

A Position-aware Bidirectional Attention Network for Aspect-level Sentiment Analysis

Shuqin Gu¹, Lipeng Zhang², Yuexian Hou^{1*} and Yin Song¹

¹School of Computer Science and Technology, Tianjin University, Tianjin, China

²School of Computer Software, Tianjin University, Tianjin, China

{shuqingu,lpzhang, yxhou, songyin}@tju.edu.cn

Abstract

Aspect-level sentiment analysis aims to distinguish the sentiment polarity of each specific aspect term in a given sentence. Both industry and academia have realized the importance of the relationship between aspect term and sentence, and made attempts to model the relationship by designing a series of attention models. However, most existing methods usually neglect the fact that the position information is also crucial for identifying the sentiment polarity of the aspect term. When an aspect term occurs in a sentence, its neighboring words should be given more attention than other words with long distance. Therefore, we propose a position-aware bidirectional attention network (PBAN) based on bidirectional GRU. PBAN not only concentrates on the position information of aspect terms, but also mutually models the relation between aspect term and sentence by employing bidirectional attention mechanism. The experimental results on SemEval 2014 Datasets demonstrate the effectiveness of our proposed PBAN model.

1 Introduction

Sentiment analysis, also known as opinion mining (Liu, 2012; Pang et al., 2008), is a vital task in Natural Language Processing (NLP). It divides the text into two or more classes according to the affective states and the subjective information of the text, and has received plenty of attention from both industry and academia. In this paper, we address the aspect-level sentiment analysis, which is a fine-grained task in the field of sentiment analysis. For instance, given the mentioned aspect terms $\{menu, server, specials\}$, and the sentence is “*The menu looked good, except for offering the Chilean Sea Bass, but the server does not offer up the specials that were written on the board outside.*”. For aspect term *menu*, the sentiment polarity is *positive*, but for aspect term *server*, the polarity is *negative* while for *specials*, the polarity is *neutral*.

One important challenge in aspect-level sentiment analysis is how to model the semantic relationship between aspect terms and sentences. Traditional approaches have defined rich features about content and syntactic structures so as to capture the sentiment polarity (Jiang et al., 2011). However this kind of feature-based method is labor-intensive and highly depends on the quality of the features. Compared with these methods, neural network architectures are capable of learning features without feature engineering, and have been widely used in a variety of NLP tasks such as machine translation (Cho et al., 2014), question answering (Andreas et al., 2016) and text classification (Lai et al., 2015). Recently, with the development of the neural networks, they are also applied to target-dependent sentiment analysis¹, such as Target-Dependent LSTM (TD-LSTM) (Tang et al., 2015) and Target-Connection LSTM (TC-LSTM) (Tang et al., 2015). However, these neural network-based methods can not effectively identify which words in the sentence are more important. Fortunately, attention mechanisms are an effective way to solve this problem.

Corresponding author: Yuexian Hou.

This work is licensed under a Creative Commons Attribution 4.0 International License. Licence details: <http://creativecommons.org/licenses/by/4.0/>

¹The aim of the target-dependent sentiment analysis is similar to aspect-level sentiment analysis. Given a sentence and target/aspect term, the task calls for inferring the sentiment polarity of the sentence towards the target/aspect term.

Attention, which is widely applied to Computer Vision (CV) and NLP fields, is an effective mechanism and has been demonstrated in image recognition (Mnih et al., 2014), machine translation (Bahdanau et al., 2014; Luong et al., 2015) and reading comprehension (Hermann et al., 2015; Cui et al., 2016). Therefore, some researchers have designed attention networks to address the aspect-level sentiment analysis and have obtained comparable results, such as AE-LSTM (Wang et al., 2016), ATAE-LSTM (Wang et al., 2016) and IAN (Ma et al., 2017).

However, these existing work ignores or does not explicitly model the position information of the aspect term in a sentence, which has been studied for improving performance in information retrieval (IR). In (Liu et al., 2015), the occurrence positions of the query terms were modeled via kernel functions and then integrated into traditional IR models to boost the retrieval performance. By analyzing this aspect-level sentiment analysis task and the corresponding dataset, we find that when an aspect term occurs in a sentence, its neighboring words in the sentence should be given more attention than other words with long distance. Let us take “*It’s a perfect place to have an amazing indian food.*” as an example, when the aspect term is *indian food*, its corresponding sentiment polarity is *positive*. Intuitively, we can see that the neighboring word of the *indian food* (i.e. “*amazing*”) has a greater contribution to judge the sentiment polarity of the aspect term than other words with long distance such as “*to*” and “*have*”. Sometimes this intuitive idea of judging the sentiment polarity may be interpreted as a cognitive activity, which also can be rephrased in a quantum-like language model (Niu et al., 2017). To be specific, sentiment polarity may be interpreted as a quantum-like cognition state. Inspired by this, we go one step further and propose a position-aware bidirectional attention network (PBAN) based on bidirectional Gated Recurrent Units (Bi-GRU) (Cho et al., 2014).

In addition to utilizing the position information, PBAN also mutually models the relationship between the sentence and different words in the aspect term by adopting a bidirectional attention mechanism. To be specific, our model consists of three components: 1) Obtaining position information of each word in corresponding sentence based on the current aspect term, then converting the position information into position embedding. 2) The PBAN composes of two Bi-GRU networks focusing on extracting the aspect-level features and sentence-level features respectively. 3) Using the bidirectional attention mechanism to model the mutual relation between aspect term and its corresponding sentence. We evaluate our models on SemEval 2014 Datasets, and the results show that our models are more effective than other previous methods.

The main contributions of our work can be summarized as follows: (1) We attempt to explicitly investigate the effectiveness of the position information of aspect term for aspect-level sentiment analysis. (2) We propose a position-aware bidirectional attention network (PBAN) based on Bi-GRU, which has been proved to be effective to improve the sentiment analysis performance. (3) We apply a bidirectional attention mechanism, which can enhance the mutual relation between the aspect term and its corresponding sentence, and prevent the irrelevant words from getting more attention.

2 Model Overview

In this section, we describe the proposed model position-aware bidirectional attention network (PBAN) for aspect-level sentiment analysis and PBAN is shown in Figure 1. In this paper, the set of sentiment polarity of the aspect term is $\{positive, negative, neutral\}$.

2.1 Position Representation

As for how to model the position information of the aspect term in its corresponding sentence, inspired by the position encoding vectors used in (Collobert et al., 2011; Zeng et al., 2014), we define a position index sequence whose length is equal to the length of corresponding sentence. Suppose that if a word in the aspect term occurs in the sentence, then its position index will be marked as “0”, and the position index of other words will be represented as the relative distance to the current aspect term.

$$p_i = \begin{cases} |i - j_s|, & i < j_s \\ 0, & j_s \leq i \leq j_e \\ |i - j_e|, & i > j_e \end{cases} \quad (1)$$

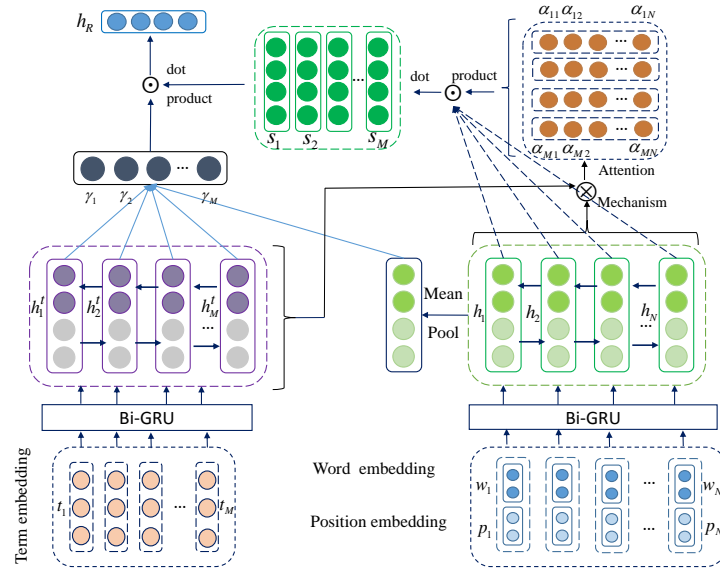


Figure 1: The architecture of position-aware bidirectional attention network for aspect-level sentiment analysis (PBAN). $\{w_1, w_2, \dots, w_N\}$ represents the word embedding in a sentence whose length is N , and $\{t_1, t_2, \dots, t_M\}$ represents the aspect term embedding whose length is M . $\{p_1, p_2, \dots, p_N\}$ is the position embedding of the aspect term, which is concatenated to the word embedding. $\{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_N\}$ denotes the hidden representation of inputs and $\{\mathbf{h}_1^t, \mathbf{h}_2^t, \dots, \mathbf{h}_M^t\}$ indicates the hidden representation of aspect term.

where, j_s and j_e denote the starting and ending indices of the aspect term respectively, and p_i can be viewed as the relative distance of the i -th word in sentence to the aspect term. For example, given a sentence “not only was the food outstanding but the little perks were great.”, and the aspect term is *food*, then the position index sequence is represented as $\mathbf{p} = [4, 3, 2, 1, 0, 1, 2, 3, 4, 5, 6, 7]$. And its corresponding position embeddings are obtained by looking up a position embedding matrix $\mathbf{P} \in \mathbb{R}^{d_p \times N}$, which is randomly initialized, and updated during the training process. Here, d_p denotes the dimension of position embedding and N indicates the length of the sentence. After the position index is converted to the embedding, the position embedding can model the different weights of words with different distance. From this example, it is obvious that the words with smaller relative distances (such as “outstanding”) play a more important role in judging the sentiment polarity of *food*. We can find that this process is basically consistent with the way people judge the sentiment polarity of the aspect term. Because we usually first observe the neighboring words of the aspect term, judging whether the neighboring words can show its sentiment polarity, after that we will focus on those words with long distance.

2.2 Word Representation

Bidirectional LSTMs have been successfully applied to various NLP tasks (Bahdanau et al., 2014), and it models the context dependency with the forward LSTM and the backward LSTM. The forward LSTM handles the sentence from left to right, and the backward LSTM processes it in the reverse order. Therefore, we can obtain two hidden representations, and then concatenate the forward hidden state and backward hidden state of each word. In this paper, we choose to use bidirectional GRU since it performs similarly to bidirectional LSTM but has fewer parameters and lower computational complexity.

Concretely, we firstly obtain the representation of each word in aspect term and sentence, and formalize the notations in our work. We suppose that a sentence consists of N words $[w_1, w_2, \dots, w_N]$ and an aspect term contains M words $[t_1, t_2, \dots, t_M]$, then we get sentence embedding and aspect term embedding by looking up a word embedding matrix $\mathbf{E} \in \mathbb{R}^{d \times v}$ respectively, where d denotes the dimension of the embedding, and v indicates the vocabulary size.

Then we input aspect term embeddings into the left Bi-GRU to get the hidden contextual representa-

tion, which consists of forward hidden state $\vec{\mathbf{h}}_i^t \in \mathbb{R}^{d_h}$ and backward hidden state $\overleftarrow{\mathbf{h}}_i^t \in \mathbb{R}^{d_h}$, where d_h denotes the number of hidden units. Finally, the hidden contextual representation of aspect term \mathbf{h}_i^t is obtained by concatenating $\vec{\mathbf{h}}_i^t$ and $\overleftarrow{\mathbf{h}}_i^t$, i.e., $\mathbf{h}_i^t = [\vec{\mathbf{h}}_i^t; \overleftarrow{\mathbf{h}}_i^t] \in \mathbb{R}^{2d_h}$. For the right Bi-GRU structure, we take the concatenation of the position embedding and word embedding as the inputs, then we can obtain the final hidden contextual representation of the inputs, i.e., $\mathbf{h}_i = [\vec{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i] \in \mathbb{R}^{2d_h}$.

2.3 Position-aware Bidirectional Attention Network Model (PBAN)

As shown in Figure 1, attention model in PBAN consists of two parts: including the aspect term to the position-aware sentence part and a position-aware sentence to the aspect term part. For the former part, we can obtain the different hidden contextual representation of a sentence according to different word in aspect term. For the later part, we can obtain the attention weights of the words in aspect term according to the position information, which is used for getting the final representation of a sentence. Details will be described in following sections.

Aspect term to position-aware sentence attention: A sentence should be represented differently based on different words in aspect term, because different words may have different effects on the final representation of the sentence. We firstly get the hidden contextual representation of the aspect term by the left Bi-GRU, and get the hidden contextual representation of inputs (i.e., the concatenation of word embedding and position embedding) by the right Bi-GRU structure. Here, we regard the position embedding as the part of the inputs, because it intuitively represents the relative distance of words in a sentence to the current aspect term as mentioned in section 2.1. Then we calculate the attention weights by adopting hidden contextual representation of aspect term and inputs, obtaining the attention weight distribution of sentence corresponding to each word in this aspect term. It can be formulated as follows:

$$\mathbf{s}_i = \sum_{j=1}^N \alpha_{ij} \mathbf{h}_j \quad (2)$$

$$\alpha_{ij} = \frac{\exp(f(\mathbf{h}_j, \mathbf{h}_i^t))}{\sum_{k=1}^N \exp(f(\mathbf{h}_k, \mathbf{h}_i^t))} \quad (3)$$

$$f(\mathbf{h}_j, \mathbf{h}_i^t) = \tanh(\mathbf{h}_j^T \mathbf{W}_m \mathbf{h}_i^t + b_m) \quad (4)$$

where α_{ij} indicates the attention weights from the word \mathbf{h}_i^t in the aspect term to the j -th word in the inputs, and \tanh is a non-linear activation function. \mathbf{W}_m is the weight matrix and b_m is the bias. Subsequently, α_{ij} is used to compute a weighted sum of the hidden representation \mathbf{s}_i , producing a semantic vector that represents the input sequence.

Position-aware sentence attention to aspect term: As we mentioned above, different words in aspect term will play different role in judging the sentiment polarity of aspect term. Since we obtain the hidden contextual representation of the inputs by the right Bi-GRU, we utilize both the position and semantic information for calculating the attention weights of different words in aspect term. The process can be formulated as follows:

$$\mathbf{h}_R = \sum_{i=1}^M \gamma_i \mathbf{s}_i \quad (5)$$

$$\gamma_i = \frac{\exp(f(\bar{\mathbf{h}}, \mathbf{h}_i^t))}{\sum_{k=1}^M \exp(f(\bar{\mathbf{h}}, \mathbf{h}_k^t))} \quad (6)$$

$$f(\bar{\mathbf{h}}, \mathbf{h}_i^t) = \tanh(\bar{\mathbf{h}}^T \mathbf{W}_n \mathbf{h}_i^t + b_n) \quad (7)$$

$$\bar{\mathbf{h}} = \frac{1}{N} \sum_{i=1}^N \mathbf{h}_i \quad (8)$$

where γ_i stands for the attention weights from inputs to the words in aspect term, denoting which word in aspect term should be more focused. $\bar{\mathbf{h}}$ is calculated by averagely pooling all Bi-GRU hidden states. Later, the sequence representation \mathbf{x} is obtained by using a non-linear layer:

$$\mathbf{x} = \tanh(\mathbf{W}_R \mathbf{h}_R + \mathbf{b}_R) \quad (9)$$

where \mathbf{W}_R and \mathbf{b}_R are weight matrix and bias respectively.

We feed \mathbf{x} into a linear layer, the length of whose output equals to the number of class labels $\|S\|$. Finally, we add a *softmax* layer to compute the probability distribution for judging the sentiment polarities as *positive*, *negative* or *neutral*:

$$\mathbf{y} = \text{softmax}(\mathbf{W}_s \mathbf{x} + \mathbf{b}_s) \quad (10)$$

where \mathbf{W}_s and \mathbf{b}_s are the weight matrix and bias respectively for *softmax* layer.

2.4 Model training

The PBAN model can be trained in an end-to-end way in a supervised learning framework, the aim of the training is to optimize all the parameters so as to minimize the objective function (loss function) as much as possible. In our work, let y_i be the correct sentiment polarity, which is represented by one-hot vector, and \hat{y}_i denotes the predicted sentiment polarity for the given sentence. We regard the cross-entropy as the loss function, and the formula is as follows:

$$\text{loss} = - \sum_{i=1}^S y_i \log(\hat{y}_i) + \frac{1}{2} \lambda \|\theta\|^2 \quad (11)$$

where λ is the regularization factor and θ contains all the parameters. Furthermore, in order to avoid over-fitting, we adopt the dropout strategy to enhance our PBAN model.

3 Experiments

3.1 Experiments Setting

Parameters Setting: In our experiments, all word embedding are initialized by the pre-trained Glove vector² (Pennington et al., 2014). All the weight matrices are given the initial value by sampling from the uniform distribution $U(-0.1, 0.1)$, and all the biases are set to zero. The dimension of the word embedding and aspect term embedding are set to 300, and the number of the hidden units are set to 200. The dimension of position embedding is set to 100, which is randomly initialized and updated during the training process. We use Tensorflow (Abadi et al., 2016) to implement our proposed model and employ the Momentum as the training method, whose momentum parameter γ is set to 0.9, λ is set to 10^{-6} , and the initial learning rate is set to 0.01.

Dataset: To evaluate our proposed methods, we conduct experiments on the dataset of SemEval 2014 Task4³, the SemEval 2014 dataset consists of reviews in *Restaurant* and *Laptop* datasets. Each review contains a list of aspect terms and corresponding polarities, which are labeled with $\{\textit{positive}, \textit{negative}, \textit{neutral}\}$. Particularly, each aspect term has its character index in the sentence, so that when different aspect term have the same word in a sentence, we can mark the relative position distance of a sentence according to the current aspect term without confusion. Table 1 shows the training and test sample numbers in each sentiment polarity.

3.2 Model Comparison

In order to evaluate the performance of our model, we compare our model with several baseline models, including LSTM (Wang et al., 2016), AE-LSTM (Wang et al., 2016), ATAE-LSTM (Wang et al., 2016), IAN (Ma et al., 2017) and MemNet (Tang et al., 2016).

²Pre-trained word vectors of Glove can be obtained from <http://nlp.stanford.edu/projects/glove/>

³The detail introduction of this dataset can be seen at: <http://alt.qcri.org/semeval2014/task4/>

Datasets	Positive		Negative		Neutral	
	Train	Test	Train	Test	Train	Test
Restaurant	2164	728	807	196	637	196
Laptop	994	341	870	128	464	169

Table 1: Samples of SemEval 2014 Dataset.

LSTM: LSTM takes the sentence as input so as to get the hidden representation of each word. Then it regards the average value of all hidden states as the representation of sentence, and puts it into *softmax* layer to predict the probability of each sentiment polarity. However, it can not capture any information of aspect term in sentence (Wang et al., 2016).

AE-LSTM: AE-LSTM first models the words in sentence via LSTM network and concatenate the aspect embedding to the hidden contextual representation for calculating the attention weights, which are employed to produce the final representation for the input sentence to judge the sentiment polarity (Wang et al., 2016).

ATAE-LSTM: ATAE-LSTM extended AE-LSTM by appending the aspect embedding to each word embedding so as to represent the input sentence, which highlights the role of aspect embedding. The other design of ATAE-LSTM is the same as AE-LSTM (Wang et al., 2016).

IAN: IAN considers the separate modeling of aspect terms and sentences respectively. IAN is able to interactively learn attentions in the contexts and aspect terms, and generates the representations for aspect terms and contexts separately. Finally, it concatenates the aspect term representation and context representation for predicting the sentiment polarity of the aspect terms within its contexts (Ma et al., 2017).

MemNet: MemNet applies attention multiple times on the word embedding, so that more abstractive evidences could be selected from the external memory. The output of the last attention layer is fed to a *softmax* layer for predictions (Tang et al., 2016).

Datasets	Restaurant		Laptop	
	Three-class	Two-class	Three-class	Two-class
LSTM	74.28	—	66.45	—
AE-LSTM	76.60	89.60	68.90	87.40
ATAE-LSTM	77.20	90.90	68.70	87.60
IAN	78.60	—	72.10	—
MemNet(9)	80.95	—	72.21	—
PBAN	81.16	91.67	74.12	87.81

Table 2: Comparison with baselines. Accuracy on Three-class and Two-class prediction about *Restaurant* and *Laptop* dataset, and Two-class denotes $\{positive, negative\}$. MemNet(9) indicates that MemNet with nine computational layers. Best scores are in bold.

Table 2 shows the performance of our model and other baseline models on datasets *Restaurant* and *Laptop* respectively. We can observe that our proposed PBAN model achieves the best performance among all methods. It is obvious that LSTM method gets the worst performance, because it treats aspect term and other words as the same, so that it can not take full advantage of the aspect term information and predicts the same polarity for different aspect terms in a sentence.

Furthermore, both AE-LSTM and ATAE-LSTM perform better than LSTM model, because they all consider the importance of the aspect term, and utilize the attention mechanism. Specifically, ATAE-LSTM outperforms AE-LSTM since it appends the aspect embedding to each word embedding and takes them as inputs, which helps the model obtain more semantic information related to aspect term. IAN realizes the importance of interaction between aspect term and context, and models aspect term and context using two connected attention networks. Thus, IAN performs better than ATAE-LSTM, and achieves an improvement of 1.40 points and 3.40 points on *Restaurant* and *Laptop* datasets in *Three-class*

respectively. MemNet(9) utilizes a more complex structure that containing nine computational layers, and it achieves better results compared to IAN since MemNet reads the useful information from external memory repeatedly.

Although both IAN and MemNet models performance better than other methods, they all perform less competitive than our PBAN both on *Restaurant* and *Laptop* datasets. For IAN model, it interactively learns the attentions between the aspect term and its corresponding sentence, but this attention mechanism is coarse-grained and it does not fully consider the influence of different words in aspect term on the sentence. For MemNet model, although it utilizes the location information, it is mainly used for calculating the memory vectors. Nevertheless, PBAN utilizes the character index of the aspect term (provided in the raw dataset) and adopts relative distance to represent the position sequence. As we have mentioned in previous sections, an aspect term contains several words and different words in aspect term should have different contributions to the final representation of sentence. In PBAN, the position information is regarded as the inputs of the Bi-GRU, so it can help calculate the weights of different words in aspect term and improve the final representation of the sentence. Moreover, when different aspect terms contain the same word, our proposed position information can effectively identify the current aspect term without confusion while MemNet can not.

Generally speaking, by integrating the position information and the bidirectional attention mechanism, PBAN achieves the state-of-the-art performances, and it can effectively judge the sentiment polarity of different aspect term in its corresponding sentence so as to improve the classification accuracy.

3.3 Analysis of PBAN Model

In this section, we design a series of models to demonstrate the effectiveness of our PBAN model. Firstly, we design an ATAE-Bi-GRU model, whose structure is similar with ATAE-LSTM. The only difference between these two models is that ATAE-Bi-GRU uses the Bi-GRU structure rather than LSTM, and other design is the same as ATAE-LSTM. Next we design a BAN model without modeling position embedding, and it just utilizes the representation of aspect term and sentence. In BAN, we still adopt bidirectional attention mechanism to model the relation between aspect term and sentence as PBAN does. The only difference between BAN and PBAN is that BAN without taking the position embedding as a part of inputs. Moreover, we also design a PAN model, whose structure is similar with the ATAE-Bi-GRU model. PAN takes the concatenation of the aspect term embedding and the word embedding as the inputs of the Bi-GRU structure to obtain the hidden contextual representation, and then PAN utilizes this representation and the position embedding of the aspect term to calculate the attention weights, so as to effectively judge the sentiment polarity of an aspect term. From Table 3, we can find that PBAN achieves the best performance among these models.

Dataset	Restaurant	Laptop
ATAE-LSTM	77.20	68.70
ATAE-Bi-GRU	77.68	69.47
PAN	78.07	71.13
IAN	78.60	72.10
BAN	78.74	72.61
PBAN	81.16	74.12

Table 3: Analysis of PBAN model.

Because Bi-GRU structure has a big advantage over LSTM, it is obvious that ATAE-Bi-GRU model performs better than ATAE-LSTM model. For PAN model, it outperforms ATAE-LSTM and ATAE-Bi-GRU models, but it is worse than BAN model. Compared with ATAE-Bi-GRU, the most difference is that PAN utilizes the position embedding to calculate the attention weights rather than the aspect term embedding like ATAE-Bi-GRU. Therefore, according to these three experimental results, we can prove the importance of the position information in aspect-level sentiment analysis task.

As for BAN model, it outperforms IAN model while performs worse than PBAN model. Because

Aspect term	Sentence	Polarity
pizza	This is one great place to eat pizza more out but not a good place for take-out pizza.	positive
take-out pizza	This is one great place to eat pizza more out but not a good place for take-out pizza.	negative

Figure 2: Case Study: The visualized attention weights for sentence and aspect term by PBAN.

compared with IAN model, BAN model can learn more semantic relationship between aspect term and sentence via bidirectional attention mechanism. However, it ignores the position information of aspect term when compared with PBAN model.

As we expect, PBAN achieves the best performance among all these models. This is because in addition to fully considering the position information of the aspect term in its corresponding sentence, PBAN also considers the mutual relationship between aspect term and sentence, which is mainly achieved by a bidirectional attention mechanism.

3.4 A Case Study

To have an intuitive understanding of our proposed model, we visualize the attention weights on the aspect term and sentence in Figure 2. The color depth indicates the importance degree of the weight, the darker the more important. In Figure 2, the sentence is “*This is one great place to eat pizza more out but not a good place for take-out pizza.*”, the polarities are *positive* and *negative* for *pizza* and *take-out pizza* respectively. From Figure 2, we can find that our model is more inclined to consider the neighboring words of the aspect term. For example, when the current aspect term is *pizza*, obviously, its neighboring words such as “*great*”, “*place*” and “*more*” get more attention and play a great role for judging sentiment polarity of *pizza*. However, those words that are far from the current aspect term such as “*but*”, “*not*” and “*take-out*” obtain less attention, which demonstrates the effectiveness of the position information. For aspect term *take-out pizza*, it is obvious that the word “*take-out*” is more important to express the aspect term than the word “*pizza*”. From Figure 2, it is worth noting that some words such as “*good*” and “*place*” get less attention even they are closer to the current aspect term than “*but*” and “*not*”. This is because different words in aspect term have different effect on a sentence, and we apply the bidirectional attention mechanism to choose more useful words. For instance, in this case, PBAN should pay more attention on the word “*take-out*”. Therefore, PBAN is capable of figuring out the important part in a sentence for judging the sentiment polarity by modeling the mutual relation between sentence and different words in aspect term.

4 Related Work

In this section, we will briefly review some research on sentiment analysis in recent years. The previous research can be divided into three directions: traditional machine learning methods, neural network methods and attention network methods.

4.1 Machine Learning for Sentiment Analysis

Traditional machine learning approaches mainly involve text representation and feature extraction, such as bag-of-words models and sentiment lexicons features, then training a sentiment classifier (Prez-Rosas et al., 2012). Rao et al. (2010) demonstrated the utility of graph-based semi-supervised learning framework for building sentiment lexicons. Kaji et al. (2007) explored to use structural clues that could extract polar sentences from HTML documents, and built lexicon from the extracted polar sentences. However, these methods are labor-intensive, and usually results in high dimensional and high sparse phenomenon for the text representation.

4.2 Neural Network for Target-dependent Sentiment Analysis

Since a simple and effective method to learn distributed representation was proposed (Mikolov et al., 2013), neural networks enhance target-dependent sentiment analysis significantly. Vo and Zhang (2015) split a tweet into a left context and a right context according to a given target, using distributed word representations and neural pooling functions to extract features. Tang et al. (2015) proposed TD-LSTM and TC-LSTM, where target information is automatically taken into account. These two models integrated the connections between target words and context words so as to significantly boost the classification accuracy. Zhang et al. (2016) proposed two gated neural networks, one was used to capture tweet-level syntactic and semantic information, and the other was used to model the interactions between the left context and the right context of a given target. With the gating mechanism, the target influenced the selection of sentiment signals over the context.

4.3 Attention Network for Aspect-level Sentiment Analysis

With the successful application of the attention mechanism in machine translation and reading comprehension, it is also applied to aspect-level sentiment analysis in recent years. Wang et al. (2016) examined the latent relatedness of the aspect term and sentiment polarity for aspect-level sentiment analysis. They designed an attention-based LSTM to learn aspect term embedding, and let the aspect term embedding participate in calculating the attention weights. Ma et al. (2017) proposed a new attention model IAN, which considered the separate modeling of aspect terms and could interactively learn attention in the contexts and aspect terms.

Despite the effectiveness of these attention mechanisms, they are coarse-grained and it is still challenging to identify different sentiment polarity at a fine-grained aspect level. However, our PBAN model makes full use of the position information of the aspect term, and PBAN uses a fine-grained bidirectional attention mechanism to model the mutual relationship between the sentence and each word in the aspect term, identifying the importance of the word in the aspect term to obtain a more effective sentence representation as described in Section 1.

5 Conclusion

In this paper, we have proposed a position-aware bidirectional network (PBAN) based on Bi-GRU for aspect-level sentiment analysis. The main idea of PBAN is to utilize the position embedding of aspect term for calculating the attention weights. Moreover, PBAN adopts a bidirectional attention mechanism, which is not only capable of mutually modeling the relation between sentence and different words in aspect term, but also takes advantage of the position information to better judge the sentiment polarity of aspect term. Experimental results on SemEval 2014 Datasets demonstrate that our proposed models can learn effective features and obtain superior performance over the baseline models.

Acknowledgements

This work is funded in part by the national key research and development program of China (2017YFE0111900), the Key Project of Tianjin Natural Science Foundation (15JCZDJC31100), the National Natural Science Foundation of China (Key Program, U1636203), the National Natural Science Foundation of China (U1736103) and MSCA-ITN-ETN - European Training Networks Project (QUARTZ).

References

- Martn Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, and Michael Isard. 2016. Tensorflow: a system for large-scale machine learning.
- Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. *arXiv preprint arXiv:1601.01705*.
- Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *Computer Science*.

- Kyunghyun Cho, Bart Van Merriënboer, 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*.
- Ronan Collobert, Jason Weston, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(1):2493–2537.
- Yiming Cui, Zhipeng Chen, Si Wei, Shijin Wang, Ting Liu, and Guoping Hu. 2016. Attention-over-attention neural networks for reading comprehension. *arXiv preprint arXiv:1607.04423*.
- Karl Moritz Hermann, Tom Koisk, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. In *International Conference on Neural Information Processing Systems*, pages 1693–1701.
- 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*, pages 151–160. Association for Computational Linguistics.
- Nobuhiro Kaji and Masaru Kitsuregawa. 2007. Building lexicon for sentiment analysis from massive collection of html documents. In *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*.
- Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.
- Baiyan Liu, Xiangdong An, and Jimmy Xiangji Huang. 2015. Using term location information to enhance probabilistic information retrieval. In *International Acm Sigir Conference on Research Development in Information Retrieval*, pages 883–886.
- Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.
- Minh-Thang Luong, Hieu Pham, and Christopher D Manning. 2015. Effective approaches to attention-based neural machine translation. *arXiv preprint arXiv:1508.04025*.
- Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. *arXiv preprint arXiv:1709.00893*.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *Computer Science*.
- Volodymyr Mnih, Nicolas Heess, Alex Graves, et al. 2014. Recurrent models of visual attention. In *Advances in neural information processing systems*, pages 2204–2212.
- Xiaolei Niu, Yuexian Hou, and Panpan Wang. 2017. Bi-directional lstm with quantum attention mechanism for sentence modeling. In *International Conference on Neural Information Processing*, pages 178–188.
- Bo Pang, Lillian Lee, et al. 2008. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135.
- Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Vernica Prez-Rosas, Carmen Banea, and Rada Mihalcea. 2012. Learning sentiment lexicons in spanish. In *Eighth International Conference on Language Resources and Evaluation*.
- Delip Rao and Deepak Ravichandran. 2010. Semi-supervised polarity lexicon induction. In *Eacl 2009, Conference of the European Chapter of the Association for Computational Linguistics, Proceedings of the Conference, March 30 - April 3, 2009, Athens, Greece*, pages 675–682.
- Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2015. Effective lstms for target-dependent sentiment classification. *arXiv preprint arXiv:1512.01100*.
- Duyu Tang, Bing Qin, and Ting Liu. 2016. Aspect level sentiment classification with deep memory network. *arXiv preprint arXiv:1605.08900*.

- Duy-Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *IJCAI*, pages 1347–1353.
- Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*, pages 606–615.
- D. Zeng, K. Liu, S. Lai, G. Zhou, and J. Zhao. 2014. Relation classification via convolutional deep neural network.
- Meishan Zhang, Yue Zhang, and Duy-Tin Vo. 2016. Gated neural networks for targeted sentiment analysis. In *AAAI*, pages 3087–3093.