Aspect-level Sentiment Classification with HEAT (HiErarchical ATtention) Network

Jiajun Cheng[*], Shenglin Zhao[†], Jiani Zhang[†], Irwin King[†], Xin Zhang[*], Hui Wang[*]
[*]College of Information Systems and Management, National University of Defense Technology
Changsha, Hunan, China
{jiajun.cheng,huiwang}@nudt.edu.cn,ijunzhang@hotmail.com
[†]Department of Computer Science and Engineering, The Chinese University of Hong Kong
Shatin, N.T., Hong Kong
{slzhao,jnzhang,king}@cse.cuhk.edu.hk

ABSTRACT

Aspect-level sentiment classification is a fine-grained sentiment analysis task, which aims to predict the sentiment of a text in different aspects. One key point of this task is to allocate the appropriate sentiment words for the given aspect. Recent work exploits attention neural networks to allocate sentiment words and achieves the state-of-the-art performance. However, the prior work only attends to the sentiment information and ignores the aspect-related information in the text, which may cause mismatching between the sentiment words and the aspects when an unrelated sentiment word is semantically meaningful for the given aspect. To solve this problem, we propose a HiErarchical ATtention (HEAT) network for aspect-level sentiment classification. The HEAT network contains a hierarchical attention module, consisting of aspect attention and sentiment attention. The aspect attention extracts the aspect-related information to guide the sentiment attention to better allocate aspect-specific sentiment words of the text. Moreover, the HEAT network supports to extract the aspect terms together with aspect-level sentiment classification by introducing the Bernoulli attention mechanism. To verify the proposed method, we conduct experiments on restaurant and laptop review data sets from SemEval at both the sentence level and the review level. The experimental results show that our model better allocates appropriate sentiment expressions for a given aspect benefiting from the guidance of aspect terms. Moreover, our method achieves better performance on aspect-level sentiment classification than state-of-the-art models.

CCS CONCEPTS

- **Information systems → Sentiment analysis**; *Social tagging*;
- **Computing methodologies → Neural networks**;

KEYWORDS

Sentiment Classification; Aspect; Hierarchical Attention Network

CIKM'17 , November 6–10, 2017, Singapore,Singapore
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4918-5/17/11…$15.00
https://doi.org/10.1145/3132847.3133037

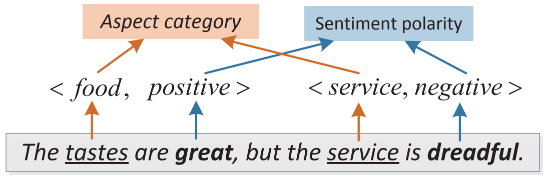


Figure 1: An example of aspect-level sentiment classification. The underline words are aspect terms and the bold words are sentiment words.

1 INTRODUCTION

Sentiment classification of user-generated reviews is an important technique to comprehend individuals' attitudes on a product or service [17]. With the popularity of online review sites such as Amazon and Yelp and online social media sites such as Twitter and Weibo, the reviews have been sharply increasing and become the most important resource for investigating a product or service in the way of crowdsourcing. For example, by collecting the tweets reviewing *iPhone 7s* and analyzing the sentiment, we can infer whether the product deserves to buy, and know whether the product will succeed and promote Apple's stock as well. In particular, product reviews often contain users' comments on different aspects of products. Hence, mining the sentiment of reviews at aspect-level provides opportunities to know the detailed feedback from customers. For instance, we can know much feedback about a restaurant from the reviews—How is the food? Is the service good? Is the price acceptable?

Aspect-level sentiment classification aims to predict the sentiment polarity of a text corresponding to a given aspect. Figure 1 shows an example, "*The tastes are great, but the service is dreadful.*" Given aspect *food* the sentiment polarity is positive, while, given aspect *service* the sentiment polarity is negative. Through extracting users' feedback on specific aspects, aspect-level sentiment classification helps businesses to discover the specific flaws of products and improve in further design, and also reports detailed information to customers with different preferences for reference to purchase.

Allocating appropriate sentiment words for a given aspect is the key for aspect-level sentiment classification. For the example in Figure 1, the sentiment feature should be "*great*" given aspect *food*, whereas, the sentiment feature should be "*dreadful*" given aspect *service*. However, it is difficult for traditional sentiment classification methods [2, 13, 40] to extract different features from the same text for different aspects. Recently, attention-based neural networks have been proposed to discover the related sentiment words for a given aspect, which turn out to be successful to allocate appropriate sentiment words for a given aspect and achieve the state-of-the-art performance for aspect-level sentiment classification [32].

However, the prior work only attends to the sentiment information and ignores the aspect-related information in the text, which may cause mismatching between the sentiment words and the aspects when an unrelated sentiment word is semantically meaningful for the given aspect. For the example in Figure 1, both "*great*" and "*dreadful*" are general sentiment words that can be used for both aspect *food* and aspect *service*. Given aspect *food*, the model may attend to both "*great*" and "*dreadful*", making it confusing to predict the sentiment. A simple way to solve this problem is to leverage the aspect terms to bridge the gap. Given aspect *food*, it is much easier to find aspect term "*tastes*" than to discriminate which sentiment word is corresponding to the aspect. Under the guidance of aspect term "*tastes*", we can easily choose the sentiment word "*great*" and decide the sentiment polarity on the aspect.

Motivated by the above intuition, we propose a **hie**rarchical **at**tention (HEAT) network to improve aspect-level sentiment classification by extracting aspect expressions. The HEAT network jointly learns to model the aspect information and the aspect-specific sentiment information from the text through a hierarchical attention module. The hierarchical attention module first attends to the aspect information under the direction of the given aspect and then pays attention to the sentiment information under the direction of the given aspect and the extracted aspect information of the text. Take the sentence in Figure 1 as an example, given aspect *food*, the hierarchical attention module first pays attention to the word "*tastes*" under the direction of *food* and then finds the word "*great*" with the information of aspect *food* and the word "*tastes*". In such a process, the hierarchical attention mechanism better locates the aspect-specific sentiment expressions of a text with the guidance of the aspect terms and thus improves the performance of aspect-level sentiment classification. Specifically, we introduce a location mask layer to represent the location information of aspect terms and sentiment expressions, which improves the attention calculation of sentiment expressions in the HEAT network. Moreover, the HEAT network supports to extract the aspect terms together with aspect-level sentiment classification by introducing the Bernoulli attention mechanism [11]. Finally, we evaluate our approach on four benchmark data sets at the sentence level and three data sets at the review level. The experimental results show that the HEAT model performs better than state-of-the-art methods for aspect-level sentiment classification.

In summary, the main contributions of our work are as follows:

- We propose the HEAT network for aspect-level sentiment classification. The model captures the **aspect information** of a text to help capture the aspect-specific **sentiment information** of the sentence, which improves the accuracy of aspect-level sentiment classification.
- The proposed model can extract aspect terms together with aspect-level sentiment classification with a Bernoulli (sigmoid) attention. Compared with the standard attention model, the new model with a Bernoulli attention can well control the instances of multiple aspect words and implicit aspect expressions.
- Experimental results show that our model can improve the performance of aspect-level sentiment classification compared with the state-of-the-art baselines. Furthermore, the case studies demonstrate that our model can capture the aspect information and sentiment information of a text very well.

2 RELATED WORK

In this section, we first present a brief review about aspect-level sentiment classification. Then, we show the related studies on attention network that is the key technology in our method.

2.1 Aspect-level Sentiment Classification

Aspect-level sentiment classification is a fine-grained sentiment classification task that needs to consider both the aspect information and the sentiment information of texts. Studies on aspect-level sentiment classification can be categorized into three different lines.

The first extracts the aspects and the sentiment of a sentence separately and associates them later. The aspects of a sentence are often extracted with linguistic patterns [25, 36], supervised sequence labeling [3, 38] or classification algorithms [35]. The sentiment of a sentence is typically classified with general sentiment classification methods, such as rule-based methods[2], feature-based classifiers [13, 34] or neural networks [27, 29, 40]. However, these methods assign only one sentiment polarity for a sentence thus cannot yield correct results for cases that the sentence expresses different opinions on different aspects, such as the example in Figure 1.

The second is target-dependent sentiment classification, which aims to infer the sentiment polarity of a sentence in response to a given target word mentioned in the sentence. Target-dependent sentiment classification is typically formulated as a text classification problem as well by adding some target-specific features [9, 12] or designing some specific neural network structures [21, 30] to consider the target word. However, target-dependent sentiment classification cannot deal with implicit aspect expressions and it is also limited for not grouping the target words into aspect categories.

The third is the recent trend for aspect-level sentiment classification, which exploits attention neural networks to predict sentiment polarity of a sentence given an aspect. In particular, Wang et al. [32] propose an attention-based LSTM to predict sentiment polarity of a sentence for a given aspect category and achieve the state-of-the-art performance. The attention model for aspect-level sentiment classification, which covers both implicit and explicit aspect expressions and groups the sentiment into aspect categories automatically, overcomes the drawbacks of the two lines as mentioned above. However, the model in [32] directly pays attention to the aspect-specific sentiment information with an attention layer, which may cause mismatching of sentiment words and aspects when an unrelated sentiment word is semantically meaningful for the target aspect.

Therefore, we propose a hierarchical attention network to solve this problem by extracting the aspect terms and using the aspect terms to help capture the aspect-specific sentiment information.

2.2 Attention Network

Attention network is a well-designed neural network which can selectively pay attention to specific inputs or features under the direction of a query. Attention networks have been used in a variety of natural language processing (NLP) tasks, such as machine translation [1, 18, 19], question answering [6, 37, 39], text summarization [20, 26] and sentiment analysis [30–32]. In particular, memory networks with multiple layer attentions have dominated many NLP tasks such as question answering [14, 28]. In addition, Lin et al. [16] employ self-attention to model sentences and find the important parts of sentences. Kim et al. [11] propose a structured attention network to select latent structure information of texts and achieve good results in machine translation, question answering, and language inference. Most attention networks are trained in an end-to-end manner[1, 28, 41], in which the attention weights of inputs are inferred under the direction of the final task without supervision. However, supervised attention can achieve better performance and learn a much clearer clue [19, 33]. Therefore, whether to use supervision on attention weights is a trade-off that depends on the task and the data set. In this paper, we attempt to take advantage of attention mechanism to model the aspect information and sentiment information, and thus to improve the performance of aspect-level sentiment classification. As for the way of learning attention weights, we employ supervision only on aspect attention, and the sentiment attention is inferred from the final sentiment classification task.

3 PROBLEM STATEMENT

In the literature of aspect-level sentiment classification, the *aspect* is an abstract and hierarchical concept with several different definitions [17, 22, 24, 30]. In this paper, we follow the the most popular definition given by [17], which is also used in [24, 32, 35]. In the following, we give the definitions of *aspect* and *aspect term* to help understand the HEAT model.

Definition 3.1 (Aspect). An aspect, also called an aspect category, of a product, is a category of similar parts or attributes of the product.

Take the laptop review as an example, *screen* and *battery* are parts of laptops. *Price* and *usability* are attributes of laptops. They are all regarded as aspects of laptops.

Definition 3.2 (Aspect term). An aspect term, also called explicit aspect expression, is a word or phrase that appears in the text explicitly indicating an aspect category.

An aspect can appear as different aspect terms in different texts. For example, aspect *screen* can be mentioned in texts with "*screen*", "*display*", and "*resolution*". Besides, aspect can be mentioned implicitly without any aspect term in texts. For example, "*It is cheap, I will choose here next time.*" is a sentence from a restaurant review. It has positive sentiment on aspect *price* and aspect *general*. Aspect *price* is indicated with sentiment word "*cheap*" and aspect *general* is inferred with the expression "*I will choose here next time*".

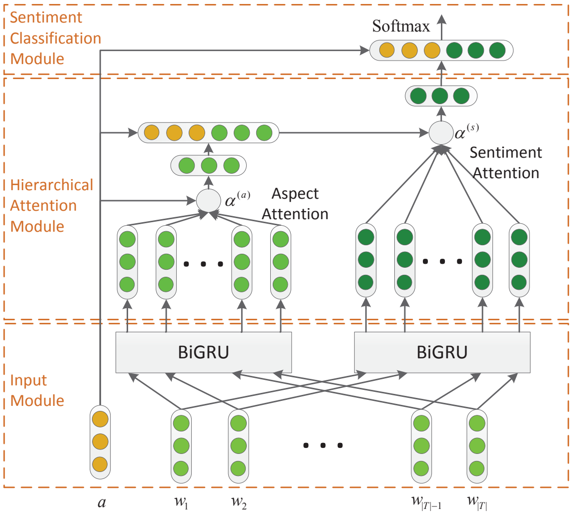


Figure 2: An illustration of the HEAT network for aspect-level sentiment classification.

The goal of aspect-level sentiment classification is to predict the sentiment polarity of a sentence or paragraph corresponding to a given aspect. With the above definitions, aspect-level sentiment classification can summarize the opinions into categories automatically and cover both explicit and implicit aspect expressions.

4 MODEL

This section presents our HEAT network architecture. First, we give an overview of the modules in the HEAT network. Then, we display the details of each module and introduce the training objective function. Finally, we show how to extract the aspect terms with the Bernoulli attention mechanism. In our description below, we denote vectors with bold small letters and matrices with bold capital letters.

4.1 HEAT Network Architecture

Figure 2 demonstrates the HEAT network architecture, which contains three modules: the input module, the hierarchical attention module, and the sentiment classification module.

Input Module. The input module encodes the text and the target aspect into distributed vector representations and extracts the contextual features of each word in the text.

Hierarchical Attention Module. The hierarchical attention module captures the aspect information and aspect-specific sentiment information with two attention layers, namely *aspect attention* and *sentiment attention*. The aspect attention aims to pay attention to the aspect information, i.e., aspect terms, under the direction of the target aspect. The sentiment attention aims to capture the sentiment feature of the text under the direction of the target aspect and the extracted aspect information.

Sentiment Classification Module. The sentiment classification module predicts the sentiment polarity of the text on the target aspect with the sentiment feature and the given aspect.

4.2 Input Module

The input of aspect-level sentiment classification contains a target aspect and a text. For the target aspect, we use aspect embeddings to represent all aspects and learn the representations end-to-end. For the text, we use recurrent neural network (RNN) to extract the contextual information of each word as the input of the text for the hierarchical attention module.

Aspect embedding. We use an embedding matrix $\mathbf{\Phi} \in \mathbb{R}^{d \times |\mathcal{A}|}$ to represent all the aspects, where d is the dimension of aspect embedding and $|\mathcal{A}|$ is the number of aspects in the data set. Further, we use vector $\boldsymbol{\phi}_a \in \mathbb{R}^d$ to represent the embedding of aspect a.

Text embedding with transfer knowledge. Text embeddings are the input of the recurrent network to learn the text representations. Because corpus domain and corpus size are significant in word embedding training [15], we exploit word embeddings with transfer knowledge for the text embedding. Namely, we first train word embeddings on an external large domain-specific corpus. Then, we use the trained embeddings as the input of the recurrent network for the supervised learning. Formally, the input text is a sequence of $|T|$ words $w_1, ..., w_i, ..., w_{|T|}$. We first train word embeddings $\mathbf{E} \in \mathbb{R}^{d_w \times |\mathcal{V}|}$ on an external large domain-specific corpus, where d_w is the dimension of word embedding and $|\mathcal{V}|$ is the size of vocabulary. Then, we use the word embedding $\mathbf{E}$ to project the text into a sequence of vectors, noted as $\mathbf{X} = [\mathbf{e}_1, \mathbf{e}_2, ..., \mathbf{e}_i, ..., \mathbf{e}_{|T|}]$, where $\mathbf{e}_i \in \mathbb{R}^{d_w}$ is a vector which represents w_i, as the input of RNN.

Choice of the recurrent network. Long short-term memory (LSTM) [8] and gated recurrent unit (GRU) [4] are two excellent RNN units which use some gates to overcome the vanishing gradient problem. Previous work [5, 10, 14] points out that LSTM and GRU perform similarly in many tasks and GRU is computationally cheaper than LSTM. Therefore, we choose GRU as the recurrent unit. Besides, the output of RNN at a step only depends on the current and the former steps, ignoring the latter ones. However, the contextual feature of a word in a text is often related to the whole text instead of only the words before it. Therefore, we use the bidirectional GRU (BiGRU) to extract the contextual feature of each word in a text. The internal mechanic of a GRU is defined as follows,

$$\mathbf{z}_i = \sigma(\mathbf{W}_z \mathbf{x}_i + \mathbf{U}_z \mathbf{h}_{i-1} + \mathbf{b}_z), \tag{1}$$

$$\mathbf{r}_i = \sigma(\mathbf{W}_r \mathbf{x}_i + \mathbf{U}_r \mathbf{h}_{i-1} + \mathbf{b}_r), \tag{2}$$

$$\tilde{\mathbf{h}}_i = \tanh(\mathbf{W}\mathbf{x}_i + \mathbf{r}_i \circ \mathbf{U}\mathbf{h}_{i-1} + \mathbf{b}_h), \tag{3}$$

$$\mathbf{h}_i = \mathbf{z}_i \circ \mathbf{h}_{i-1} + (\mathbf{1} - \mathbf{z}_i) \circ \tilde{\mathbf{h}}_i, \tag{4}$$

where $\circ$ denotes element-wise product, $\mathbf{W}_z, \mathbf{W}_r, \mathbf{W} \in \mathbb{R}^{d \times d_w}$, $\mathbf{U}_z, \mathbf{U}_r, \mathbf{U} \in \mathbb{R}^{d \times d}$ and $\mathbf{b}_z, \mathbf{b}_r, \mathbf{b}_h \in \mathbb{R}^d$ are parameters, σ is the sigmoid function, tanh is the hyperbolic tangent function. We abbreviate the computation of the forward GRU with $\overrightarrow{\mathbf{h}}_i = \overrightarrow{\mathrm{GRU}}(\mathbf{x}_i, \overrightarrow{\mathbf{h}}_{i-1})$ and the backward GRU with $\overleftarrow{\mathbf{h}}_i = \overleftarrow{\mathrm{GRU}}(\mathbf{x}_i, \overleftarrow{\mathbf{h}}_{i+1})$. We concatenate $\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ as the output of BiGRU at step i,

$$\overleftrightarrow{\mathbf{h}}_i = [\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i]. \tag{5}$$

Let $\overleftrightarrow{\mathbf{H}} = [\overleftrightarrow{\mathbf{h}}_1, \overleftrightarrow{\mathbf{h}}_2, ..., \overleftrightarrow{\mathbf{h}}_{|T|}] \in \mathbb{R}^{2d \times |T|}$, we abbreviate the computation of a BiGRU layer with $\overleftrightarrow{\mathbf{H}} = \mathrm{BiGRU}(\mathbf{X})$.

4.3 Hierarchical Attention Module

The hierarchical attention module models the aspect information of the text at first and then uses the aspect information to help capture the sentiment information of the text.

Aspect attention. Aspect attention finds the possible aspect terms and represents the aspect information of the text in response to the target aspect. The input of aspect attention is the contextual feature of each word from the BiGRU of the input module,

$$\overleftrightarrow{\mathbf{H}}^{(a)} = \mathrm{BiGRU}^{(a)}(\mathbf{X}), \tag{6}$$

where $\overleftrightarrow{\mathbf{H}}^{(a)} = [\overleftrightarrow{\mathbf{h}}^{(a)}_1, \overleftrightarrow{\mathbf{h}}^{(a)}_2, ..., \overleftrightarrow{\mathbf{h}}^{(a)}_{|T|}] \in \mathbb{R}^{2d \times |T|}$, column $\overleftrightarrow{\mathbf{h}}^{(a)}_i$ represents the aspect feature of word w_i. The attention mechanism calculates the weight of each word based on the target aspect embedding and the contextual features of the words,

$$g^{(a)}_i = (\mathbf{u}^{(a)})^T \tanh(\mathbf{W}^{(a)}[\boldsymbol{\phi}_a; \overleftrightarrow{\mathbf{h}}^{(a)}_i] + \mathbf{b}^{(a)}), \tag{7}$$

$$\alpha^{(a)}_i = \frac{\exp(g^{(a)}_i)}{\sum_{j=1}^{|T|} \exp(g^{(a)}_j)}, \tag{8}$$

where $\mathbf{W}^{(a)} \in \mathbb{R}^{d \times 3d}$ is a weight matrix, $\mathbf{b}^{(a)} \in \mathbb{R}^d$ is a bias vector, $\mathbf{u}^{(a)} \in \mathbb{R}^d$ is a weight vector and $(\mathbf{u}^{(a)})^T$ is the transpose of $\mathbf{u}^{(a)}$. $\alpha^{(a)}_i$ is the weight of word w_i that contains the information of the target aspect in the text. Then the aspect information of the text on the target aspect is the weighted sum of the contextual features of all the words in the text,

$$\mathbf{v}^{(a)} = \sum_{i=1}^{|T|} \alpha^{(a)}_i \overleftrightarrow{\mathbf{h}}^{(a)}_i. \tag{9}$$

Sentiment attention. Sentiment attention extracts the sentiment feature of the text under the direction of the target aspect and the aspect information of the text extracted by aspect attention. Similar to aspect attention, the input of sentiment attention is the contextual feature of each word from the BiGRU of the input module,

$$\overleftrightarrow{\mathbf{H}}^{(s)} = \mathrm{BiGRU}^{(s)}(\mathbf{X}), \tag{10}$$

where $\overleftrightarrow{\mathbf{H}}^{(s)} = [\overleftrightarrow{\mathbf{h}}^{(s)}_1, \overleftrightarrow{\mathbf{h}}^{(s)}_2, ..., \overleftrightarrow{\mathbf{h}}^{(s)}_{|T|}] \in \mathbb{R}^{2d \times |T|}$, column $\overleftrightarrow{\mathbf{h}}^{(s)}_i$ represents the sentiment feature of word w_i. Because aspect information and sentiment information need different features, BiGRU$^{(a)}$ and BiGRU$^{(s)}$ should not share parameters.

We calculate the attention score for each word $g^{(s)}_i$ based on the concatenations of the target aspect embedding, the aspect feature of the text, and the contextual features of the words,

$$g^{(s)}_i = (\mathbf{u}^{(s)})^T \tanh(\mathbf{W}^{(s)}[\boldsymbol{\phi}_a; \mathbf{v}^{(a)}; \overleftrightarrow{\mathbf{h}}^{(s)}_i] + \mathbf{b}^{(s)}), \tag{11}$$

where $\mathbf{W}^{(s)} \in \mathbb{R}^{d \times 5d}$ is a weight matrix, $\mathbf{b}^{(s)} \in \mathbb{R}^d$ is a bias vector. $\mathbf{u}^{(s)} \in \mathbb{R}^d$ is a weight vector. $g^{(s)}_i$ captures semantic relation of word w_i with the aspect-specific sentiment of the text.

To better calculate the attention weights, we further consider the location information of aspect terms. The location information is helpful for sentiment attention because a sentiment expression closer to the aspect term is often more important than a further one [30]. Therefore, we use a location mask layer to represent the location information of aspect terms and sentiment expressions. Since we do not have the exact locations of aspect terms, we use the aspect attention weight vector $\boldsymbol{\alpha}^{(a)}$ to calculate the location mask. Specifically, we define a location matrix $\mathbf{M} \in R^{|T| \times |T|}$ to represent the proximity of each word in a text,

$$M_{ij} = 1 - \frac{|i-j|}{|T|}, \tag{12}$$

where $|T|$ is the length of the text and $i, j \in \{1, 2, ..., |T|\}$. Then we calculate the location mask $\mathbf{m}^{(l)}$ according to the aspect attention weight:

$$\mathbf{m}^{(l)} = \mathbf{M}\boldsymbol{\alpha}^{(a)}. \tag{13}$$

Now a word closer to aspect terms will have a larger value in $\mathbf{m}^{(\mathbf{l})}$ because aspect terms will have larger values in $\boldsymbol{\alpha}^{(\boldsymbol{a})}$. Finally, the sentiment attention scores are calculated as follows,

$$\alpha_i^{(s)} = \frac{\exp(m_i^{(l)} g_i^{(s)})}{\sum_{j=1}^{|T|} \exp(m_i^{(l)} g_j^{(s)})}. \tag{14}$$

The sentiment feature of the text on the target aspect is the weighted sum of the context features of all the words in the text with weight vector $\boldsymbol{\alpha}^{(\boldsymbol{s})}$,

$$\mathbf{v}^{(s)} = \sum_{i=1}^{|T|} \alpha_i^{(s)} \overleftrightarrow{\mathbf{h}}_i^{(s)}. \tag{15}$$

4.4 Sentiment Classification Module

The sentiment classification module predicts the sentiment of the text on the target aspect after the hierarchical attention module. Considering that some sentiment words may have different sentiment polarities for different aspects, we concatenate the aspect vector $\boldsymbol{\phi_a}$ and sentiment feature $\mathbf{v}^{(\mathbf{s})}$ as the neural input to predict the sentiment polarity of the text on the target aspect,

$$\mathbf{y} = \mathrm{softmax}(\mathbf{W}_y[\boldsymbol{\phi}_a; \mathbf{v}^{(s)}] + \mathbf{b}_y), \tag{16}$$

where $\mathbf{W}_y \in \mathbb{R}^{d_c \times 3d}$ is a weight matrix, $\mathbf{b}_y \in \mathbb{R}^{d_c}$ is a bias vector, d_c is the number of sentiment classes,

$$\mathrm{softmax}(\mathbf{x})_i = \frac{\exp(x_i)}{\sum_{j=1}^{d_c} \exp(x_j)}. \tag{17}$$

4.5 Objective Function

In order to ensure that the aspect attention and the sentiment attention extract the aspect information and the sentiment information, respectively, we use supervision in both the aspect attention and the final sentiment classification.

For aspect attention, the training set provides aspect terms of each text on the target aspect. We use an aspect term mask vector $\mathbf{m} \in \mathbb{R}^{|T|}$ to mark the aspect terms, where $m_i = 1$ when word w_i is an aspect term of the target aspect, and $m_i = 0$ otherwise. Then the loss of aspect attention is

$$\mathcal{L}^{(a)} = -\frac{1}{\sum_{i=1}^{|T|} m_i} \sum_{i=1}^{|T|} m_i \log(\alpha_i^{(a)}). \tag{18}$$

For sentiment classification, $\hat{\mathbf{y}}$ is the labeled sentiment distribution and $\mathbf{y}$ is the predicted sentiment distribution. We use cross entropy loss between $\hat{\mathbf{y}}$ and $\mathbf{y}$ as the loss of classification,

$$\mathcal{L}^{(s)} = -\sum_{k=1}^{d_c} (\hat{y}_k \log(y_k) + (1 - \hat{y}_k) \log(1 - y_k)). \tag{19}$$

At last, the objective function is

$$\mathcal{L} = \lambda \mathcal{L}^{(a)} + (1 - \lambda) \mathcal{L}^{(s)} + \lambda^{'} ||\theta||^2, \tag{20}$$

where $\lambda \in (0, 1)$ is a hyper parameter used to balance the aspect attention loss and the sentiment classification loss, $\lambda^{'}$ is the L_2-regularization term, and θ is the parameter set.

4.6 Aspect Term Extraction with Bernoulli Attention

Aspect terms are significant information for aspect-level sentiment analysis. For example, in restaurant reviews, *food* is an aspect. "Taste", "hamburger" and "pizza" are aspect terms of *food.* Apart from the whole sentiment to each aspect category, businesses and customers need more details such as which food (e.g., pizza or chicken) is good. Therefore, extracting aspect terms together with aspect-level sentiment classification is meaningful. However, the standard attention mechanism can not denote the definite aspect terms, especially when the text has multiple aspect terms or no aspect term for an aspect because the weights of the words are relative.

In the HEAT network, the aspect attention module can highlight the importance of each word in representing the target aspect. A word with higher weight in $\boldsymbol{\alpha}^{(a)}$ has a relatively higher probability of being an aspect term, which motivates us to extract the aspect terms together with the sentiment classification task. Moreover, we use Bernoulli (sigmoid) attention [11] to calculate $\boldsymbol{\alpha}^{(a)}$ for extracting aspect terms instead of the softmax attention (e.g., Equation (8)),

$$\alpha_i^{(a)} = \sigma(g_i^{(a)}). \tag{21}$$

In such a process, aspect attention becomes a sequence labeling model, in which $\alpha_i^{(a)}$ is regarded as the probability of word w_i to be an aspect term. We simply assume words with the weights higher than 0.5 to be aspect terms. Using the Bernoulli attention mechanism to calculate the weights, $\alpha_i^{(a)}$ can be zero for all terms in the text. According to Equation (13), the location mask $\mathbf{m}^{(l)}$ will be zero as well, which leads to the invalidation of sentiment attention in Equation (14). To avoid this problem, we add a small number ϵ to the location mask,

$$\alpha_i^{(s)} = \frac{\exp((m_i^{(l)} + \epsilon) g_i^{(s)})}{\sum_{j=1}^{|T|} \exp((m_i^{(l)} + \epsilon) g_j^{(s)})}. \tag{22}$$

After introducing the Bernoulli attention mechanism, the loss function of aspect attention is the cross entropy loss between $\boldsymbol{\alpha}^{(a)}$ and $\mathbf{m}$,

$$\mathcal{L}^{(a)} = -\frac{1}{|T|} \sum_{i=1}^{|T|} (m_i \log(\alpha_i^{(a)}) + (1 - m_i) \log(1 - \alpha_i^{(a)})). \tag{23}$$

Table 1: Data sets with number of reviews (R), sentences (S), and $\langle aspect, sentiment \rangle$ tuples (T).

Dataset	Training			Test			Total		
	#R	#S	#T	#R	#S	#T	#R	#S	#T
REST14	-	3,401	3,713	-	800	1,025	-	4,201	4,738
REST15	254	1,315	1,476	96	685	730	350	2,000	2,206
REST16	350	2,000	2,107	90	676	703	440	2,676	2,810
LAPT15	277	1,739	1,901	173	761	912	350	2,500	2,813

It is notable that we use different loss functions for standard aspect attention and Bernoulli aspect attention. For standard aspect attention, the loss function (Equation (18)) only concerns the weights of the labeled aspect terms. Because the sum of the weights is 1, higher weight for an aspect term means lower weights for other words. However, for Bernoulli aspect attention, the weights of the words have no direct relation, so we have to evaluate the losses of all the words (Equation (23)). The different loss functions have another advantage when processing implicit aspect expressions. When a text implicitly expresses sentiment to an aspect, there is no aspect term. The aspect attention layer may become noise for sentiment attention layer. However, the loss of standard aspect attention will be zero during the training process. Then the hierarchical attention model becomes a classification model with two attention layers. For Bernoulli aspect attention, the attention loss directs all the aspect attention weights to be zero. The sentiment attention layer uses only the aspect embedding and the contextual information of the words. In summary, the designed loss functions of the two different attention types can evaluate the loss of aspect attention well and avoid the disadvantage of the model in processing implicit aspect expressions.

5 EXPERIMENTS

5.1 Data Set

We evaluate our model with review data sets at the sentence level and the review level. Specifically, the review level sentiment classification aims to predict the sentiment of a meta review on a given aspect. The sentence level sentiment classification aims to predict the sentiment of each sentence delimited from the reviews by ending punctuations on a given aspect. The data sets are taken from SemEval Challenge 2014 task 4 [24], SemEval Challenge 2015 task 12 [23] and SemEval Challenge 2016 task 5 [22]. The details of each data set are displayed in Table 1. "REST" means restaurant domain and "LAPT" means laptop domain. REST15 and REST16 have labeled ⟨aspect, aspect term, sentiment polarity⟩ tuples, which is sufficient for our model. The original REST14 data set labels ⟨aspect term, sentiment polarity⟩ and ⟨aspect, sentiment polarity⟩ of each sentence separately. In order to compare our model to the state-of-the-art [32], we manually map each aspect term to the corresponding aspect category. The original LAPT15 data set only labels the ⟨aspect, sentiment polarity⟩ tuples of each sentence. In order to evaluate our model on electronic product reviews, which is much different from restaurant reviews, we manually label the aspect terms of each aspect for each sentence. To keep consistent with [32], we take the attributes and components of products as aspects, ignoring the aspect of the components in REST15, REST16, and LAPT15. Because REST14 does not have the review information of sentences, we use the other three data sets to evaluate our model at review level.

5.2 Experimental Setting

As mentioned in Section 4.2, we train word embeddings on two domain-specific corpora with much larger size rather than the labeled data sets. Following [31], we use the Yelp Challenge data set[1] for restaurant domain, which contains 4.1M reviews on different restaurants, and we use the Amazon electronic data set [7] for laptop domain, which contains more than 1.5M reviews. For both corpora, we train 100-dimension word embeddings with the word2vec tool.[2]

To make the experiments repeatable, we explain the parameter settings as follows. We implement our model with Tensorflow[3] and TensorLayer.[4] All the parameters except word embeddings are initialized with uniform distribution $\mathbf{U}(-0.1, 0.1)$. The dimension of aspect embedding and the size of hidden layer are 32. The weight of aspect attention loss λ is different in different models and data sets, and it is tuned with 5-fold cross-validation. The L_2 regularization weight λ' is set to be 0.001. The training is carried with stochastic gradient descent with the momentum of 0.9 and the batch size of 30. We use learning rate decay and early stop in the training process. Specifically, for each data set, 10% of training data is held out as a validation set. The initial learning rate is 0.001. We decay the learning rate by half when the classification loss on the validation set increases and stop training when the learning rate is lower than 0.00001. In order to alleviate the impact of different initialization of parameters, we run each model 5 times with different initial parameters on each data set and report the mean value as the result of the model on the data set.

5.3 Candidate Models for Comparison

We compare the performance of the following models.

AT-LSTM: Attention-based LSTM (AT-LSTM) [32] uses an LSTM layer to extract the contextual feature of each word and attends to the sentiment expressions under the direction of the given aspect. The aspect-specific sentiment information is extracted depending on the aspect embedding vector and the contextual feature.

[1] https://www.yelp.com/dataset_challenge

[2] https://radimrehurek.com/gensim/models/word2vec.html

[3] https://www.tensorflow.org

[4] http://tensorlayer.org

Table 2: Accuracy on aspect-level sentiment classification (sentence level).

Model	REST14		REST15		REST16		LAPT15	
	Bin.	Thr.	Bin.	Thr.	Bin.	Thr.	Bin.	Thr.
AT-LSTM	89.6	83.1	81.0	77.2	87.6	83.0	86.3	82.1
ATAE-LSTM	89.9	84.0	80.9	77.4	87.2	82.7	85.8	82.3
AT-BiGRU	90.4	84.3	82.8	79.2	90.4	86.7	87.0	84.3
HEAT-GRU	89.6	84.3	81.2	79.1	89.7	85.8	87.4	84.5
HEATB-GRU	89.4	84.0	81.8	79.6	89.2	85.4	87.3	84.2
HEAT-BiGRU	**91.3**	**85.1**	83.0	80.1	90.8	87.1	87.9	84.9
HEATB-BiGRU	91.1	84.9	**83.4**	**80.5**	**91.1**	**87.5**	**88.0**	**85.1**

Table 3: Accuracy on aspect-level sentiment classification (review level).

Model	REST15		REST16		LAPT15	
	Bin.	Thr.	Bin.	Thr.	Bin.	Thr.
AT-LSTM	78.3	75.0	83.2	82.1	79.6	74.6
ATAE-LSTM	78.8	76.4	83.6	81.8	79.3	74.2
AT-BiGRU	79.7	75.8	83.5	80.5	83.6	80.2
HEAT-GRU	83.9	81.0	86.5	82.3	83.8	80.4
HEATB-GRU	84.4	81.5	**86.6**	82.9	84.3	80.4
HEAT-BiGRU	84.2	81.7	86.3	**83.0**	84.9	81.7
HEATB-BiGRU	**84.8**	**82.1**	85.6	82.9	**85.5**	**82.4**

ATAE-LSTM: Attention-based LSTM with aspect embedding (ATAE-LSTM) [32] is an improvement of AT-LSTM by concatenating aspect embedding to word embedding in the LSTM layer. In such a process, the LSTM layer can take advantage of the aspect information for the contextual feature extraction, and thus can capture more important information in response to the aspect. ATAE-LSTM has achieved the state-of-the-art performance on REST14.

AT-BiGRU: Attention-based BiGRU (AT-BiGRU) replaces the LSTM layer in AT-LSTM with BiGRU, which contains a similar sentiment attention module with our model. We compare it to verify the improvement of adding aspect attention module.

HEAT-BiGRU: HEAT-BiGRU is our proposed HEAT network with standard attention in aspect attention module. The model captures the aspect information of a text and uses the aspect information to capture the aspect-specific sentiment information.

HEATB-BiGRU: HEATB-BiGRU is our proposed HEAT network with Bernoulli attention in aspect attention module. The model uses Bernoulli attention to label the aspect terms in the aspect attention layer.

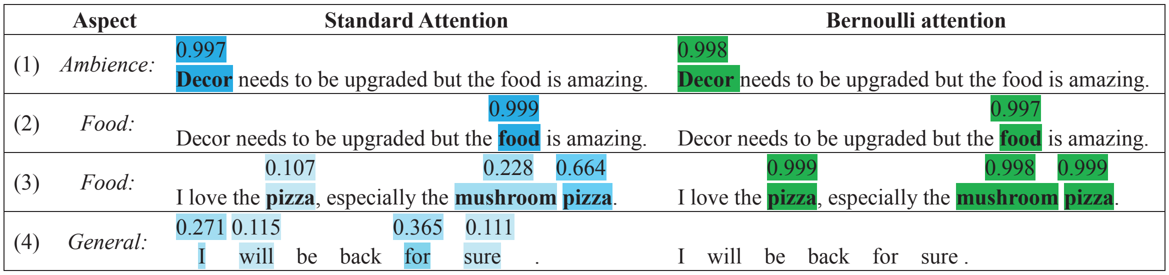
HEAT-GRU, HEATB-GRU: HEAT-GRU and HEATB-GRU are the unidirectional version of HEAT-BiGRU and HEATB-BiGRU, respectively, using GRU to extract contextual information as inputs for the hierarchical attention module. We compare with them to verify the choice of recurrent network.

5.4 Evaluation

Sentence level. Table 2 shows the experimental results of sentence-level evaluation, where "Bin." means binary classification (positive, negative) and "Thr." denotes 3-class prediction (positive, negative, and neutral). The best scores are in bold. We obtain the following observations. 1) Our proposed HEAT networks achieve better results than state-of-the-art attention models. In addition, all best scores are achieved by HEAT-BiGRU and HEATB-BiGRU, and most of them are achieved by HEATB-BiGRU. 2) All the bidirectional models achieve much higher accuracies than the corresponding unidirectional models. Bidirectional models can grasp both the former and the later contextual information of each word, and thus are powerful than the unidirectional models. 3) Most hierarchical attention networks obtain better results than the corresponding baseline models. Hierarchical models can extract the aspect information, which is helpful to capture the aspect-specific sentiment information of a sentence. Besides, standard aspect attention and Bernoulli aspect attention get similar prediction accuracies.

It is notable that our implementations of AT-LSTM and ATAE-LSTM on REST14 obtain lower results than [32]. Our accuracies of AT-LSTM and ATAE-LSTM are 87.9 and 87.8 for binary classification, 80.8 and 81.0 for 3-class classification. It might be caused by the lower quality of our embeddings. Wang et al. [32] train word embeddings on an external corpus with 840 billion texts, which is much larger than ours. Because we also evaluate on REST14, we report the results of [32] in Table 2: AT-LSTM and ATAE-LSTM are 89.6 and 89.9 for binary classification, 83.1 and 84.0 for 3-class classification.

Review level. Table 3 gives the review level evaluation results. Similar to sentence level results, our hierarchical attention neural networks achieve significantly better performance than the baseline models. However, there are some different observations from sentence level results. First, the BiGRU does not bring significant improvements like in the sentence level evaluation. For the binary classification task on REST16 data, the bidirectional models even perform a little worse than the corresponding unidirectional models. This phenomenon can be explained as follows: The sentiment of a review on an aspect mainly depends on the local contextual features. However, the review level texts usually contain several sentences, in which the bidirectional model increases the risk of containing much sentiment information from sentences unrelated to the given aspect. Second, the improvement of the hierarchical attention is more significant than sentence level evaluation. Review texts are more likely to contain the case where several sentiment words are semantically meaningful for the given aspect. This case is easier mismatched by prior attention models than our proposed



	Aspect	Standard Attention	Bernoulli attention
(1)	*Ambience:*	0.997 **Decor** needs to be upgraded but the food is amazing.	0.998 **Decor** needs to be upgraded but the food is amazing.
(2)	*Food:*	0.999 Decor needs to be upgraded but the **food** is amazing.	0.997 Decor needs to be upgraded but the **food** is amazing.
(3)	*Food:*	0.107 0.228 0.664 I love the **pizza**, especially the **mushroom** **pizza**.	0.999 0.998 0.999 I love the **pizza**, especially the **mushroom** **pizza**.
(4)	*General:*	0.271 0.115 0.365 0.111 I will be back for sure .	I will be back for sure .

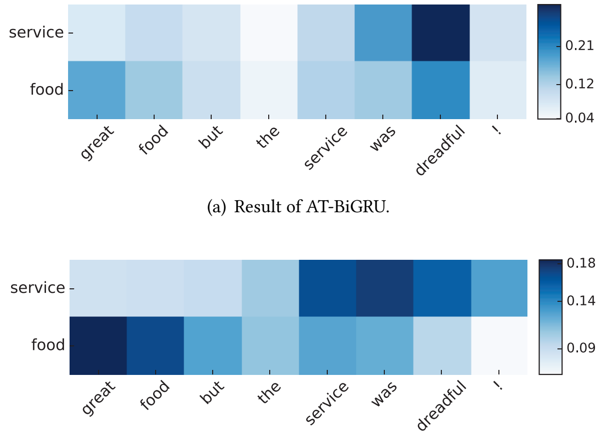
Figure 3: Examples of learned aspect attention scores. The numbers on the top of words are the aspect attention scores of the words. The scores lower than 0.1 are not labeled. The bold words are the labeled aspect terms. The color depth expresses the important degree of the word.

hierarchical neural models. Therefore, our proposed hierarchical neural models achieve more significant improvement on the review level evaluation than the sentence level, which also proves the effectiveness of the hierarchical attention mechanism.

Comparing the results at the review level and the sentence level, we find that the accuracies of REST16 and LAPT15 at the review level are obviously lower than the results at the sentence level. It is because that a review often mentions more aspects than a sentence, bringing more noisy information when predicting the sentiment polarity on the target aspect. However, the review level accuracies of REST15 are higher than the sentence level results. We check the test set of REST15 and find that it has more sentences which are beyond the control of our model (see Section 5.6) than other data sets. That is also why the sentence level results of REST15 are obviously worse than the results of REST14 and REST16. However, review level sentiment classification can alleviate this problem because a review often contains more information about an aspect than a sentence. For example, sentence "*Eerything was going good until we got our meals.*" expresses negative sentiment to aspect *food* but our model fails to predict it. The review of this sentence is "*Everything was going good until we got our meals. I took one look at the chicken and I was appalled. It was served with skin, over a bed of extremely undercooked spinach and mashed potatoes.*", which contains more information about aspect *food* and the model can predict the sentiment correctly. Therefore, the review level classification results of REST15 are better than the sentence-level results and are comparable to the review-level classification results of REST16.

5.5 Visualization of Attentions

Aspect attention. Figure 3 displays the performance of both standard attention and Bernoulli attention to find the aspect terms on four examples. Example (1) and (2) are the same sentence in response to different aspects. Both attention mechanisms can find the aspect terms obviously. Example (3) is a sentence with more than one aspect terms for the target aspect. We can find that both attention mechanisms get obviously higher scores for the labeled aspect terms than the other words. However, the scores of the first "*pizza*" and the "*mushroom*" with standard attention are quite small because the sum of the scores is limited to 1. Example (4) is a sentence without any aspect term. All the scores with Bernoulli attention mechanism are lower than 0.1, correctly indicating that no aspect term is labeled. However, the standard attention mechanism is forced to attend to some words. From the last two sentences, we can find that standard aspect attention will get confused when there are more than one aspect words or no aspect term exists, whereas Bernoulli aspect attention resolves this problem quite well.

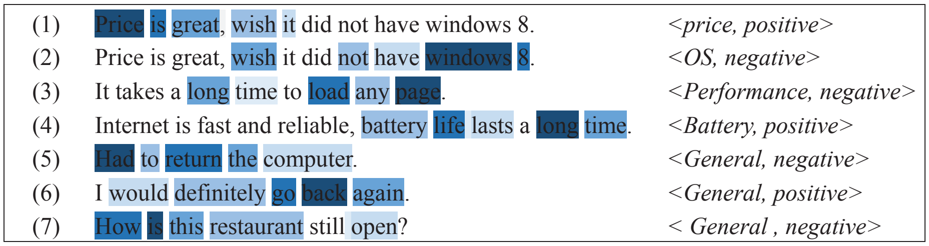


(a) Result of AT-BiGRU.

(b) Result of HEATB-BiGRU.

Figure 4: Sentiment attention results of HEATB-BiGRU and AT-BiGRU for a sentence with different sentiment on two aspects. The color depth expresses the weight in sentiment attention vector $\alpha^{(s)}$. The words on the left show the target aspect. The two sub-figures show the effect of our hierarchical attention mechanism.

Sentiment attention. We compare the sentiment attention results of our model and the baseline on a sentence with different aspects to see the effect of the hierarchical attention. We choose HEATB-BiGRU and AT-BiGRU for illustration because HEATB-BiGRU mostly achieves the best in our proposed models and AT-BiGRU is the strongest competitor among all baselines. Figure 4 shows the sentiment attention results of HEATB-BiGRU and AT-BiGRU for a sentence with two aspects. Comparing Figure 4(a) and Figure 4(b), we observe that HEAT-BiGRU performs much better than AT-BiGRU in the case that a sentence contains multiple sentiment words semantically meaningful for more than one aspect. In



(1)	Price is great, wish it did not have windows 8.	<price, positive>
(2)	Price is great, wish it did not have windows 8.	<OS, negative>
(3)	It takes a long time to load any page.	<Performance, negative>
(4)	Internet is fast and reliable, battery life lasts a long time.	<Battery, positive>
(5)	Had to return the computer.	<General, negative>
(6)	I would definitely go back again.	<General, positive>
(7)	How is this restaurant still open?	< General , negative>

Figure 5: Examples of sentiment attention and classification results. The words with top-5 scores are colored and the color depth expresses the weight of the word in sentiment attention.

other words, AT-BiGRU is easier to mismatch unrelated sentiment words to an aspect than HEAT-BiGRU when the irrelevant sentiment words are semantically meaningful for the given aspect. In particular, we find that AT-BiGRU gets confused to locate sentiment word for aspect *food* in Figure 4(a). Given aspect *food*, both "*great*" and "*dreadful*" obtain high scores. On the contrary, in Figure 4(b) HEATB-BiGRU solves the problem well—the expression "*service was dreadful!*" gets higher scores than other words given aspect *service* and the expression "*great food*" achieves the top scores given aspect *food*. Compared with AT-BiGRU, our HEATB-BiGRU attends to aspect term "*food*" and leverage it to find the corresponding sentiment word "*great*" correctly through the hierarchical attention mechanism.

Besides, from Figure 4, we find that the sentiment attention does not strictly focus on the sentiment words. When given aspect *service*, the scores of "*service was dreadful!*" are comparably high. We consider that it is because some sentiment words express different sentiment in different contexts, even in the same aspect. For example, word "*fast*" expresses positive sentiment in sentence "*The battery charges fast*" while negative sentiment in sentence "*The power goes fast*". Therefore, attending to the whole sentiment expression keeps more contextual information for sentiment classification.

5.6 Quality Analysis

As demonstrated above, our model obtains the state-of-the-art performance on aspect-level sentiment classification. In this subsection, we show the advantages of our model and analyze where the error lies in through some typical examples.

Case studies. In Figure 5, we list some examples of sentiment classification results from the test sets. Example (1)-(5) are laptop reviews and (6)-(7) are restaurant reviews. In example (1) and (2), sentence "Price is great, wish it did not have windows 8." mentions two aspects: *price* and *operation system (OS)*. The results show that our model can discriminate different sentiment polarities for different aspects. In example (3) and (4), the two sentences have the same sentiment term "*long time*" with different sentiment polarities. "*Long time*" is an advantage for battery life while a disadvantage for loading Internet pages. Our model has attended to the term "*long time*" in both sentences, also accurately discriminates different sentiment polarities. The results demonstrate that our model can capture the contextual information well and discriminate different meanings of words with different contexts. Example (5) and (6) are two sentences which express sentiment without any sentiment word. As the two sentences are simple and some similar sentences often appear in reviews, our model can learn the regularization and predict the sentiment polarities of them correctly. Example (7) is a sentence with a rhetorical question, which is difficult to predict the sentiment because the sentiment is inverted by the question tone. Our model gives the highest scores to "*how is*" and correctly predicts the sentiment polarity, demonstrating that our model has a good ability to capture semantic meanings of texts.

Error analysis. We carry out an error analysis of our model and find some similar cases where our model often goes wrong. For sentiment classification, three kinds of sentences often confuse our model. The first is the sentences with comparative sentiment. An example is "*Frankly, the Chinese food here is something I can make better at home*". The model attends to word "*better*" but cannot discriminate whether it is positive for the target in the review. The second is some obscure expressions of sentiment. An example is "*Save your money and your time and go somewhere else*". The sentence is a restaurant review and expresses negative sentiment to the restaurant. Similar cases often appear in comments on aspect *service* of restaurants and aspect *support* of laptops. These reviews express opinions with no sentiment word and instead with implicit expressions such as a story, a piece of advice, or a plan. Our model can handle some simple and frequently appeared ones such as example (5) and (6) in Figure 5. But we cannot handle some complex ones because they rarely appear and need reasoning the meanings. The third is conditional sentences, such as "*Would be a very nice laptop if the mouse pad worked properly*". The sentence expresses negative sentiment to aspect *mouse*. This kind of review frequently appears in laptop reviews to express the wish of users. Our model can attend to "*worked properly*" but fails to capture the conditional indicator "*if*", and thus gets an incorrect prediction. For aspect attention, there are two main problems. The first is unknown words. Some words, especially the names of some foods in restaurant reviews, are rarely used and have not appeared in the pre-train corpus. We have no embedding of them and can not attend to them in the aspect attention layer. The second is complex aspect expressions such as "*Chinese style Indian food*" in sentence "*Not a very fancy place but very good Chinese style Indian food*" and "*selection of bottled beer*" in sentence "*Not much of a selection of bottled beer*". The model can only label out "*Indian food*" and "*beer*" respectively, failing to get the qualifiers.

6 CONCLUSIONS AND FUTURE WORK

In this paper, we propose the HEAT network for aspect-level sentiment classification. The key idea of the model is to learn the aspect information of a sentence to help capture the sentiment of the sentence, aiming to improve the performance of aspect-level sentiment classification. Further, we improve the attention mechanism of aspect attention to extract the aspect terms together with aspect-level sentiment classification. Experimental results on sentiment-level and review-level data sets show that our model obtains better performance over the baselines.

Though our proposed model improves the aspect-level sentiment classification and achieves better performance than state-of-the-art methods, there are some special cases our model cannot handle, such as the comparative sentiment (analyzed in Section 5.6). Those special cases shown in Section 5.6 are still unresolved in the area of aspect-level sentiment classification. In the future, we will attempt to design specific neural structures to handle these special cases.

ACKNOWLEDGMENTS

The work described in this paper was partially supported by the National Natural Science Foundation of China (No. 71331008) and the Research Grants Council of the Hong Kong Special Administrative Region, China (No. CUHK 14208815 of the General Research Fund)

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural Machine Translation by Jointly Learning to Align and Translate. *Computer Science* (2014).

[2] Danushka Bollegala, David Weir, and John Carroll. 2013. Cross-domain sentiment classification using a sentiment sensitive thesaurus. *Knowledge and Data Engineering, IEEE Transactions on* 25, 8 (2013), 1719–1731.

[3] Jiajun Cheng, Xin Zhang, Pei Li, Sheng Zhang, Zhaoyun Ding, and Hui Wang. 2016. Exploring sentiment parsing of microblogging texts for opinion polling on chinese public figures. *Applied Intelligence* (2016), 1–14.

[4] Kyunghyun Cho, Bart Van MerriÃńnboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* (2014).

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555* (2014).

[6] David Golub and Xiaodong He. 2016. Character-level question answering with attention. *arXiv preprint arXiv:1604.00727* (2016).

[7] Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *Proceedings of the 25th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 507–517.

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

[9] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. 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, 151–160.

[10] Rafal Jozefowicz, Wojciech Zaremba, and Ilya Sutskever. 2015. An Empirical Exploration of Recurrent Network Architectures. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*. 2342–2350.

[11] Yoon Kim, Carl Denton, Luong Hoang, and Alexander M Rush. 2017. Structured attention networks. *arXiv preprint arXiv:1702.00887* (2017).

[12] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval 2014*. 437–442.

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

[14] Ankit Kumar, Ozan Irsoy, Peter Ondruska, Mohit Iyyer, James Bradbury, Ishaan Gulrajani, Victor Zhong, Romain Paulus, and Richard Socher. 2016. Ask Me Anything: Dynamic Memory Networks for Natural Language Processing. In *Proceedings of The 33rd International Conference on Machine Learning*. 1378–1387.

[15] Siwei Lai, Kang Liu, Shizhu He, and Jun Zhao. 2016. How to generate a good word embedding. *IEEE Intelligent Systems* 31, 6 (2016), 5–14.

[16] Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A Structured Self-attentive Sentence Embedding. *arXiv preprint arXiv:1703.03130* (2017).

[17] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (2012), 1–167.

[18] Minh Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective Approaches to Attention-based Neural Machine Translation. *Computer Science* (2015).

[19] Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. *arXiv preprint arXiv:1608.00112* (2016).

[20] P. Nema, M. Khapra, A. Laha, and B. Ravindran. 2017. Diversity driven Attention Model for Query-based Abstractive Summarization. *ArXiv e-prints arXiv:1704.08300* (April 2017).

[21] Thien Hai Nguyen and Kiyoaki Shirai. PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis. In *EMNLP*. 2509–2514.

[22] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, and Orphee De Clercq. SemEval-2016 Task 5: Aspect Based Sentiment Analysis. In *International Workshop on Semantic Evaluation*.

[23] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. (2015).

[24] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. *Proceedings of International Workshop on Semantic Evaluation at* (2014), 27–35.

[25] Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational linguistics* 37, 1 (2011), 9–27.

[26] Alexander M. Rush, Sumit Chopra, and Jason Weston. 2015. A Neural Attention Model for Abstractive Sentence Summarization. *Computer Science* (2015).

[27] 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 *EMNLP*, Vol. 1631. Citeseer, 1642.

[28] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-To-End Memory Networks. *Computer Science* (2015).

[29] 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*. 1422–1432.

[30] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900* (2016).

[31] Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. Coupled Multi-layer Attentions for Co-extraction of Aspect and Opinion Terms. In *31st AAAI Conference on Artificial Intelligence (AAAI-17)*. AAAI Press.

[32] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *Conference on Empirical Methods in Natural Language Processing*. 606–615.

[33] Jason Weston, Sumit Chopra, and Antoine Bordes. 2014. Memory networks. *arXiv preprint arXiv:1410.3916* (2014).

[34] Haibing Wu and Xiaodong Gu. 2014. Reducing Over-Weighting in Supervised Term Weighting for Sentiment Analysis.. In *COLING*. 1322–1330.

[35] Haibing Wu, Yiwei Gu, Shangdi Sun, and Xiaodong Gu. 2016. Aspect-based Opinion Summarization with Convolutional Neural Networks. In *2016 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 3157–3163.

[36] Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*. Association for Computational Linguistics, 1533–1541.

[37] Caiming Xiong, Victor Zhong, and Richard Socher. 2016. Dynamic Coattention Networks For Question Answering. *arXiv preprint arXiv:1611.01604* (2016).

[38] Bishan Yang and Claire Cardie. 2013. Joint Inference for Fine-grained Opinion Extraction.. In *ACL (1)*. 1640–1649.

[39] Wenpeng Yin, Mo Yu, Bing Xiang, Bowen Zhou, and Hinrich Schütze. 2016. Simple Question Answering by Attentive Convolutional Neural Network. *arXiv preprint arXiv:1606.03391* (2016).

[40] Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. Association for Computational Linguistics.

[41] Jiani Zhang, Xingjian Shi, Irwin King, and Dit-Yan Yeung. 2017. Dynamic Key-Value Memory Networks for Knowledge Tracing. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 765–774.