24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Sequence Prediction Model for Aspect-level Sentiment Classification

Qianlong Wang [1] and **Jiangtao Ren*** [2]

Abstract. Aspect-level sentiment classification aims to distinguish the sentiment polarity of each aspect in a given sentence. It is more complex than text-level sentiment classification in that it is a fine-grained task. Existing methods, which formulate this task as predicting the sentiment polarity of a provided (sentence, aspect) pair, tend to ignore the relationship between the sentiment polarity of aspects. In this paper, we propose a sequence prediction model with a sentiment polarity fusion module which sequentially predicts the sentiment polarity of each aspect within sentence. Besides, we use the temporal attention mechanism to keep track of what has been focused on, which discourages repeated attention to the context words with strong sentiment polarity when predicting the sentiment polarity of different aspects. Experimental results on five benchmarking collections illustrate that our proposed model[3] outperforms a range of baseline models by a substantial margin, and further demonstrate that the relationship between the sentiment polarity of aspects is helpful to solve the aspect-level sentiment classification.

1 Introduction

Aspect-level sentiment classification is an important task in the field of natural language processing, which can be applied in many real-world scenarios, such as opinion mining on the aspects of the product. Given a sentence and an aspect occurring in the sentence, this task aims at inferring the sentiment polarity (e.g. *positive, negative, neutral*) of aspect. For example, in sentence "*great food but the service was dreadful!*", the sentiment polarity of aspect *food* is *positive* while that of aspect *service* is *negative*.

In the early stage, most works typically used machine learning algorithms and built sentiment classifier in the supervised manner to handle the aspect-level sentiment classification. Among them, one of the most successful approaches is feature based Support Vector Machine (SVM). Experts could design effective feature templates and make use of external resources like parser and sentiment lexicons to improve classifier performance [1], [8]. Although machine learning algorithms have achieved acceptable results, they are mostly based on feature-engineering which is expensive and needs labor cost.

In recent years, neural networks have achieved great success in the field of sentiment classification [2]. With the development of neural networks, they are also applied to the aspect-level sentiment classification. For instance, Tang et al. introduced a deep memory network [21] for aspect-level sentiment classification which explicitly captures the importance of each context word when inferring the sentiment polarity of an aspect. Besides, to effectively identify which words in the sentence are more important, some researchers designed attention networks to address the aspect-level sentiment classification and obtained comparable results, such as AE-LSTM [22]. Compared with machine learning algorithms, neural network models are capable of learning the powerful features from the sentence without careful feature-engineering and capturing semantic relationship between context words and aspect in a more scalable way.

Though neural network models have achieved promising improvements recently, there are still many details are not studied well, such as taking the relationship between the sentiment polarity of aspects into account. We assume that the relationship between the sentiment polarity of different aspects contributes to improve the accuracy of aspect-level sentiment classification. Look at a example "*great food but the service was dreadful!*". Here, according to the conjunction *but*, we could know that the relationship between the sentiment polarity of aspect *food* and *service* is *opposite*[4]. This relationship enables the model to infer the sentiment polarity of aspect *service* according to that of aspect *food*. Therefore, there are two ways to judge the sentiment polarity of aspect *service* intuitively. One is based on the descriptor *dreadful* of aspect *service*; The other is based on relationship between the sentiment polarity of aspect *food* and *service* when the sentiment polarity of aspect *food* and the conjunction *but* are already known. However, most existing neural network models only consider the first way but ignore the other. In other words, most models identify the descriptor of aspect to solve the aspect-level sentiment classification by modelling the semantic relationship between sentence and aspect.

In this paper, inspired by the tremendous success of sequence-to-sequence framework in machine translation [11], abstractive summarization [18] and other domains, we propose a sequence prediction model to solve the aspect-level sentiment classification. The proposed model consists of an encoder and a decoder with attention mechanism. The decoder uses a long short-term memory (LSTM) [6] to predict the sentiment polarity of each aspect in a given sentence sequentially. To be specific, at time-step t, the decoder inputs the previous state, the representation vector of t-th aspect and the embedding vector of sentiment polarity of t-1-th aspect, and outputs the hidden state s_t. The model then concatenates the hidden state s_t and the context c_t to judge the sentiment polarity of t-th aspect. Here, we apply the attention mechanism [11] to produce the context vector c_t by fo-

[1] School of Data and Computer Science, Sun Yat-sen University, China, Email: wangqlong3@mail2.sysu.edu.cn
[2] School of Data and Computer Science, Guangdong Province Key Lab of Computational Science, Sun Yat-sen University, China, Email: issrjt@mail.sysu.edu.cn, ∗Corresponding author
[3] Source code is available at: `https://github.com/qlwang25/SPM`

[4] In addition to the relationship *opposite*, the relationship *same* and *none* may occur. For instance, in a sentence "*The food is surprisingly good and the decor is nice!*", according to the conjunction *and*, the relationship between the sentiment polarity of aspect *food* and *decor* is *same*.

cusing on different portions of sentence and aggregating the hidden representation of informative words. If we remove the embedding vector of sentiment polarity of t-1-th aspect along with the previous state in the input, our model uses only the representation vector of t-th aspect in aspect-level sentiment classification. That is, our model identifies the descriptor of aspect for judging its sentiment polarity by modelling the semantic relationship between sentence and aspect. Looking from the other side, if we add these two inputs, our model will sequentially predict the sentiment polarity of each aspect within sentence. In this case, our model also predicts the sentiment polarity of t-th aspect using the polarity information of t-1-th aspect. For the conjunction, we assume that the model could capture it automatically through training, just like capturing descriptor. That is, our model can take the relationship between the sentiment polarity of aspects into account when predicting the polarity. Therefore, when classifying, the proposed sequence prediction model not only is capable of capturing the descriptor of aspect, but also considers the relationship between the sentiment polarity of aspects. Moreover, to access the sentiment polarity information of all previous aspects, this paper proposes a sentiment polarity fusion module allowing the model to take advantage of earlier sentiment polarity information. To discourage repeated attention to the context words with strong sentiment polarity when inputting the different aspects, we use the temporal attention mechanism to keep track of what has been focused on and modify the conventional attention distribution.

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

- To the best of our knowledge, this paper is first work to take advantage of the relationship between the sentiment polarity of aspects in the aspect-level sentiment classification task.
- We propose a sequence prediction model[5] with a sentiment polarity fusion module which not only models the semantic relationship between sentence and aspect, but also considers the relationship between the sentiment polarity of aspects.
- To avoid repeated attention to the context words with strong sentiment polarity when predicting the sentiment polarity of different aspects, we apply the temporal attention mechanism for aspect-level sentiment classification.
- We conduct experiments on five benchmarking datasets to verify the effectiveness of our model. Experimental results show that our proposed model outperforms many baselines by a large margin.

The rest of this paper is organized as follows: We first give the related work in Section 2. Then, Section 3 introduces our model in detail and Section 4 presents the experimental results and analysis. Finally, we conclude in Section 5.

2 Related Work

The previous studies on aspect-level sentiment classification could be divided into two directions: traditional machine learning methods and neural network models.

2.1 Traditional Machine Learning Methods

Traditional machine learning methods define rich features and syntactic structures about sentence so as to capture the sentiment polarity of aspect. One of the most successful approaches in literature is feature based SVM. Experts could design effective feature templates and make use of external resources like parser and sentiment lexicons [1], [8]. Most machine learning methods based on feature-engineering, however, are labor-intensive and highly depend on the quality of features.

2.2 Neural Network Models

In recent years, some neural network models have been used for aspect-level sentiment classification. Lakkaraju et al. [9] proposed a hierarchical deep learning based framework for solving the problem of aspect-level sentiment classification in which the joint modeling of aspect and sentiment is carried out. Nguyen and Shirai [14] presented an extension of recursive neural network (RNN) that takes both dependency and constituent trees of a sentence into account to identify sentiment of an aspect. Tang et al. [21] developed a deep memory network that captures importance of context words for aspect-level sentiment classification. Compared with RNN, this approach is simpler and faster. In addition, Wang et al. [22] designed an attention-based LSTM to learn the aspect embedding, and examined the latent relatedness of aspect and sentiment polarity in the aspect-level sentiment classification. Ma et al. [13] proposed an interactive attention network which considers the separate modeling of aspects and could interactively learn attention in the context words and aspect. Gu et al. [5] proposed a position-aware bidirectional network for aspect-level sentiment classification which utilizes the position embedding of aspect for calculating the attention weights. Despite the effectiveness of these neural network models, they formulate the aspect-level sentiment classification task as predicting the sentiment polarity of a provided (sentence, aspect) pair yet tend to ignore the relationship between the sentiment polarity of aspects.

3 The Proposed Model

We introduce the proposed model in detail in this section. Firstly, we give the overview of model. Then, we introduce the details of the proposed sequence prediction model. Finally, we present the loss function.

3.1 Overview

First of all, we define some notations and describe the aspect-level sentiment classification task. Given a sentiment polarity set $\mathbb{P}$, such as $\{$*positive*, *negative*, *neutral*$\}$, a sentence $\mathbf{x} = \{w_1, w_2, ..., w_i, ..., w_n\}$ consisting of n words and m aspects $\{a_1, a_2, ..., a_m\}$ occurring in sentence $\mathbf{x}$, the aspect-level sentiment classification aims at determining the sentiment polarity of each aspect in the sentence $\mathbf{x}$. For example, in a comment about restaurant saying "*great food but the service was dreadful!*", the polarity of aspect *food* is *positive*, while the polarity towards aspect *service* is *negative*. Unlike text-level sentiment classification where one sentiment polarity is assigned to sentence $\mathbf{x}$ according to the global context, each aspect within sentence $\mathbf{x}$ in the aspect-level sentiment classification is assigned with sentiment polarity according to the local context (i.e. the descriptor of aspect). Besides, as we mentioned above, the relationship between the sentiment polarity of aspects is also an obvious characteristic of aspect-level sentiment classification. From the perspective of sequence-to-sequence, the aspect-level sentiment classification task can be modeled as finding an optimal sentiment polarity sequence $\mathbf{y}^*$ that maximizes the conditional probability $p(\mathbf{y}|\mathbf{x})$,

[5] Note that there is a previous work [12] that uses a seq2seq model. However, there are clear differences between two works. They use a seq2seq model to make full use of the overall meaning of the sentence in aspect term extraction, while we use a seq2seq model to consider relationship between the sentiment polarity of aspects in aspect-level sentiment classification.

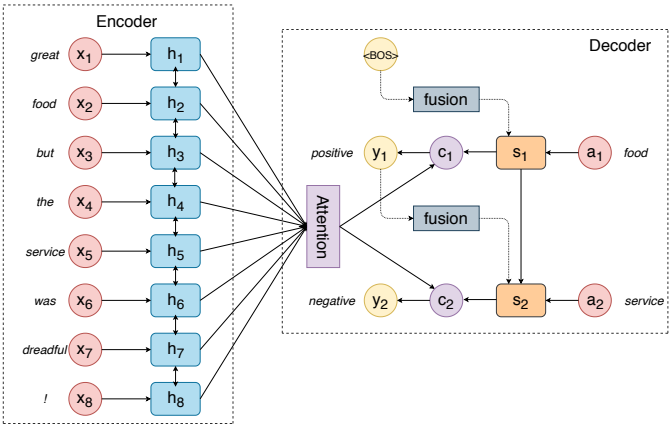


Figure 1. The overview of our proposed model. *Attention* denotes the temporal attention mechanism. *Fusion* denotes the sentiment polarity fusion module. The symbol *BOS* is regarded as the initial sentiment polarity. In this figure, the sentence $\mathbf{x}$ consists of eight words {*great*, *food*, *but*, *the*, *service*, *was*, *dreadful*, *!*} and contains two aspects {*food*, *service*} whose sentiment polarities are *positive* and *negative* respectively.

which is calculated as follows:

$$p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{m} p(y_t|y_1, ..., y_{t-1}, \mathbf{x}) \tag{1}$$

The overview of our proposed model is shown in Figure 1. We input the aspect in turn according to the order in which it appears in the sentence. As shown in Figure 1, the text sequence $\mathbf{x}$ is encoded to the hidden states $\mathbf{h}$. On each time-step t of decoding, the decoder first receives the embedding vector of previous sentiment polarity y_{t-1} (while training, this is the previous ground truth sentiment polarity; at test time it is the previous sentiment polarity predicted by the decoder) and the representation vector of current aspect a_t, then updates its hidden state s_t. Finally, the softmax layer takes the context vector c_t and the current hidden state s_t of the decoder as inputs to calculate the probability distribution of sentiment polarity y_t. Here, the context vector c_t, produced by the temporal attention mechanism which can modify the conventional attention distribution to avoid paying attention to the context words with strong sentiment polarity repeatedly, is an aggregation of the hidden states of the encoder $\mathbf{h}$. Besides, to use earlier sentiment polarity information, our model integrates a sentiment polarity fusion module which produces the polarity embedding vector including the polarity information of all previous aspects.

3.2 The Proposed Sequence Prediction Model

In this subsection, we introduce the details of the proposed model.

3.2.1 Encoder

We first use the WordPiece[6] [23] technique to segment words in sentence $\mathbf{x}$ into subword-level, i.e. $\mathbf{x} = \{w_{1'}, w_{2'}, ..., w_{i'}, ..., w_{n'}\}$ where $n' \geq n$. And then each subword $w_{i'}$ is embedded to a dense embedding vector by WordPiece embeddings $\mathbf{E_w} \in \mathbb{R}^{|V| \times d_w}$. Here $|V|$ is the size of vocabulary, and d_w is the dimension of embedding vector.

BERT[7] [3] is designed to produce the deep bidirectional representations by jointly conditioning on both left and right context in all layers. In this work, to obtain the high-quality representations, we use a BERT to read the text sequence $\mathbf{x}$ and compute the hidden states $\{h_{CLS}, h_1, h_2, ..., h_i, ..., h_{n'}\}$ for all subwords. Note that the first token of every text sequence is always a special symbol *CLS* in the original settings. The final hidden state corresponding to this symbol h_{CLS} is used as the aggregate text sequence representation. Since it is not very useful for our work, we remove it, let the hidden states of the encoder $\mathbf{h} = \{h_1, h_2, ..., h_i, ..., h_{n'}\}$.

3.2.2 Decoder

The decoder uses a LSTM to forecast the sentiment polarity of aspects sequentially. It predicts the current sentiment polarity y_t based on the previously predicted sentiment polarity y_{t-1} and the current aspect a_t. Therefore, the proposed model could consider the relationship between the sentiment polarity of aspects by predicting the sentiment polarity sequentially through LSTM structure (here, the conjunction is automatically captured through training).

The hidden state s_t of the decoder at time-step t is updated as follows:

$$s_t = \text{LSTM}(s_{t-1}, [e_{a_t}; e_{y_{t-1}}]) \tag{2}$$

where $[e_{a_t}; e_{y_{t-1}}]$ means the concatenation of vectors e_{a_t} and $e_{y_{t-1}}$. In this work, we employ the WordPiece technique to segment words in the current aspect a_t into subword-level, and use the average of WordPiece embeddings of these subwords as the representation vector e_{a_t}; we embed the sentiment polarity y_{t-1} to a dense embedding vector $e_{y_{t-1}}$ by sentiment polarity embeddings $\mathbf{E_p} \in \mathbb{R}^{|\mathbb{P}| \times d_p}$. Here $|\mathbb{P}|$ is the size of sentiment polarity set, and d_p is the dimension of embedding vector.

When the model predicts the sentiment polarity of different aspects, not all context words make the same contribution. The attention mechanism is an effective way to consider this. The attention mechanism produces a context vector c_t by focusing on important portions of the text sequence $\mathbf{x}$ and aggregating the hidden representation of informative words. Specially, the attention mechanism [11] assigns a weight α_{ti} to $w_{i'}$ at time-step t as follows:

$$e_{ti} = s_t^T W_a h_i \tag{3}$$

$$\alpha_{ti} = \frac{exp(e_{ti})}{\sum_{j=1}^{n'} exp(e_{tj})} \tag{4}$$

where W_a is weight parameter and s_t is the current hidden state of the decoder. For simplicity, all bias terms are omitted in this paper.

[6] WordPiece can segment out-of-vocabulary words into subword-level. For example, if word "displaying" is not in vocabulary, it is segmented into "display" and "##ing", where "##" means not the beginning of a word.

[7] In this paper, BERT refers to $\text{BERT}_{\text{BASE}}$.

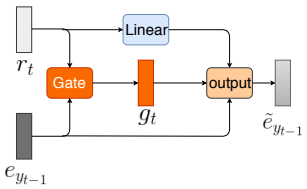


Figure 2. The sentiment polarity fusion module.

The final context vector c_t is calculated as follows:

$$c_t = \sum_{i=1}^{n'} \alpha_{ti} h_i \tag{5}$$

Then, c_t obtained by Equation 5 is fed into a fully-connected layer along with the hidden state s_t, followed by a softmax normalization layer to yield a probability distribution $p_t \in \mathbb{R}^{|\mathbb{P}|}$ over sentiment polarity decision space:

$$\tilde{s}_t = tanh(W_c[s_t; c_t]) \tag{6}$$

$$p_t = softmax(W_o \tilde{s}_t) \tag{7}$$

where W_c and W_o are weight parameters. It can be seen that s_t has contained the previous polarity information y_{t-1}. Then according to the automatically captured conjunction information, our model could consider the relationship between the sentiment polarity of aspect a_{t-1} and a_t, and further judge the polarity of current aspect a_t.

However, for aspect-level sentiment classification task, the conventional sequence-to-sequence framework has two disadvantages: i) the model does not have the ability to use earlier sentiment polarity information; ii) the attention mechanism is prone to focus on the context words with strong sentiment polarity repeatedly. Thus, in this paper, we introduce sentiment polarity fusion module and temporal attention mechanism to avoid these two disadvantages.

3.2.3 *Sentiment Polarity Fusion Module*

As described above, we assume that the previously predicted sentiment polarity and the automatically captured conjunction may do a favor in inferring the sentiment polarity of current aspect. Consequently, as shown in Equation 2, we take the concatenation of the representation vector of aspect a_t and the embedding vector of sentiment polarity y_{t-1} as input. However, this operation only guarantees the use of sentiment polarity information of the last time step. As a result, the model is incapable of taking advantage of earlier sentiment polarity information. For example, in a sentence "*Huge portions, great and attentive service, and pretty good prices!*", when predicting the sentiment polarity of aspect *prices*, the model fails to acquire the sentiment polarity information of aspect *portions* because we only input the sentiment polarity of aspect *service*. In fact, according to the conjunction *and*, the relationship between the sentiment polarity of aspect *portions* and *prices* also conduces to predict.

The simplest way to utilize the sentiment polarity of all previous aspects is to concatenate them directly, i.e. the input in Equation 2 is changed from $[e_{a_t}; e_{y_{t-1}}]$ to $[e_{a_t}; e_{y_0}; e_{y_1}; ...; e_{y_{t-1}}]$. However, it is known that these sentiment polarities do not make the same contribution and the concatenation result often contains some common information. Thus, the simply concatenating may hinder the model learning an accurate representation of sentiment polarity and make the process of optimizing the objective function more difficult.

To tackle this problem, this paper proposes a sentiment polarity fusion module which enables the model not only to access the sentiment polarity information of all previous aspects, but also to learn an accurate sentiment polarity representation automatically. This fusion module is illustrated in Figure 2. Here, we maintain a sentiment polarity accumulation vector r_t, which denotes some earlier sentiment polarity information. Note that r_1 is a zero vector. To avoid the disadvantages of concatenation operation, we need to distill the complementary information from r_t. Inspired by the gate mechanism in LSTM, a gate is designed to eliminate the common information in r_t that has already appeared in the sentiment polarity information of the last time step $e_{y_{t-1}}$. Specifically, the gate is designed as:

$$g_t = 1 - \sigma((W_{f_1} r_t) \odot e_{y_{t-1}}) \tag{8}$$

where σ is the sigmoid function, W_{f_1} is weight parameter and $\odot$ denotes the element-wise multiplication. From the definition of g_t, it can be seen that if the values on some specific dimension of $W_{f_1} r_t$ and $e_{y_{t-1}}$ are both large, which indicates the same information appears in both earlier sentiment polarity information r_t and sentiment polarity information of the last time step $e_{y_{t-1}}$, the gate g_t will be closed. So, if $W_{f_1} r_t$ is multiplied to the gate g_t, the information that is only contained in r_t is allowed to pass through. Thus, the complementary information in r_t is eventually computed as:

$$\bar{r}_t = (W_{f_1} r_t) \odot g_t \tag{9}$$

Then, we concatenate the complementary information $\bar{r}_t$ to the sentiment polarity embedding of the last time step $e_{y_{t-1}}$ to produce the final sentiment polarity embedding:

$$\tilde{e}_{y_{t-1}} = W_{f_2}[e_{y_{t-1}}; \bar{r}_t] \tag{10}$$

where W_{f_2} is weight parameter. In this way, given the sentiment polarity accumulation vector r_t and the sentiment polarity embedding vector of the last time step $e_{y_{t-1}}$, we successfully extract the complementary information from r_t and create the final sentiment polarity representation $\tilde{e}_{y_{t-1}}$.

Finally, we use the sentiment polarity information of all previous aspects $\tilde{e}_{y_{t-1}}$ to update r_t and change Equation 2

$$r_{t+1} = \tilde{e}_{y_{t-1}} \tag{11}$$

$$s_t = \text{LSTM}(s_{t-1}, [e_{a_t}; \tilde{e}_{y_{t-1}}]) \tag{12}$$

In this way, the decoder can produce the richer hidden state s_t based on $\tilde{e}_{y_{t-1}}$ including the sentiment polarity information of the last time step as well as earlier sentiment polarity information. Thus, for same conjunction information, the model using the richer hidden state is more capable of judging the sentiment polarity of current aspect accurately than that using the hidden state computed by Equation 2.

3.2.4 *Temporal Attention Mechanism*

When the model predicts the sentiment polarity of different aspects, the attention mechanism focuses on different portions of the sentence. Therefore, the attention mechanism can play a critical role in modeling semantic relatedness between context words and aspect. Actually, this critical role is based on the foundation that the attention mechanism can pay attention on aspect related context words when predicting its sentiment polarity. However, the attention mechanism may lead to the different aspects mistakenly attending to the identical context words as descriptors. Look at a concrete example "*Its size is ideal and the weight is acceptable.*". The word *ideal* is noticed when inferring the sentiment polarity of aspect *size*, which is the same as expected. Nevertheless, owing to the strong sentiment

polarity of word *ideal*, the attention mechanism may identify it as the descriptor of aspect *weight*, which is in fact not the case.

Here, we adapt the temporal attention mechanism to address this issue. We maintain an attention accumulation vector u_t, which is the sum of attention distributions over all previous decoder time steps:

$$u_t = \sum_{t'=1}^{t-1} \alpha_{t'} \tag{13}$$

Intuitively, u_t is a unnormalized distribution over the context words that represents the degree of attention that those words have received from the attention mechanism so far.

The accumulation vector u_t is used to change Equation 4:

$$\alpha_{ti} = \frac{exp(\frac{e_{ti}}{u_{ti}})}{\sum_{j=1}^{n'} exp(\frac{e_{tj}}{u_{tj}})} \tag{14}$$

Obviously, if u_{ti} is large, which means that i-th word has gained a lot of attention, α_{ti} will be small at time-step t. Therefore, this modification should make it easy for the attention mechanism to avoid repeatedly attending to the same locations (i.e. certain words with strong sentiment polarity) when modeling semantic relatedness between context words and aspect. Note that u_1 is a one vector, because at the first time step, we don't need to modify the attention distribution.

3.3 Loss Function

The loss function is the cross entropy loss of sentiment polarity:

$$L = -\sum_{t=1}^{m} y_t log p_t \tag{15}$$

where y_t is the ground truth of sentiment polarity of aspect a_t, and p_t is probability distribution of sentiment polarity of a_t, computed by Equation 7.

4 Experiments

4.1 Datasets

To evaluate our proposed model, we conduct experiments on five datasets: one (Twitter) is originally built by Dong et al. [4] containing twitter posts, while the other four (Lap14, Rest14, Rest15, Rest16) are respectively from SemEval 2014 task 4 [17], SemEval 2015 task 12 [16] and SemEval 2016 task 5 [15], consisting of data from two categories, i.e. laptop and restaurant. Each sample contains a list of aspects and corresponding sentiment polarities, which are labeled with {*positive, negative, neutral*}. It is worth noting that some original datasets contain the fourth sentiment polarity - *conflict*, which means that a sample expresses both *positive* and *negative* towards an aspect. Here, following the previous work [21], [24], we remove samples (which are difficult to model) with *conflict* polarity or without explicit aspects. The statistics of datasets are given in Table 1.

4.2 Experimental Details

We implement our experiments base on BERT's code[8] and use uncased pretrained model to initialize our WordPiece embeddings (dimensions is 768) and encoder's parameters. The hyperparameters (e.g. word pieces vocabulary size, hidden size of encoder and learning rate) and optimizer of our model are the same as that of BERT. In addition, the hidden size of the decoder is 768, and the number of LSTM layers is 3. We use dropout [19] to prevent our networks from overfitting. For the maximum length of input sentence after WordPiece tokenization, we set it to 256. That is, the input sentence longer than this setting will be truncated, and the zero padding is used if the input sentence shorter than this setting. Following the previous work, the accuracy and macro-F1 scores are adopted as the evaluation metrics.

[8] https://github.com/huggingface/transformers

Table 1. The statistics of datasets. #Pos, #Neg and #Neu denotes the number of samples with *Positive*, *Negative* and *Neutral* sentiment polarity, respectively. #1, #2/3, #4/5, #6 denotes the number of samples with one, two or three, four or five and more than six aspects, respectively.

Dataset		#Pos	#Neg	#Neu	#1	#2/3	#4/5	#6
Twitter	Train	1561	1560	3127	6248	0	0	0
	Test	173	173	346	692	0	0	0
Lap14	Train	994	870	464	956	485	49	9
	Test	341	128	169	269	133	15	1
Rest14	Train	2164	807	637	1063	803	133	19
	Test	728	196	196	302	269	40	4
Rest15	Train	912	256	36	601	226	23	0
	Test	326	182	34	305	95	6	1
Rest16	Train	1240	439	69	904	323	29	1
	Test	469	117	30	314	104	12	1

4.3 Baselines

We compare our proposed model with the following baselines:

- SVM [8] is a basic baseline model, which has won SemEval 2014 task 4 with conventional feature extraction methods.
- LSTM [20] uses the last hidden vector of LSTM to predict sentiment polarity.
- MemNet [21] selects more abstractive evidences from the external memory and applies the output of the last attention layer for prediction.
- AOA [7] employs the idea of attention-over-attention to solve aspect-level sentiment classification.
- IAN [13] first models the sentence and aspect respectively, then concatenates the context representation and aspect representation for predicting the sentiment polarity of aspect.
- TNet-LF [10] puts forward Context-Preserving Transformation to preserve and strengthen the informative part of contexts.
- ASGCN-DT [24] builds Graph Convolutional Network over the dependency tree of sentence to exploit syntactical information and word dependencies. Note that ASGCN-DG is the variant of ASGCN-DT replacing dependency tree with dependency graph.

Besides, we design two BERT based baselines:

- BERT-SEP packs a sentence and an aspect together into a single sequence. Here, we separate them with a special symbol *SEP* and use the final hidden state corresponding to the classification symbol *CLS* (i.e. h_{CLS}) for prediction.
- BERT-ATT adds the attention mechanism over final hidden state sequence (i.e. $\mathbf{h}$) where an aspect is treated as query, and uses the output of attention operation for classification.

It is noteworthy that these two baselines have the same experimental setting as our model for fair comparison.

Table 2. Comparison with baselines. The best two results with each dataset are in bold. The results of baselines with † are retrieved from the corresponding papers, ‡ denotes some results are retrieved from Zhang et al. , and the results with # are retrieved from Dong et al. . Accuracy and macro-F1 scores are the average value over 2 runs with random initialization.

Model	Twitter		Lap14		Rest14		Rest15		Rest16	
	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1	Accuracy	F1
SVM†	63.40	63.30	70.49#	N/A	80.16#	N/A	N/A	N/A	N/A	N/A
LSTM†‡	69.56	67.70	69.28	63.09	78.13	67.47	77.37	55.17	86.80	63.88
MemNet†‡	71.48	69.90	72.37	65.17	80.95	69.64	77.31	58.28	85.44	65.99
AOA†‡	72.30	70.20	74.50	67.52	81.20	70.42	78.17	57.02	87.50	66.21
IAN†‡	72.50	70.81	72.10	67.38	78.60	70.09	78.54	52.65	84.74	55.21
TNet-LF†‡	**74.68**	73.36	76.01	71.47	80.79	70.84	78.47	59.47	**89.07**	**70.43**
ASGCN-DT†	71.53	69.68	74.14	69.24	80.86	**72.19**	79.34	60.78	88.69	66.64
ASGCN-DG†	72.15	70.40	75.55	71.05	80.77	72.02	79.89	61.89	88.99	67.48
BERT-SEP	72.54	71.40	76.89	72.11	**82.32**	70.19	80.62	60.41	88.93	70.42
BERT-ATT	**74.71**	**73.72**	**76.95**	**72.54**	80.98	71.17	**81.01**	**62.65**	88.47	70.31
SPM	74.56	**73.39**	**78.68**	**73.49**	**82.69**	**72.86**	**81.18**	**64.14**	**89.36**	**73.32**

4.4 Results

For simplicity, we denote the proposed **s**equence **p**rediction **m**odel as SPM. Table 2 shows the performance of our model and baseline models on five test sets. We can observe that SPM consistently outperforms all compared models[9] on all datasets except Twitter. The results illustrate the superiority of SPM and the significance of relationship between the sentiment polarity of aspects. For Twitter, SPM only achieves comparable results compared with baseline TNet-LF. This is because each sample in Twitter dataset only has an aspect (see Table 1), restricting the efficacy. Meanwhile, we can find that BERT-SEP and BERT-ATT are strong performers and substantially surpass than all other baseline models on Twitter, Lap14 and Rest15 datasets. A possible reason is that BERT can build the powerful representation for the sentence and is more efficient to model the semantic relationship between sentence and aspect in the aspect-level sentiment classification. In addition, LSTM method gets bad performance because it can not take full advantage of the contextual semantic information of sentence. Thus, it is obvious that the modeling capabilities of LSTM and BERT are quite different. This is why we choose BERT as the encoder in this work. For BERT-SEP and BERT-ATT baselines, the performance is not easy to distinguish. For instance, although BERT-ATT is better than BERT-SEP on Twitter, Lap14 and Rest15 dataset, BERT-SEP performs more competitive than BERT-ATT on Rest16 dataset. Thus, we assume the multi-layer and multi-head attention in BERT can strongly construct the contextual representations on datasets. Additionally, SPM performs better than BERT-SEP and BERT-ATT on all datasets except Twitter. For example, SPM achieves an improvement of 3.73 points and 1.49 points over BERT-SEP and BERT-ATT in term of F1 score on Rest15 dataset respectively. This demonstrates that the improvements of our model not only are from the help of BERT, but also benefit from the temporal attention mechanism and the sentiment polarity fusion module.

4.5 Ablation Study

To demonstrate the effectiveness of temporal attention mechanism and sentiment polarity fusion module, four variants of SPM are evaluated: (1) SPM(w/o TS): SPM without both **t**emporal attention mechanism and **s**entiment polarity fusion module, i.e. SPM uses Equation 2 and 4 instead of Equation 12 and 14. (2) SPM(w/o S): SPM without **s**entiment polarity fusion process where SPM updates the state of the decoder with Equation 2. (3) SPM(w/o T): SPM without **t**emporal attention mechanism where SPM utilizes Equation 4 to compute the attention distribution. (4) SPM(w/o I): SPM completely removes the sentiment polarity **i**nformation where the input in Equation 2 is changed from $[e_{a_t}; e_{y_{t-1}}]$ to $[e_{a_t}]$. Four variants are compared with SPM on all datasets except Twitter. The results are shown in Table 3. According to the experimental results, we can come to the following conclusions. First, removal of sentiment polarity fusion module (i.e. SPM(w/o S)) leads to a slight performance degradation on four datasets. Thus, we conclude that the use of sentiment polarity information of all previous aspects provides a benefit to performance. Moreover, after we get rid of the sentiment polarity information, SPM(w/o I) could not keep as competitive as SPM(w/o S) on all metrics except accuracy on Rest16 dataset. This verifies the significance of the sentiment polarity information. Second, when we remove the temporal attention mechanism, SPM(w/o T) is slightly inferior to SPM on four datasets. One underlying reason is that SPM(w/o T) is prone to overly focus on the context words with strong sentiment polarity when predicting the sentiment polarity of different aspects. Thus it could be concluded that the temporal attention mechanism contributes to SPM since it prevents SPM from repeatedly attending to the same context words. Finally, when we clear up both components at the same time, compared with SPM(w/o S) and SPM(w/o T), SPM(w/o TS) is much less powerful on four datasets. These results again strongly demonstrate the usefulness of temporal attention mechanism and sentiment polarity fusion mod-

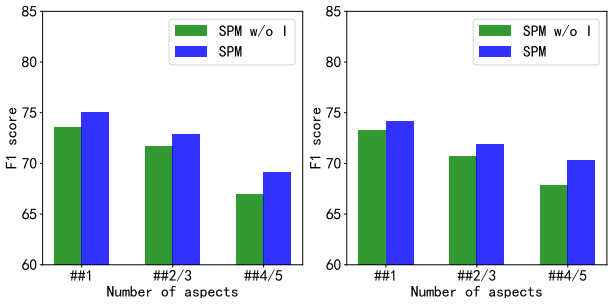
Table 3. Ablation study of SPM. Accuracy and macro-F1 scores are the average value over 2 runs with random initialization.

Dataset		Variants				SPM
		w/o TS	w/o S	w/o T	w/o I	
Lap14	Accuracy	76.17	77.89	78.21	76.80	78.68
	F1	70.96	72.56	72.81	71.57	73.49
Rest14	Accuracy	81.03	81.11	81.38	80.66	82.69
	F1	71.70	71.90	72.70	70.50	72.86
Rest15	Accuracy	79.70	80.07	80.99	79.88	81.18
	F1	62.16	63.87	63.75	62.51	64.14
Rest16	Accuracy	88.07	88.21	89.10	88.27	89.36
	F1	71.13	72.63	72.83	72.19	73.32

[9] All baseline models formulate the aspect-level sentiment classification as predicting the sentiment polarity of a provided (sentence, aspect) pair.

Table 4. Case study. The color depth indicates the importance degree of weight, the darker the more important. The marker √ indicates correct prediction.

Model	Attention Visualization	Aspect	Prediction
BERT-ATT	great food but the service was dreadful !	food	positive√
	great food but the service was dreadful !	service	negative√
	Works well , and I am extremely happy to be back to an apple OS .	works	positive√
	Works well , and I am extremely happy to be back to an apple OS .	apple OS	positive√
SPM	great food but the service was dreadful !	food	positive√
	great food but the service was dreadful !	service	negative√
	Works well , and I am extremely happy to be back to an apple OS .	works	positive√
	Works well , and I am extremely happy to be back to an apple OS .	apple OS	positive√



(a) On Lap14 test set. (b) On Rest14 test set.

Figure 3. Macro-F1 score (the value of one run with random initialization) versus the number of aspects within sentence.

ule. Besides, the performance gap between SPM(w/o S) and SPM is larger than that between SPM(w/o T) and SPM on all metrics except F1 score on Rest15 dataset. This phenomenon reveals the level of benefit brought by the sentiment polarity fusion module is greater than that brought by the temporal attention mechanism. Therefore, according to these experimental results, we could prove the contributions of both components in the aspect-level sentiment classification.

4.6 Effects of Multiple Aspects

In this paper, we leverage the relationship between the sentiment polarity of aspects to accomplish the aspect-level sentiment classification. Therefore, the number of aspects may affect the model performance. Now, we discuss this impact. Here, we select SPM(w/o I) (does not include sentiment polarity information) and SPM as our research objects. We divide the test samples in Lap14 and Rest14 datasets into three groups (##1, ##2/3 and ##4/5) (see Table 1) based on the number of aspects within sentence and compute F1 score differences between these groups. It can be seen in Figure 3 that when the number of aspects within sentence increases, F1 score decreases. It indicates that the more aspects the sentence contains, the more difficult it is to predict the sentiment polarities correctly. This is in line with the assumption that the more aspects, the more difficult it is to model the semantic relatedness between sentence and aspect accurately. However, as shown in Figure 3, the magnitude of both models' performance decline is significantly different. For example, when the test set on Lap14 dataset changes from group ##2/3 to group ##4/5, SPM(w/o I) drops from 71.63 to 66.95, while SPM decreases from 72.88 to 69.13. This shows that the sentiment polarity information could slow down the performance decline to some extent when there is more than one aspect in a sentence.

4.7 Case Study

To have an intuitive understanding of SPM, we present the case study with two testing examples. Here, we visualize the attention scores offered by BERT-ATT and SPM in Table 4, along with their predictions. In the first example, we can find that BERT-ATT and SPM give more attention to the context words *great* and *food* when the current aspect is *food*. This is as we would expect, since the word *great* contributes to prediction. Although BERT-ATT and SPM pay more attention on the word *dreadful* when faced with another aspect *service*, the weight change of word *great* is significantly different. It can be observed from Table 4 that BERT-ATT basically maintains the weight of *great* due to the strong sentiment polarity of the word itself, while SPM greatly reduces that of *great*. Obviously, the latter is more acceptable and desirable. This shows the effects of temporal attention mechanism which can avoid repeatedly attending to the context words with strong sentiment polarity. Furthermore, SPM can automatically capture the conjunction which reflects the relationship between the sentiment polarity of aspects through training. For example, when the aspect is *service*, clearly, the conjunction *but* gets more attention, and reveals the relationship *opposite* between the sentiment polarity of aspect *food* and *service*, which may help judge the sentiment polarity of *service*. This shows the effectiveness of considering the previous sentiment polarity, which allows SPM to classify polarity using the relationship between the sentiment polarity of aspects. Additionally, it is worth noting that SPM is more capable of identifying the descriptor of aspect than BERT-ATT. For example, when predicting the polarity of *food*, SPM assigns a greater weight to word *great* than to word *food*, while BERT-ATT does the opposite. And, in the second example, although BERT-ATT predicts the correct polarity label towards aspect *works*, BERT-ATT actually pays attention to word *happy*. This is totally inconsistent with the fact. We can draw similar conclusions from the second example. Due to limited space, we do not analyze it in detail.

5 Conclusion

In this paper, we propose a sequence prediction model (SPM) based on sequence-to-sequence framework for aspect-level sentiment classification. The main idea of SPM is to leverage the relationship between the sentiment polarity of aspects. To use earlier sentiment polarity information, SPM integrates a sentiment polarity fusion module. Moreover, SPM adopts the temporal attention mechanism, which discourages repeated attention to the context words with strong sentiment polarity when predicting the sentiment polarity of different aspects. Experimental results on five benchmarking datasets demonstrate that SPM obtains superior performance over baseline models.

ACKNOWLEDGEMENTS

This research is partially supported by the National Natural Science Foundation of China (No. U1811462 and U1711263).

References

[1] Caroline Brun, Diana Nicoleta Popa, and Claude Roux, 'Xrce: Hybrid classification for aspect-based sentiment analysis', in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 838–842, (2014).

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

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova, 'Bert: Pre-training of deep bidirectional transformers for language understanding', in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, (2019).

[4] Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu, 'Adaptive recursive neural network for target-dependent twitter sentiment classification', in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (volume 2: Short papers)*, pp. 49–54, (2014).

[5] Shuqin Gu, Lipeng Zhang, Yuexian Hou, and Yin Song, 'A position-aware bidirectional attention network for aspect-level sentiment analysis', in *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 774–784, (2018).

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

[7] Binxuan Huang, Yanglan Ou, and Kathleen M Carley, 'Aspect level sentiment classification with attention-over-attention neural networks', in *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pp. 197–206. Springer, (2018).

[8] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad, 'Nrc-canada-2014: Detecting aspects and sentiment in customer reviews', in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 437–442, (2014).

[9] Himabindu Lakkaraju, Richard Socher, and Chris Manning, 'Aspect specific sentiment analysis using hierarchical deep learning', in *NIPS Workshop on Deep Learning and Representation Learning*, (2014).

[10] Xin Li, Lidong Bing, Wai Lam, and Bei Shi, 'Transformation networks for target-oriented sentiment classification', in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 946–956, (2018).

[11] Thang Luong, Hieu Pham, and Christopher D Manning, 'Effective approaches to attention-based neural machine translation', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, (2015).

[12] Dehong Ma, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang, 'Exploring sequence-to-sequence learning in aspect term extraction', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3538–3547, (2019).

[13] Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang, 'Interactive attention networks for aspect-level sentiment classification', in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 4068–4074. AAAI Press, (2017).

[14] Thien Hai Nguyen and Kiyoaki Shirai, 'Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis', in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2509–2514, (2015).

[15] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, AL-Smadi Mohammad, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al., 'Semeval-2016 task 5: Aspect based sentiment analysis', in *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pp. 19–30, (2016).

[16] Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos, 'Semeval-2015 task 12: Aspect based sentiment analysis', in *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pp. 486–495, (2015).

[17] Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar, 'Semeval-2014 task 4: Aspect based sentiment analysis', in *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pp. 27–35, (2014).

[18] Abigail See, Peter J Liu, and Christopher D Manning, 'Get to the point: Summarization with pointer-generator networks', in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1073–1083, (2017).

[19] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, 'Dropout: a simple way to prevent neural networks from overfitting', *The Journal of Machine Learning Research*, **15**(1), 1929–1958, (2014).

[20] Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu, 'Effective lstms for target-dependent sentiment classification', in *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 3298–3307, (2016).

[21] Duyu Tang, Bing Qin, and Ting Liu, 'Aspect level sentiment classification with deep memory network', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 214–224, (2016).

[22] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao, 'Attention-based lstm for aspect-level sentiment classification', in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pp. 606–615, (2016).

[23] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al., 'Google's neural machine translation system: Bridging the gap between human and machine translation', *ArXiv Preprint ArXiv:1609.08144*, (2016).

[24] Chen Zhang, Qiuchi Li, and Dawei Song, 'Aspect-based sentiment classification with aspect-specific graph convolutional networks', in *2019 Conference on Empirical Methods in Natural Language Processing*, (2019).