[5] [41] are used to initialize English datasets, the dimension of which is 300 and are trained on Common Crawl Corpus size. For Chinese dataset, pre-trained word vectors trained by Word2vec [33] are trained on Taobao Review and used for initializing, the dimension of which is 50.

The size of hidden layer is the same as the word embedding dimensions and the length of attention weights is the same as the length of input sentences. Other parameters are randomized with uniform distribution $U(-\varepsilon, \varepsilon)$. The L2-regularization weight is set as 0.001 and the learning rate is set as 0.01. For AELA-DLSTMs, the parameter $\alpha$ is initialized with 1 and $\beta$ is initialized with 0. Theano [78] framework is applied to implement neural network models in the experiment. AdaGrad Optimizer [79, 80] is used to train models with mini-batch strategy and each batch consists of 25 samples.

### *4.3. Comparison between our models and baseline models*

AE-DLSTMs and AELA-DLSTMs models are compared with the following baseline methods on both English and Chinese datasets.

**LSTM:** Standard LSTM based on Recurrent Neural Network is used for a sentence inputting sequentially and only uses the last hidden state to calculate the output [12].

**BiLSTM:** BiLSTM contains two Standard LSTM. One can obtain output in a forward scan while the other can obtain output in a backward scan of the text [44].

[5]Pre-trained word vectors of Glove can be obtained from http://nlp.stanford.edu/projects/glove/.

| | Restaurant | Laptops | Taobao |
|---|---|---|---|
| LSTM | 89.25 | 83.71 | 82.26 |
| BiLSTM | 89.43 | 85.06 | 82.88 |
| TD-LSTM | 89.83 | 86.35 | 92.08 |
| TC-LSTM | 88.63 | 85.16 | 92.47 |
| MemNet(9) | 89.07 | 85.55 | 85.21 |
| ATAE-LSTM | 89.58 | 86.60 | 93.77 |
| IAN | 89.85 | 86.57 | 93.96 |
| GCAE | 90.56 | 86.94 | 94.51 |
| AE-DLSTMs | **91.69** | **89.98** | **96.42** |

Table 2: The Percentage Accuracies of AE-DLSTMs and Other Baseline Models in Two-way Classification

**TD-LSTM:** TD-LSTM uses two LSTM neural networks, a forward one and a backward one toward the aspects to model the preceding and following contexts respectively [26].

**TC-LSTM:** TC-LSTM extends TD-LSTM by appending a target embedding into each word input vector which explicitly utilizes the connections between target word and each context word [26].

**MemNet:** MemNet is a deep memory network for aspect-level sentiment classification [24]. MemNet(9) that contains 9 computational layers is conducted in the experiment.

**ATAE-LSTM:** ATAE-LSTM model appends the input aspect embedding into each word input vector and uses attention mechanism to calculate the output based on Standard LSTM [14].

**IAN:** IAN model interactively learns attentions in the contexts and targets, and generates the representations for targets and contexts separately [71].

**GCAE:** GCAE model is based on convolutional neural networks and gating mechanisms, which has two separate convolutional layers on the top of the embedding layer, whose outputs are combined by gating units [66].

| | Restaurant | Laptops | Taobao |
|---|---|---|---|
| LSTM | 74.30 | 67.24 | 71.75 |
| BiLSTM | 76.39 | 68.26 | 72.56 |
| TD-LSTM | 76.43 | 67.24 | 82.34 |
| TC-LSTM | 76.01 | 65.62 | 82.89 |
| MemNet(9) | 75.89 | 65.80 | 74.51 |
| ATAE-LSTM | 75.18 | 67.76 | 86.18 |
| IAN | 76.87 | 68.34 | 85.90 |
| GCAE | 77.28 | 69.14 | 86.37 |
| AE-DLSTMs | **79.57** | **72.10** | **88.47** |

Table 3: The Percentage Accuracies of AE-DLSTMs and Other Baseline Models in Three-way Classification

For model TD-LSTM, TC-LSTM, MemNet[6] and ATAE-LSTM[7], the source codes of original papers are used to calculate the accuracy results on three datasets. The percentage accuracies of the above models in Two-way and Three-way classification are given in Table 2 and Table 3.

As is shown in Table 2 and Table 3, AE-DLSTMs acquire the best results on all datasets and far surpass other baselines including TC-LSTM, TD-LSTM, MemNet, ATAE-LSTM, IAN and GCAE. For instance, in Laptops dataset, the accuracy of our AE-DLSTMs model is almost 3.04 percent higher than baseline methods in Two-way classification and 2.96 percent higher in Three-way classification.

On the one hand, AE-DLSTMs apply DLSTMs structure which processes a sentence in two directions, making better use of contextual semantic information from both the preceding and following contexts towards aspect words. On the other hand, the attention weights generated in our models are more efficient. Compared to ATAE-LSTM, IAN and GCAE, our model can extract more contextual semantic information by using DLSTMs structure while ATAE-LSTM and IAN using LSTM structure. AE-DLSTMs

[6]The source code of TD-LSTM, TC-LSTM and MemNet are publicly available at http://ir.hit.edu.cn/dytang.

[7]The source code of ATAE-LSTM are publicly available at http://www.aihuang.org/p/publications.html.

| | Restaurant | Laptops | Taobao |
|---|---|---|---|
| AE-DLSTMs | 91.69 | 89.98 | 96.42 |
| AELA-DLSTMs | **92.23** | **91.53** | **97.20** |
| AE-DLSTMs+Absolute_Loc | 90.04 | 91.48 | 96.60 |
| AE-DLSTMs+Gaussian_Loc | 91.87 | 91.51 | 96.87 |

Table 4: The Percentage Accuracies of AE-DLSTMs, AELA-DLSTMs, AE-DLSTMs+Absolute_Loc and AE-DLSTMs+Gaussian_Loc in Two-way Classification

| | Restaurant | Laptops | Taobao |
|---|---|---|---|
| AE-DLSTMs | 79.57 | 72.10 | 88.47 |
| AELA-DLSTMs | **80.35** | **73.91** | **90.22** |
| AE-DLSTMs+Absolute_Loc | 77.50 | 71.30 | 88.67 |
| AE-DLSTMs+Gaussian_Loc | 79.69 | 72.87 | 89.13 |

Table 5: The Percentage Accuracies of AE-DLSTMs, AELA-DLSTMs, AE-DLSTMs+Absolute_Loc and AE-DLSTMs+Gaussian_Loc in Three-way Classification

achieve improved performance compared with TC-LSTM, TD-LSTM, MemNet, IAN and GCAE, demonstrating that the attention generating method used in our model can indeed enhance the aspect information and take full advantages of the correlation between aspect words and their context words.

### *4.4. Effects of Input Location Information*

Experiments are conducted on AE-DLSTMs, AELA-DLSTMs, AE-DLSTMs+ Absolute_Loc and AE-DLSTMs+Gaussian_Loc. AE-DLSTMs+Absolute_Loc combines AE-DLSTMs with the location weights method proposed by Tang et al. [24] and Chen et al. [74], which define the location of a context word as its absolute distance with the aspect word in the original sentence sequence. AE-DLSTMs+Gaussian_Loc combines AE-DLSTMs with the location weights on Gaussian probability density function [23]. The experimental results are shown in Table 4 and Table 5.

In Table 4 and Table 5, AELA-DLSTMs perform better than other two baselines in all datasets. Comparing the accuracies of AE-DLSTMs and AELA-DLSTMs, it can be

| | Restaurant | Laptops |
|---|---|---|
| Aspect word with average pooling | **91.69** | **89.98** |
| Aspect word with none-average pooling | 90.89 | 89.54 |

Table 6: Two-way (positive and negative) percentage accuracies of AE-DLSTMs with aspect word averaged and aspect word none-averaged.

| | Restaurant | Laptops |
|---|---|---|
| Aspect word with average pooling | **79.57** | **72.10** |
| Aspect word with none-average pooling | 79.21 | 71.78 |

Table 7: Three-way (positive, neutral and negative) percentage accuracies of AE-DLSTMs with aspect word averaged and aspect word none-averaged.

concluded that the model with location weights is beneficial to classify the sentiment polarity of aspect words and the location information plays an effective role in our tasks. The comparison of AELA-DLSTMs and AE-DLSTMs+Absolute_Loc illustrates that the location weights generated in our models are more suitable for aspect-level sentiment classification, while the location weights generated in Tang et al. [24] ignore the differences between the two sides of aspect word. By comparing the accuracies of AELA-DLSTMs and AE-DLSTMs+Gaussian_Loc, it indicates that the Laplacian probability density function is more suitable for our task than Gaussian probability density function.

### *4.5. Effects of Word Representation of Aspect Word*

The word representation of aspect word $\{e_{l_1}, e_{l_2}, \ldots, e_{l_m}\}$ are averaged into one word embedding $e_a$ in this work. Experiments are conducted on AE-DLSTMs with aspect word average pooling and aspect word none-average pooling in two datasets: Restaurant and Laptops, because the aspect word of these two datasets may contain multiple words, such as “thin crusted pizza”, “battery life”, while the aspect word of Taobao dataset only consists of one word, such as “质量”, “服务”. The experimental results are shown in Table 6 and Table 7.

In Table 6 and Table 7, AELA-DLSTMs with aspect word averaged perform better


Figure 4: Attention Visualizations on an aspect word in sentence. The sample "We always have a delicious meal and always leave feeling satisfied." has an aspect word "meal". The color map in the right shows the value of attention weights and $\alpha$ denotes the importance of input words towards aspect word.

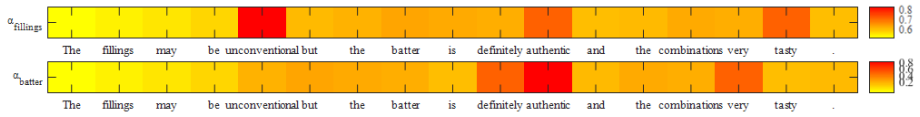


Figure 5: Attention Visualizations on different aspect words in the same sentence. The sample "The fillings may be unconventional but the batter is definitely authentic and the combinations very tasty." has aspect words "fillings" and "batter". This picture demonstrates two cases where aspect words are "fillings" and "batter" respectively. The color map in the right shows the value of attention weights and $\alpha$ denotes the importance of input words towards aspect word.

than with aspect word none-averaged in almost all datasets. In fact that the aspect word itself does not have any emotional information, and multi aspect words will make the previous emotional characteristics diluted in the process of LSTM model. Thus, compressing the aspect words can retain the emotional information extracted for the aspect word as much as possible.

### *4.6. Case Study and Visualize Models*

We visualize the attention weights of hidden states to get a better understanding of how our models work in the aspect-level sentiment classification tasks. As the sentence "We always have a delicious meal and always leave feeling satisfied." has an aspect word "meal". As is illustrated in Figure 4, the input word "delicious" occupies bigger weights than other words in this sentence, because it is crucial for the aspect word "meal".

As the sentence "The fillings may be unconventional but the batter is definitely authentic and the combinations very tasty." has three aspect words "fillings", "batter" and "combinations". our models can make better use of different aspect words more effectively and extract correlations between aspect words towards their contexts more

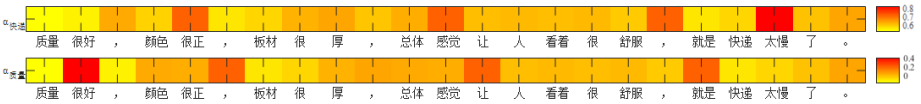


Figure 6: Attention Visualizations on different aspect words in the same sentence. The sample "质量很好, 颜色很正，板材很厚，总体感觉让人看着很舒服，就是快递太慢了" also has two aspect words "质量" and "快递". This picture demonstrates two cases where aspect words are " 质量" and "快递" respectively. The color map in the right shows the value of attention weights and $\alpha$ denotes the importance of input words towards aspect word.


Figure 7: AELA-DLSTM Location Weights Visualizations. The sample is "The food was mediocre at best but it was the horrible service that made me vow never to go back." and the aspect word is "service". The color map in the right shows the value of the input location weights and $\lambda$ denotes the location information of the input words towards the aspect word.

precisely in the same sentence. As is illustrated in Figure 5, we take two aspect words "fillings" and "batter" as an example, the input word "unconventional" occupies bigger weights to the aspect word "fillings" as input word "authentic" does for aspect word "batter", which represents that "authentic" plays a more important role to the aspect word "batter" and "unconventional" is more essential to "fillings".

And the sentence "质量很好, 颜色很正, 板材很厚, 总体感觉让人看着很舒服，就是快递太慢了" also has four aspect words "质量", " 颜色", "板材" and "快递". As is illustrated in Figure 6, we take two aspect words "质量" and "快递" as an example, the input word "很好" occupies bigger weights to the aspect word "质量" as input word "太慢" does for aspect word "快递", which represents that "很好" plays a more important role to the aspect word "质量" and "太慢" is more essential to "快递".

AELA-DLSTM assigns location weights for input word according to their location information towards aspect word. To indicate how location information works and improve the performance, location weights are visualized in Figure 7. In the sample "The food was mediocre at best but it was the horrible service that made me vow never to go back.", whose aspect word of which is "service", the location weights are trained with models as they are initialized with the same value. It can be observed that "horrible"

is the key word for aspect word, locating on the right side of it, having bigger location weight than other left side words in training. In addition, the words closer to aspect word occupy higher weights than the farther ones. The above location weights contribution indicates that location information takes effects on the results and is helpful for aspect-level sentiment classification.

## 5. Conclusion

In this paper, we propose two models named AE-DLSTMs and AELA-DLSTMs for aspect-level sentiment classification. AE-DLSTMs make better use of the contextual semantic information towards aspect words and effectively take better advantage of the correlation between aspect words and their context words. As the location information has considerable influence on classification results, AELA-DLSTMs incorporate the location information of context words with regard to aspect words to generate the weighted input word vectors. We train our models in an end-to-end way on both English and Chinese datasets in Two-way and Three-way classification. The experimental results have demonstrated that our models achieve remarkable performances and outperform all the baseline models. Since attention weights generating method and models are effective, our models and method can be applied to NLP Attention Tasks, such as Machine Translation, for obtaining further better performances.

## ACKNOWLEDGMENTS

This work is supported in part by National key research and development program of China (2016QY01W0200) and National Natural Science Foundation of China: Research on Differentially private frequent pattern mining (61502047). The work is also supported by State Key Laboratory of Networking and Switching Technology of Beijing University of Posts and Telecommunications.

## References

[1] B. Pang, L. Lee, et al., Opinion mining and sentiment analysis, Foundations and Trends® in Information Retrieval 2 (1–2) (2008) 1–135.

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

[3] B. Pang, L. Lee, S. Vaithyanathan, Thumbs up?: sentiment classification using machine learning techniques, Proceedings of Emnlp (2002) 79–86.

[4] P. D. Turney, Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews, in: Proceedings of the 40th annual meeting on association for computational linguistics, Association for Computational Linguistics, 2002, pp. 417–424.

[5] M. Hu, B. Liu, Mining and summarizing customer reviews, in: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2004, pp. 168–177.

[6] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, Proceedings of International Workshop on Semantic Evaluation at (2014) 27–35.

[7] C. Zhou, C. Sun, Z. Liu, F. Lau, A c-lstm neural network for text classification, arXiv preprint arXiv:1511.08630.

[8] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, R. Ward, Deep sentence embedding using long short-term memory networks: Analysis and application to information retrieval, IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP) 24 (4) (2016) 694–707.

[9] D. Wang, E. Nyberg, A long short-term memory model for answer sentence selection in question answering, in: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers), Vol. 2, 2015, pp. 707–712.

[10] J. Cheng, M. Lapata, Neural summarization by extracting sentences and words, arXiv preprint arXiv:1603.07252.

[11] I. Sutskever, O. Vinyals, Q. V. Le, Sequence to sequence learning with neural networks, in: Advances in neural information processing systems, 2014, pp. 3104–3112.

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

[13] K. S. Tai, R. Socher, C. D. Manning, Improved semantic representations from tree-structured long short-term memory networks, arXiv preprint arXiv:1503.00075.

[14] Y. Wang, M. Huang, L. Zhao, et al., Attention-based lstm for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.

[15] Z. Huang, W. Xu, K. Yu, Bidirectional lstm-crf models for sequence tagging, arXiv preprint arXiv:1508.01991.

[16] C. Dyer, M. Ballesteros, W. Ling, A. Matthews, N. A. Smith, Transition-based dependency parsing with stack long short-term memory, arXiv preprint arXiv:1505.08075.

[17] Y.-F. Ma, H.-J. Zhang, Contrast-based image attention analysis by using fuzzy growing, in: Proceedings of the eleventh ACM international conference on Multimedia, ACM, 2003, pp. 374–381.

[18] V. Mnih, N. Heess, A. Graves, et al., Recurrent models of visual attention, in: Advances in neural information processing systems, 2014, pp. 2204–2212.

[19] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, Y. Bengio, Show, attend and tell: Neural image caption generation with visual attention, in: International Conference on Machine Learning, 2015, pp. 2048–2057.

[20] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, Y. Bengio, Attention-based models for speech recognition, in: Advances in neural information processing systems, 2015, pp. 577–585.

[21] D. Bahdanau, J. Chorowski, D. Serdyuk, P. Brakel, Y. Bengio, End-to-end attention-based large vocabulary speech recognition, in: Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on, IEEE, 2016, pp. 4945–4949.

[22] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473.

[23] M.-T. Luong, H. Pham, C. D. Manning, Effective approaches to attention-based neural machine translation, arXiv preprint arXiv:1508.04025.

[24] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, arXiv preprint arXiv:1605.08900.

[25] M. Yang, W. Tu, J. Wang, F. Xu, X. Chen, Attention based lstm for target dependent sentiment classification., in: AAAI, 2017, pp. 5013–5014.

[26] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, arXiv preprint arXiv:1512.01100.

[27] T. Nasukawa, J. Yi, Sentiment analysis: Capturing favorability using natural language processing, in: Proceedings of the 2nd international conference on Knowledge capture, ACM, 2003, pp. 70–77.

[28] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Ng, C. Potts, Recursive deep models for semantic compositionality over a sentiment treebank, in: Proceedings of the 2013 conference on empirical methods in natural language processing, 2013, pp. 1631–1642.

[29] E. Martínez-Cámara, M. T. Martín-Valdivia, L. A. Urena-López, A. R. Montejo-Ráez, Sentiment analysis in twitter, Natural Language Engineering 20 (1) (2014) 1–28.

[30] D. Tang, B. Qin, T. Liu, Document modeling with gated recurrent neural network for sentiment classification, in: Proceedings of the 2015 conference on empirical methods in natural language processing, 2015, pp. 1422–1432.

[31] X. A. Wang, F. Xhafa, X. Luo, S. Zhang, Y. Ding, A privacy-preserving fuzzy interest matching protocol for friends finding in social networks, Soft Computing 22 (8) (2018) 2517–2526.

[32] Z. Gao, Y. Sun, X. Cui, Y. Wang, Y. Duan, X. A. Wang, Privacy-preserving hybrid k-means, International Journal of Data Warehousing and Mining (IJDWM) 14 (2) (2018) 1–17.

[33] Y. Kim, Convolutional neural networks for sentence classification, arXiv preprint arXiv:1408.5882.

[34] N. Kalchbrenner, E. Grefenstette, P. Blunsom, A convolutional neural network for modelling sentences, arXiv preprint arXiv:1404.2188.

[35] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, B. Qin, Learning sentiment-specific word embedding for twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), Vol. 1, 2014, pp. 1555–1565.

[36] Y. Ren, Y. Zhang, M. Zhang, D. Ji, Context-sensitive twitter sentiment classification using neural network., in: AAAI, 2016, pp. 215–221.

[37] S. Rosenthal, N. Farra, P. Nakov, Semeval-2017 task 4: Sentiment analysis in twitter, in: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017), 2017, pp. 502–518.

[38] Y. Bengio, R. Ducharme, P. Vincent, C. Jauvin, A neural probabilistic language model, Journal of machine learning research 3 (Feb) (2003) 1137–1155.

[39] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, J. Dean, Distributed representations of words and phrases and their compositionality, in: Advances in neural information processing systems, 2013, pp. 3111–3119.

[40] Q. Le, T. Mikolov, Distributed representations of sentences and documents, in: International Conference on Machine Learning, 2014, pp. 1188–1196.

[41] J. Pennington, R. Socher, C. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), 2014, pp. 1532–1543.

[42] T. Mikolov, M. Karafiát, L. Burget, J. Černockỳ, S. Khudanpur, Recurrent neural network based language model, in: Eleventh Annual Conference of the International Speech Communication Association, 2010.

[43] W. Zaremba, I. Sutskever, O. Vinyals, Recurrent neural network regularization, arXiv preprint arXiv:1409.2329.

[44] S. Lai, L. Xu, K. Liu, J. Zhao, Recurrent convolutional neural networks for text classification., in: AAAI, Vol. 333, 2015, pp. 2267–2273.

[45] L. Arras, G. Montavon, K.-R. Müller, W. Samek, Explaining recurrent neural network predictions in sentiment analysis, arXiv preprint arXiv:1706.07206.

[46] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (2) (1994) 157–166.

[47] M. Sundermeyer, R. Schlüter, H. Ney, Lstm neural networks for language modeling, in: Thirteenth Annual Conference of the International Speech Communication Association, 2012.

[48] H. Sak, A. Senior, F. Beaufays, Long short-term memory recurrent neural network architectures for large scale acoustic modeling, in: Fifteenth annual conference of the international speech communication association, 2014.

[49] A. Graves, A.-r. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, IEEE, 2013, pp. 6645–6649.

[50] R. Socher, D. Chen, C. D. Manning, A. Ng, Reasoning with neural tensor networks for knowledge base completion, in: Advances in neural information processing systems, 2013, pp. 926–934.

[51] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, C. Dyer, Neural architectures for named entity recognition, arXiv preprint arXiv:1603.01360.

[52] E. Kiperwasser, Y. Goldberg, Simple and accurate dependency parsing using bidirectional lstm feature representations, arXiv preprint arXiv:1603.04351.

[53] X. Ding, B. Liu, P. S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the 2008 international conference on web search and data mining, ACM, 2008, pp. 231–240.

[54] S. Blair-Goldensohn, K. Hannan, R. McDonald, T. Neylon, G. A. Reis, J. Reynar, Building a sentiment summarizer for local service reviews, in: WWW workshop on NLP in the information explosion era, Vol. 14, 2008, pp. 339–348.

[55] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, Target-dependent twitter sentiment classification, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1, Association for Computational Linguistics, 2011, pp. 151–160.

[56] S. M. Mohammad, S. Kiritchenko, X. Zhu, Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets, arXiv preprint arXiv:1308.6242.

[57] C. Brun, D. N. Popa, C. Roux, Xrce: Hybrid classification for aspect-based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 838–842.

[58] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, Nrc-canada-2014: Detecting aspects and sentiment in customer reviews, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 437–442.

[59] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, L. Tounsi, Dcu: Aspect-based polarity classification for semeval task 4, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 223–229.

[60] R. Collobert, J. Weston, A unified architecture for natural language processing: Deep neural networks with multitask learning, in: Proceedings of the 25th international conference on Machine learning, ACM, 2008, pp. 160–167.

[61] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa, Natural language processing (almost) from scratch, Journal of Machine Learning Research 12 (Aug) (2011) 2493–2537.

[62] Y. Kim, Y. Jernite, D. Sontag, A. M. Rush, Character-aware neural language models., in: AAAI, 2016, pp. 2741–2749.

[63] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), Vol. 2, 2014, pp. 49–54.

[64] D.-T. Vo, Y. Zhang, Target-dependent twitter sentiment classification with rich automatic features., in: IJCAI, 2015, pp. 1347–1353.

[65] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, A.-S. Mohammad, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, et al., Semeval-2016 task 5: Aspect based sentiment analysis, in: Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016), 2016, pp. 19–30.

[66] W. Xue, T. Li, Aspect based sentiment analysis with gated convolutional networks, arXiv preprint arXiv:1805.07043.

[67] A. M. Rush, S. Chopra, J. Weston, A neural attention model for abstractive sentence summarization, arXiv preprint arXiv:1509.00685.

[68] J. Li, M.-T. Luong, D. Jurafsky, A hierarchical neural autoencoder for paragraphs and documents, arXiv preprint arXiv:1506.01057.

[69] K. M. Hermann, T. Kocisky, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, P. Blunsom, Teaching machines to read and comprehend, in: Advances in Neural Information Processing Systems, 2015, pp. 1693–1701.

[70] A. Kumar, O. Irsoy, P. Ondruska, M. Iyyer, J. Bradbury, I. Gulrajani, V. Zhong, R. Paulus, R. Socher, Ask me anything: Dynamic memory networks for natural language processing, in: International Conference on Machine Learning, 2016, pp. 1378–1387.

[71] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, arXiv preprint arXiv:1709.00893.

[72] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji, X. Wang, Modeling mention, context and entity with neural networks for entity disambiguation., in: IJCAI, 2015, pp. 1333–1339.

[73] T. Rocktäschel, E. Grefenstette, K. M. Hermann, T. Kočiskỳ, P. Blunsom, Reasoning about entailment with neural attention, arXiv preprint arXiv:1509.06664.

[74] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452–461.

[75] L.-Y. Deng, The cross-entropy method: a unified approach to combinatorial optimization, monte-carlo simulation, and machine learning (2006).

[76] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, The Journal of Machine Learning Research 15 (1) (2014) 1929–1958.

[77] S. Ioffe, C. Szegedy, Batch normalization: accelerating deep network training by reducing internal covariate shift (2015) 448–456.

[78] F. Bastien, P. Lamblin, R. Pascanu, J. Bergstra, I. Goodfellow, A. Bergeron, N. Bouchard, D. Warde-Farley, Y. Bengio, Theano: new features and speed improvements, arXiv preprint arXiv:1211.5590.

[79] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research 12 (Jul) (2011) 2121–2159.

[80] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, A. Senior, P. Tucker, K. Yang, Q. V. Le, et al., Large scale distributed deep networks, in: Advances in neural information processing systems, 2012, pp. 1223–1231.

**Kai Shuang** is an associate professor in Beijing University of Posts and Telecommunications. His research interests include deep learning, natural language processing, image processing, cloud computing and big data technology. Contact him at shuangk@bupt.edu.cn.

**Xintao Ren** is a research assistant and a Master Degree Candidate in Beijing University of Posts and Telecommunications. Her research interests include deep learning, natural language processing, sentiment analysis. Contact her at rxt2012kc@bupt.edu.cn.

**Qianqian Yang** is a research assistant and a Master Degree Candidate in Beijing University of Posts and Telecommunications. Her research interests include deep learning, natural language processing, sentiment analysis. Contact her at echo_yang@bupt.edu.cn.

**Jonathan Loo** is a professor in University of West London. His research interests include deep learning, natural language processing, information technology, network security and computer applications. Contact him at jonathan.loo@uwl.ac.uk.

**Rui Li** is a research assistant and a Master Degree Candidate in Beijing University of Posts and Telecommunications. His research interests include deep learning, natural language processing, text analysis. Contact him at lirui@bupt.edu.cn.