Aspect Level Sentiment Classification with Unbiased Attention and Target Enhanced Representations

Peng Liu[1,2], Tingwen Liu[1,2], Jinqiao Shi[3], Xuebin Wang[1,2] (✉), Zelin Yin[1,2] and Can Zhao[1,2]
[1] Institute of Information Engineering Chinese Academy of Sciences, Beijing, China
[2] School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3] Beijing University of Posts and Telecommunications, Beijing, China
{liupeng1995,liutingwen,wangxuebin,yinzelin,canzhao}@iie.ac.cn
shijinqiao@bupt.edu.cn

ABSTRACT

Aspect-level sentiment classification aims at inferring the sentiment polarities of opinion targets for a given sentence. As a sentence might contain multiple sentiment-target pairs, extracting relevant information concerning the given target entity is the main challenge of this task. We try to overcome the challenge from two aspects. First, the attention mechanism is able to focus on the relevant part of the given entity and is well suited for this task. However, previous attention-based models still suffer from the problem of paying too much attention to some sentiment words that are irrelevant to the target. We call this as attention bias problem. To alleviate the biases, in this work, we introduce an adversarial training method to get unbiased attention. Second, we try to enhance the impact of the target from the perspective of word representations. Thus we propose an Embedding-Preserving Gating (EPGating) Mechanism. The mechanism dynamically incorporates target-related features into word representations as well as retains original word information. The experimental results on SemEval datasets demonstrate the effectiveness of our model.

CCS CONCEPTS

• **Information systems** → **Sentiment analysis**; *Information extraction*; Web mining;

KEYWORDS

Aspect-based sentiment classification, Attention bias, Adversarial training, Gating mechanism

ACM Reference Format:
Peng Liu[1,2], Tingwen Liu[1,2], Jinqiao Shi[3], Xuebin Wang[1,2] (✉), Zelin Yin[1,2] and Can Zhao[1,2]. 2020. Aspect Level Sentiment Classification with, Unbiased Attention and Target Enhanced Representations. In *The 35th ACM/SIGAPP Symposium on Applied Computing (SAC '20), March 30-April 3, 2020, Brno, Czech Republic.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3341105.3373869

SAC '20, March 30-April 3, 2020, Brno, Czech Republic

ACM ISBN 978-1-4503-6866-7/20/03…$15.00
https://doi.org/10.1145/3341105.3373869

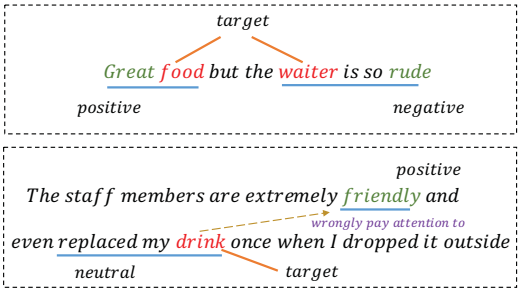

Figure 1: Examples for aspect-level sentiment classification and attention bias respectively.

1 INTRODUCTION

The goal of aspect-level sentiment classification is to identify the sentiment polarities toward the target entities that appear in text. Take the first sentence in Figure 1 as an example, the given opinion targets are *"food"* and *"waiter"*, the goal is to predict their sentiment polarities, which are positive and negative respectively. One important challenge in aspect-level sentiment classification is how to extract relevant information concerning the target entities and differentiate sentiment polarities towards different targets. We believe that there are two ways to overcome this challenge. One is to apply the attention mechanism, which has shown to be effective in previous state-of-art models. The other is to learn context representations based on relevance to the opinion targets. Our work starts from the above two aspects.

First, the attention mechanism has been widely applied to this task [1, 13, 19, 23, 25]. Most attention-based models used to feed the sequence of word embeddings into a bi-directional long short-term memory (BiLSTM) layer, then attention weights are computed based on the hidden states of the targets and other words. Despite the attention mechanism has brought significant improvements to aspect-level sentiment classification, there still exists a problem. That is, sentiment words tend to get more attention despite the fact that they may be irrelevant to the target entities. As shown in the second sentence of Figure 1, the target word is *"drink"* and its real sentiment polarity is neutral. However, the sentiment word *"friendly"* tends to get surprising high attention which may induce the model to output positive polarity. The phenomenon in which attention cannot focus on the really crucial part is also observed in machine translation [20] and is called as attention bias problem.

We propose an adversarial framework to alleviate the biases. As mentioned above, attention is calculated based on the sequence of word embeddings. Intuitively, if the word sequence has no preference for sentiment polarity, attention may not be biased to sentiment words. Based on this idea, we consider removing sentiment information from the word embedding layer, which will be fed into a BiLSTM to calculate attention scores. Thus we design a sentiment discriminator. The discriminator learns its parameters by maximizing its classification accuracy. On the contrary, the word embedding layer for attention is optimized to confuse the discriminator to minimize its accuracy. In the wake of convergence of adversarial training process, the discriminator will fail to distinguish sentiment polarity of the word embedding layer, which means that different sentiment polarities of words lie in a similar region in embedding space and sentiment information has been discarded.

Second, a sentence sometimes contains multiple opinion targets, the contexts of which may express different sentiment information. For example, in the sentence *"great food but the service was dreadful!"*, there exist different sentiment polarities toward *"food"* and *"service"*. When judging the sentiment polarity about *"service"*, we hope that the sentiment information irrelevant to the target can be filtered. To this end, we propose an Embedding-Preserving Gating (EPGating) Mechanism to learn word representations. The gating mechanism, which can optionally regulate the flow of information, will help to learn word representations based on the relevance to the opinion targets. In the example above, *"great"* is irrelevant to *"service"* and its sentiment polarity will be weakened. Intuitively, the gating mechanism may discard some original word information which may be useful for sentiment classification. To preserve original word information as much as possible, word embeddings are retained in a similar way as the residual connection [6]. Instead of conventional summation operation, the concatenation operation is conducted to preserve original word information better. Finally, attention-weighted word representations will be fed into a convolutional neural network.

The main contributions of our paper can be summarized as follows:

- We propose an adversarial architecture to remove sentiment information from the word sequence. The architecture can help to alleviate attention biases brought by sentiment words.
- We propose an EPGating Mechanism to learn target-aware word representations.
- Experimental results on two benchmark datasets demonstrate that our model outperforms the comparative baselines significantly.

2 RELATED WORK

2.1 Aspect Level Sentiment Classification

Early works use dictionary-based methods [17]. Traditional supervised learning approaches [8, 10] try to define rich handcrafted features to solve aspect-level sentiment classification, which may be labor-intensive. Neural network approaches can automatically learn useful features and are widely used in this task. Especially, recurrent neural networks with attention mechanism [1, 13, 19, 25] have shown promising results. [19] introduced a deep memory network which utilizes attention mechanism to extract target-related information. [1] improved [19] by combining the results of multiple attentions with a GRU network. [13] proposed an interactive attention network to interactively learn attentions in the contexts and targets. We review their models and find that they experience the attention bias problem which may occur when there exists multiple sentiment words in a sentence. In this paper, we address this problem via an adversarial training method.

2.2 Adversarial Training

In the NLP community, adversarial training is firstly used for domain adaption [5], which aims at learning invariant features with respect to the shift between the source domain and target domain. For a similar reason, it is also employed in style transfer [4], NER from crowd annotations [24], multi-task learning [2, 12] and cross-lingual learning [9, 26]. Most of them have an additional discriminator used to learn similar distributions in the source and target domains. In our work, we introduce an adversarial training method to alleviate the biases in attention.

3 MODEL DESCRIPTION

In this section, we try to give a description about the architecture of our model, as shown in Figure 2. For a given sentence $s = (w_1, w_2, w_3, ..., w_n)$ consisting of n words and an opinion target $t = (t_1, t_2, t_3 ..., t_m)$ consisting of m words that occur in the sentence s, we assume that the position of t in s is $[p, p + m)$. The goal of our model is to infer the sentiment polarity y of the target t. The sentiment polarity belongs to the set P = {"positive", "negative", "neutral"}.

3.1 Word Embedding Layer

The bottom of our model are two word embedding layers. Like most of the models in NLP tasks, we use pre-trained Glove vectors [14] to get word embeddings. We define $\mathbb{L} \in \mathbb{R}^{d \times |V|}$ as an embedding lookup table generated by Glove, where d is the dimension of vectors and $|V|$ is the size of the vocabulary set V. We retrieve $\mathbb{L}$ and acquire word vectors, which are used to initialize the word embedding layer. As clarified in the introduction, sentiment information will be removed from the embeddings for attention, so we must use another embedding layer to keep sentiment information for the word representations. The two different word embedding layers are defined as $(e_{a_1}, e_{a_2}, e_{a_3}, ..., e_{a_n})$ and $(e_{r_1}, e_{r_2}, e_{r_3}, ..., e_{r_n})$ respectively.

3.2 Word Representation Network

The word representation network aims at transforming the original embeddings into informative word representations.

Contextual Layer: We employ a BiLSTM on top of the embedding layer to get contextualized word representations. The BiLSTM consists of both backward and forward LSTM layers, which help it to make use of past features and future features about sequence at every time step. The BiLSTM layer takes e_{r_i} as input to generate a

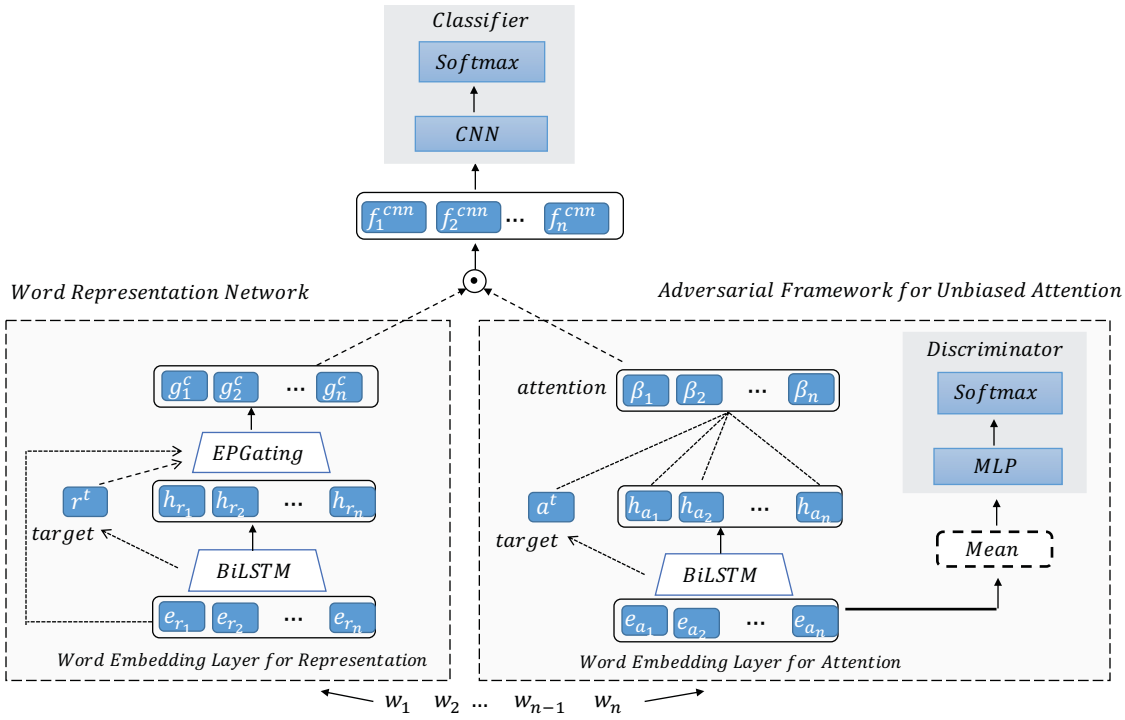

Figure 2: Architecture of our model.

forward hidden state $\overrightarrow{h_{r_i}}$ and a backward hidden state $\overleftarrow{h_{r_i}}$.

$$\overrightarrow{h_{r_i}} = \overrightarrow{\text{LSTM}}(e_{r_i}); \quad i \in [1, n] \tag{1}$$

$$\overleftarrow{h_{r_i}} = \overleftarrow{\text{LSTM}}(e_{r_i}); \quad i \in [1, n] \tag{2}$$

$$h_{r_i} = [\overrightarrow{h_{r_i}}; \overleftarrow{h_{r_i}}] \tag{3}$$

We concatenate $\overrightarrow{h_{r_i}}$ and $\overleftarrow{h_{r_i}}$ to get $h_{r_i} \in \mathbb{R}^{2d_h}$, which is the word representation at time step i.

EPGating Mechanism: The gating mechanism is used to control the path through which information flows in a network, and has proven to be useful for the RNN. We use the gating mechanism to incorporate target-related information into the word representations dynamically. Gated Tanh Unit [21], which is one of the effective gating units, is chosen in our model. We compute target-related word representations as:

$$f_i = \tanh(W_1 * h_{r_i} + W_2 * r^t + b) \tag{4}$$

$$s_i = \sigma(V_1 * h_{r_i} + V_2 * r^t + c) \tag{5}$$

$$g_i = s_i \otimes f_i \tag{6}$$

where W_1, W_2, V_1 and $V_2 \in \mathbb{R}^{2d_h \times 2d_h}$, b and $c \in \mathbb{R}^{2d_h}$ are the weights of the gate. σ is the sigmoid function and $\otimes$ is the element-wise product between two vectors. $r^t \in \mathbb{R}^{2d_h}$ is the target representation. Similar to previous works [1, 13], r^t is obtained by averaging the hidden states of the target:

$$r^t = \frac{1}{m} \sum_{k=p}^{p+m-1} h_{r_k} \tag{7}$$

Equation 4 conducts an interaction between h_{r_i} and r_t to generate target-specific features. Equation 5 outputs a vector with values range between 0.0 to 1.0. s_i can be regarded as the relevance vector between the target and the i-th word. Based on s_i, we dynamically adjust f_i to get target-related word representation g_i.

The gating mechanism acts on contextualized feature h_{r_i}, it may discard some original word information which might be useful for sentiment classification. To alleviate this problem, we try to incorporate original word embedding e_{r_i} and propose an embedding-preserving mechanism, which conducts a concatenation operation similar to the residual connection between g_i and e_{r_i} like:

$$g_i^c = [g_i; e_{r_i}] \tag{8}$$

$g_i^c \in \mathbb{R}^{2d_h+d}$ is the final representation of the i-th word.

3.3 Adversarial Method for Unbiased Attention

We have clarified that the attention will be biased toward sentiment words. However, if the sequence of word embeddings has no preference for sentiment polarity, this problem may be addressed. This is because sentiment words cannot be distinguished from other words. Attention can focus on capturing semantic relatedness between the target and its context without being disturbed by irrelevant sentiment words. Thus we propose an adversarial training method to discard sentiment information from the word sequence.

Adversarial Method: To achieve the above objective, we design a sentiment discriminator, which is a multi-layer perceptron (MLP) and its parameters are defined as θ_d. The discriminator is used to classify sentiment polarity of the word embedding layer

for attention, whose parameters are denoted as θ_{emd}. We get input for the discriminator by averaging the embedding sequence $(e_{a_1}, e_{a_2}, e_{a_3}, ..., e_{a_n})$,[1] which is denoted as follows:

$$s_{avg} = \frac{1}{n}\sum_{i=1}^{n} e_{a_i} \tag{9}$$

The word embedding layer is optimized to confuse the discriminator. Thus we design a loss function based on the Kullback-Leibler Divergence:

$$L_{adv1}(\theta_{emb}) = \mathrm{D_{KL}}(Q(y|s_{avg}; \theta_{emb}, \theta_d)||P_u) \tag{10}$$

$Q(y|s_{avg}; \theta_{emb}, \theta_d)$ represents the probability distribution which is outputted by the discriminator condition on s_{avg}. P_u is an equiprobability distribution. $P_u(i) = \frac{1}{3}$, $i \in \{1, 2, 3\}$. θ_{emb} is optimized to minimize $L_{adv1}(\theta_{emb})$. According to the characteristics of KL, $Q(y|s_{avg})$ will be close to equiprobability distribution when $L_{adv1}(\theta_{emb})$ becomes small, which means that the word embedding layer for attention has no preference for sentiment polarity.

The discriminator learns its parameters by maximizing its classification accuracy. The loss function of the sentiment discriminator minimizes the negative log probability of the sentiment label:

$$L_d(\theta_d) = -\log P(y_i|s_{avg}; \theta_{emb}, \theta_d) \tag{11}$$

Attention Layer: e_{a_i} and e_{r_i} have gaps in distribution as e_{a_i} doesn't keep sentiment information, so we use different LSTM to model them. The hidden states in the LSTM for attention can be generated by:

$$h_{a_i} = [\overrightarrow{\mathrm{LSTM}}(e_{a_i}), \overleftarrow{\mathrm{LSTM}}(e_{a_i})] \tag{12}$$

where $h_{a_i} \in \mathbb{R}^{2d_h}$. The target representation can be calculated in the same way as Equation 7:

$$a^t = \frac{1}{m}\sum_{k=p}^{p+m-1} h_{a_k} \tag{13}$$

For every hidden state, attention layer computes a weight β_i which can explicitly reveal the importance of each context word. The process can be described as:

$$f(h_{a_i}, a^t) = \tanh(h_{a_i}^T a^t) \tag{14}$$

$$\beta_i = \frac{\exp(f(h_{a_i}, a^t))}{\sum_{k=1}^{n} \exp(f(h_{a_k}, a^t))} \tag{15}$$

where $\beta_i \in \mathbb{R}$.

As $L_{adv1}(\theta_{emb})$ is optimized, the discriminator will misclassify sentiment polarity of the word embedding layer. This is because different sentiment polarities of words lie in a similar region in embedding space. However, similar word embeddings may make context words be similar to each other. Attention weights of different context words will become similar. To ensure diversified distribution for attention weights, we minimize the regularization term to enforce the uniqueness:

$$L_{adv2}(\theta_a, \theta_{emb}) = \mathrm{H}(P_{atten}) \tag{16}$$

[1]There are many methods to represent a sentence. Using the average value is popular and useful. We choose a simple way as it is not our main point.

Denote θ_a as the parameters of the attention layer. P_{atten} represents the distribution of attention $(\beta_1, \beta_2, ...\beta_n)$ and $\mathrm{H}(P_{atten})$ is the entropy of P_{atten}. As entropy decreases, P_{atten} will deviate from the uniform distribution.

3.4 CNN Classifier

The sentiment polarity of a target is usually determined by some key phrases. We choose CNN as our classifier as it is useful for extracting n-gram features. To produce final feature f_i^{cnn} for CNN, we weight word representation g_i^c with attention β_i, namely $f_i^{cnn} = \beta_i g_i^c$. We then use a convolutional layer and a max-pooling layer to process f^{cnn}:

$$c_i = \mathrm{relu}(W_{conv}[f_{i-1}^{cnn}, f_i^{cnn}, f_{i+1}^{cnn}] + b_{conv}) \tag{17}$$

$$z = \mathrm{maxpooling}(c_1, c_2, ..., c_n) \tag{18}$$

where $i \in [1, n]$, W_{conv} and b_{conv} are parameters of the convolutional kernel. Finally, z is fed into a softmax layer to predict sentiment polarity of the target. The cross-entropy loss function is given below:

$$L_c(\theta_c, \theta_{emb}) = -\log P(y_i|z; \theta_c, \theta_{emb}) \tag{19}$$

where θ_c is the parameter of the whole model except θ_{emb} and θ_d.

3.5 Training Objective

The objective functions we mentioned above include two parts. $L_c(\theta_c, \theta_{emb})$ is for the target sentiment classification. $L_{adv1}(\theta_{emb})$ and $L_{adv2}(\theta_a, \theta_{emb})$ are for getting unbiased attention via adversarial training. The final loss function is defined as:

$$L_{total}(\theta_c, \theta_{emb}) = L_c + \lambda(L_{adv1} + L_{adv2}) \tag{20}$$

where λ is a hyperparameter to control the influence of the adversarial loss. The discriminator and other parts of the model are trained alternately to minimize $L_d(\theta_d)$ and $L_{total}(\theta_c, \theta_{emb})$ respectively. Another phenomenon we found is that the discriminator converges faster than other parts, thus we train the discriminator every D_{step} epoch, D_{step} is a hyperparameter to set.

4 EXPERIMENTS

4.1 Experimental Setting

Dataset Settings: We conduct experiments on two datasets: REST14 from SemEval2014 task4 [16] and REST15 from SemEval2015 task12 [15]. These datasets contain lots of user reviews and each review associates with a list of opinion targets and corresponding sentiment polarities. For those user reviews which contain multiple targets, we construct a training sample for every target. We give two examples in Table 1. Following the previous works [7, 19], we remove samples with conflicting polarity in the datasets. After preprocessing, statistics of the final datasets are given in Table 2.

Hyper-parameters: We randomly select 20% of the training data as the development set to tune hyperparameters. We train the network for 200 epochs and select the best model according to the performance on the development set. The final hyperparameters are selected when they produce the highest accuracies on the development set. Finally, the dimension of word embeddings (from Glove) is set to 300 and the dimension of the BiLSTM hidden vectors is 100. The convolutional kernel size is 3. The weight matrix

Input		Output
Sentence	opinion target	sentiment label
Great food but the service was dreadful	food	positive
Great food but the service was dreadful	service	negative

Table 1: Two example sentences of restaurant review dataset from SemEval 2014.

Dataset	Pos	Neg	Neu
REST14 Train	2,164	807	637
REST14 Test	728	196	196
REST15 Train	1,178	382	50
REST15 Test	439	328	35

Table 2: Statistics of datasets.

and bias are initialized by sampling from a uniform distribution U(-0.01, 0.01). To avoid overfitting, we apply dropout on the input word embeddings. Adam with learning rate set to 0.0005 is used for optimizing the models. During our experiments, we found that the dropout rate and λ have relatively significant influence on the results. The experiments for hyperparameter tuning of the full model on the development set of REST15 are illustrated in Figure 3. We tune dropout rate and λ from 0 to 0.9 respectively. Finally, we set the dropout rate of experiments on REST14 and REST15 to 0.85 and 0.6. The hyperparameter λ is set to 0.3 for REST14 and 0.1 for REST15. D_{step} is set to 20, which means that the discriminator is trained every 20 rounds.

4.2 Compared Methods

To evaluate the effectiveness of our method, our model is compared with the following baselines:

- **LSTM+ATT**: A common BiLSTM with attention mechanism.
- **TDLSTM**: It uses a forward LSTM and a backward LSTM to model the preceding and following contexts surrounding the target, then concatenates the last hidden states from the two LSTMs to classify the sentiment polarity [18].
- **ATAE-LSTM**: It appends the target embedding into each word input vector which is fed into an attention-based LSTM model [23].
- **MemNet**: MemNet employs multiple attention layers to select abstractive evidences from memory, the output vector of the last layer is considered as the sentence representation with regard to the opinion target [19].
- **IAN**: IAN uses two LSTMs to model the target and context respectively, then employs interactive attention to get final sentence representation [13].
- **RAM**: RAM uses multiple attention layers to extract target-related information from a positon-weighted memory, and nonlinearly combine the result from memory with a GRU [1].
- **TMN**: TMN proposes target-sensitive memory networks, which can capture the sentiment interaction between the targets and contexts, to address the target-dependent sentiment problem [22].
- **TNet**: TNet is a CNN-based model, it introduces a target-specific transformation component for generating the target-specific word representations. Besides, TNet designs a context-preserving mechanism to preserve the original contextual information from the RNN layer [11].
- **MGAN**: MGAN leverages both the coarse-grained and fine-grained attentions to compose the multi-grained attention network [3].
- **LSTM+SynATT+TarRep**: It represents the target as a weighted summation of aspect embeddings and proposes an attention mechanism that encodes the syntactic structure of a sentence [7].

To investigate the impact of the EPGating mechanism and unbiased attention independently, we develop other ablated models. LSTM+ATT+CNN is our proposed baseline. It uses a BiLSTM to generate a representation vector for every context word, the representations weighted with common attention are fed into a CNN network. Based on this model, we add the gating mechanism and unbiased attention. LSTM+ATT+EPGating+CNN denotes the model where the EPGating mechanism is added based on LSTM+ATT+CNN. LSTM+AdvATT+CNN denotes the model where common attention is replaced by the unbiased attention via adversarial training and other parts remain the same as LSTM+ATT+CNN.

4.3 Overall Performance Comparison

The performance of aspect-level sentiment classification is evaluated using Accuracy and Macro-F1, which are widely used in previous works [1, 3, 7, 11]. The experimental results are shown in Table 3. Based on the table, we have the following observations:

(1) MGAN outperforms other attention-based baselines since it utilizes the fine-grained attention. It not only considers to learn the attention weights on context towards the aspect but also considers to learn the weights on aspect words towards the context.

(2) CNN layer can bring significant improvements to this task. Compared with ATAE-LSTM, IAN, TMN and LSTM+ATT, which don't use CNN layer, LSTM+ATT+CNN achieves better results on both accuracy and macro-F1 scores for the two datasets. This is because that the sentiment polarity of a target is usually determined by some key phrases. CNN layer can effectively extract informative n-gram features, which means it can capture not only word-level but also phrase-level information.

	Models	REST14		REST15	
		Acc.	Macro-F1	Acc.	Macro-F1
Baselines	TDLSTM [18]	78.00	66.73	76.39*	58.70*
	ATAE-LSTM [23]	77.20	-	78.48*	62.84*
	MemNet [19]	80.32	-	77.89*	59.52*
	IAN [13]	78.60	-	-	-
	LSTM+ATT	76.83*	66.48*	77.38*	60.52*
	TMN [22]	78.79	68.84	-	-
	RAM [1]	80.23	70.80	79.98*	60.57*
	TNet [11]	80.69	71.27	-	-
	MGAN [3]	**81.25**	71.94	-	-
	LSTM+SynATT+TarRep [7]	80.63	71.32	81.67	66.05
Ablated models	LSTM+ATT+CNN	80.00	70.75	82.67	63.42
	LSTM+AdvATT+CNN	80.98	71.73	82.92	65.11
	LSTM+ATT+EPGating+CNN	80.63	72.11	83.04	66.20
Full model	LSTM+AdvATT+EPGating+CNN	**81.25**†	**73.00**†	**83.17**	**68.09**†

Table 3: The performance comparisons of different methods on the two datasets. The results with $*$ are retrieved from [7]. The rest of the baseline methods are retrieved from the original papers. The best result of each dataset is in bold. The marker † refers to $p-$value < 0.05 when comparing with the ablated models.

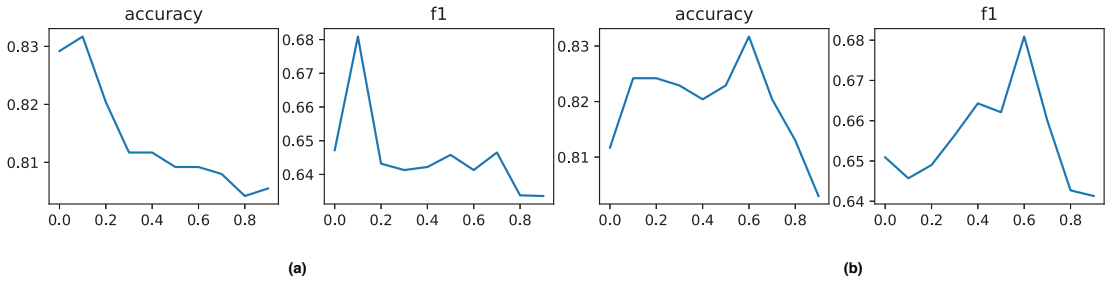

Figure 3: Experiments related to hyperparameters. (a) and (b) show the effect of λ and dropout rate respectively.

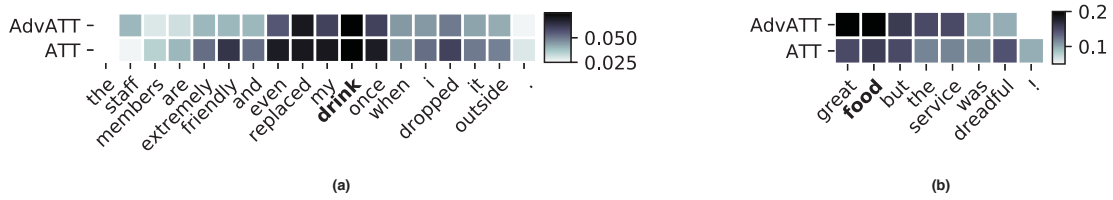

Figure 4: Attention visualization, the opinion target is in bold.

(3) LSTM+AdvATT+CNN outperforms the baseline LSTM+ATT+CNN by 0.98% and 0.25% for accuracy, 0.98% and 1.69% for Macro-F1. This demonstrates that our model can benefit from unbiased attention.

(4) LSTM+ATT+EPGating+CNN can outperform the baseline LSTM+ATT+CNN by 0.63% and 0.37% for accuracy, 1.36% and 2.78% for Macro-F1, which can verify the effectiveness of the gating mechanism.

(5) The adversarial attention and the EPGating mechanism are complementary. The full model achieves best performance compared to the ablated models.

Loss function	Acc	Macro-F1
L_{total} w/o L_{adv1}	82.04	63.39
L_{total} w/o L_{adv2}	82.04	64.04
L_{total}	82.92	65.11

Table 4: Experimental results of the ablated adversarial loss on the REST15.

Models	Acc	Macro-F1
LAC+Gating	81.92	64.81
LAC+EP	82.67	64.12
LAC+EPGating	83.04	66.20

Table 5: Experimental results of the embedding-preserving mechanism and the gating mechanism on the REST15.

4.3.1 Analysis of Unbiased Attention. In our model, we try to get unbiased attention using the adversarial training method. The adversarial loss includes two parts. L_{adv1} is for removing biases toward sentiment words. As L_{adv1} tends to assimilate word embeddings, we use L_{adv2} to ensure diversified distribution for attention weights. To verify the necessities of L_{adv1} and L_{adv2} respectively, we conduct experiments for ablated adversarial loss on REST15. As shown in Table 4, the experiment with L_{total} outperforms the experiment only with L_{adv1} by 0.88% for accuracy and 1.07% for Macro-F1, and outperforms the experiment only with L_{adv2} by 0.88% for accuracy and 1.72% for Macro-F1. In order to understand how unbiased attention contributes to the model's performance, we pick some test examples from the full model and the ablated model without unbiased attention (LSTM+ATT+EPGating+CNN). Figure 4 shows the visualized attention results. For the sentence in Figure 4(a), the target word is *"drink"*. However, the common attention model also assigns a relatively high weight to the sentiment word *"friendly"*, which actually is used to modify the opinion target *"staff"*. Eventually, this sentence is misclassified as positive polarity. As shown in Figure 4(a), the problem is corrected in the AdvATT model. Figure 4(b) shows a similar phenomenon, where the common attention model pays much attention to the irrelevant sentiment word *"dreadful"*.

4.3.2 Analysis of EPGating Mechanism. The gating mechanism is used for extracting target-related features and the embedding-preserving mechanism is used for preserving original word information. In this section, we try to verify their necessities respectively. LAC stands for LSTM+ATT+CNN. As shown in Table 5, LAC+EPGating achieves the best performance. Specifically, LAC+EPGating outperforms LAC+EP by 0.34% for accuracy and 2.08% for Macro-F1, and outperforms LAC+Gating by 1.12% for accuracy and 1.39% for Macro-F1. Experimental results demonstrate that the embedding-preserving mechanism and the gating mechanism can both contribute to word representations.

CONCLUSION

We propose two mechanisms, namely unbiased attention and the EPGating mechanism, to make full use of target-related information for aspect-level sentiment classification. First, we re-examine the drawbacks of previous attention-based models and find that the attention mechanism often suffers from the problem of paying much attention to sentiment words. To make the attention focus on the really crucial part for the target, a novel adversarial training method is used to get unbiased attention. Second, we propose an EPGating mechanism to learn target-aware word representations. The gating mechanism adjusts the word representations based on relevance to the targets, thus target-related sentiment information can be enhanced. Experimental results on two benchmark datasets from the SemEval-Task demonstrate the effectiveness of unbiased attention and the EPGating mechanism.

ACKNOWLEDGMENTS

This work was supported by the National Key Research and Development Program of China (Grant No.2017YFC0820700) and National Defense Science and Technology Innovation Special Zone Project.

REFERENCES

[1] Peng Chen, Zhongqian Sun, Lidong Bing, and Wei Yang. 2017. Recurrent Attention Network on Memory for Aspect Sentiment Analysis. In *EMNLP*. Association for Computational Linguistics, 452–461.
[2] Xinchi Chen, Zhan Shi, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-Criteria Learning for Chinese Word Segmentation. In *ACL (1)*. Association for Computational Linguistics, 1193–1203.
[3] Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained Attention Network for Aspect-Level Sentiment Classification. In *EMNLP 2018*. Association for Computational Linguistics, 3433–3442.
[4] Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2018. Style Transfer in Text: Exploration and Evaluation. In *AAAI*. AAAI Press, 663–670.
[5] Yaroslav Ganin and Victor S. Lempitsky. 2015. Unsupervised Domain Adaptation by Backpropagation. In *ICML*. JMLR.org, 1180–1189.
[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *CVPR*. IEEE Computer Society, 770–778.
[7] Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2018. Effective Attention Modeling for Aspect-Level Sentiment Classification. In *COLING*. Association for Computational Linguistics, 1121–1131.
[8] Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent Twitter Sentiment Classification. In *ACL*. The Association for Computer Linguistics, 151–160.
[9] Joo-Kyung Kim, Young-Bum Kim, Ruhi Sarikaya, and Eric Fosler-Lussier. 2017. Cross-Lingual Transfer Learning for POS Tagging without Cross-Lingual Resources. In *EMNLP*. Association for Computational Linguistics, 2832–2838.
[10] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. NRC-Canada-2014: Detecting Aspects and Sentiment in Customer Reviews. In *SemEval@COLING*. The Association for Computer Linguistics, 437–442.
[11] Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation Networks for Target-Oriented Sentiment Classification. In *ACL 2018*. Association for Computational Linguistics, 946–956.
[12] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial Multi-task Learning for Text Classification. In *ACL (1)*. Association for Computational Linguistics, 1–10.
[13] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive Attention Networks for Aspect-Level Sentiment Classification. In *IJCAI*. ijcai.org, 4068–4074.
[14] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global Vectors for Word Representation. In *EMNLP*. ACL, 1532–1543.
[15] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. SemEval-2015 Task 12: Aspect Based Sentiment Analysis. In *SemEval@NAACL-HLT*. The Association for Computer Linguistics, 486–495.
[16] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. SemEval-2014 Task 4: Aspect Based Sentiment Analysis. In *SemEval@COLING*. The Association for Computer Linguistics, 27–35.
[17] Kim Schouten and Flavius Frasincar. 2015. Survey on aspect-level sentiment analysis. *IEEE Transactions on Knowledge and Data Engineering* 28, 3 (2015),

813–830.

[18] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016. Effective LSTMs for Target-Dependent Sentiment Classification. In *COLING*. ACL, 3298–3307.

[19] Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect Level Sentiment Classification with Deep Memory Network. In *EMNLP*. The Association for Computational Linguistics, 214–224.

[20] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling Coverage for Neural Machine Translation. In *ACL (1)*. The Association for Computer Linguistics.

[21] Aäron van den Oord, Nal Kalchbrenner, Lasse Espeholt, Koray Kavukcuoglu, Oriol Vinyals, and Alex Graves. 2016. Conditional Image Generation with PixelCNN Decoders. In *NIPS*. Neural Information Processing Systems, 4790–4798.

[22] Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-Sensitive Memory Networks for Aspect Sentiment Classification. In *ACL 2018*. Association for Computational Linguistics, 957–967.

[23] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based LSTM for Aspect-level Sentiment Classification. In *EMNLP*. The Association for Computational Linguistics, 606–615.

[24] YaoSheng Yang, Meishan Zhang, Wenliang Chen, Wei Zhang, Haofen Wang, and Min Zhang. 2018. Adversarial Learning for Chinese NER From Crowd Annotations. In *AAAI*. AAAI Press, 1627–1635.

[25] Yue Zhang and Jiangming Liu. 2017. Attention Modeling for Targeted Sentiment. In *EACL (2)*. Association for Computational Linguistics, 572–577.

[26] Bowei Zou, Zengzhuang Xu, Yu Hong, and Guodong Zhou. 2018. Adversarial Feature Adaptation for Cross-lingual Relation Classification. In *COLING*. Association for Computational Linguistics, 437–448.