CAN: Constrained Attention Networks for Multi-Aspect Sentiment Analysis

Mengting Hu\textsuperscript{1,}\footnote{This work was done when Mengting Hu was a research intern at IBM Research - China.} Shiwan Zhao\textsuperscript{2,}\footnote{Corresponding author.} Li Zhang\textsuperscript{2} Keke Cai\textsuperscript{2} Zhong Su\textsuperscript{2} Renhong Cheng\textsuperscript{1} Xiaowei Shen\textsuperscript{2}
\textsuperscript{1} Nankai University \textsuperscript{2} IBM Research - China
mthu@mail.nankai.edu.cn, \{zhaosw, lizhang, caikeke, suzhong\}@cn.ibm.com
chengrh@nankai.edu.cn, xwshen@cn.ibm.com

Abstract
Aspect level sentiment classification is a fine-grained sentiment analysis task. To detect the sentiment towards a particular aspect in a sentence, previous studies have developed various attention-based methods for generating aspect-specific sentence representations. However, the attention may inherently introduce noise and downgrade the performance. In this paper, we propose constrained attention networks (CAN), a simple yet effective solution, to regularize the attention for multi-aspect sentiment analysis, which alleviates the drawback of the attention mechanism. Specifically, we introduce orthogonal regularization on multiple aspects and sparse regularization on each single aspect. Experimental results on two public datasets demonstrate the effectiveness of our approach. We further extend our approach to multi-task settings and outperform the state-of-the-art methods.

1 Introduction
Sentiment analysis (Nasukawa and Yi, 2003; Liu, 2012), an important task in natural language understanding, receives much attention in recent years. Aspect level sentiment classification (ALSC) is a fine-grained sentiment analysis task, which aims at detecting the sentiment towards a particular aspect in a sentence. ALSC is especially critical for multi-aspect sentences which contain multiple aspects. A multi-aspect sentence can be categorized as overlapping or non-overlapping. A sentence is annotated as non-overlapping if any two of its aspects have no overlap. Our study found that around 85% of the multi-aspect sentences are non-overlapping in the two public datasets. Figure 1 shows a simple example. The non-overlapping sentence contains two aspects. The aspect food is on the left side of the aspect service. Their distributions on words are orthogonal to each other. Another observation is that only a few words relate to the opinion expression in each aspect. As shown in Figure 1, only the word “good” is relevant to the aspect food and “ok” to service. The distribution of the opinion expression of each aspect is sparse.

To detect the sentiment towards a particular aspect, previous studies (Wang et al., 2016; Ma et al., 2017; Cheng et al., 2017; Ma et al., 2018; Huang et al., 2018) have developed various attention-based methods for generating aspect-specific sentence representations. In these works, the attention may inherently introduce noise and downgrade the performance (Li et al., 2018) since the attention scatters across the whole sentence and is prone to attend on noisy words, or the opinion words from other aspects. Take Figure 1 as an example, for the aspect food, we visualize the attention weights from the model (Wang et al., 2016). Much of the attention focuses on the noisy word “sometimes”, and the opinion word “ok” which is relevant to the aspect service rather than food.

To alleviate the above issue, we propose a model for multi-aspect sentiment analysis, which regularizes the attention by handling multiple aspects of a sentence simultaneously. Specifically, we introduce orthogonal regularization for attention weights among multiple non-overlapping as-
pects. The orthogonal regularization tends to make the attention weights of multiple aspects concentrate on different parts of the sentence with less overlap. We also introduce the sparse regularization, which tends to make the attention weights of each aspect concentrate only on a few words. We call our networks with such regularizations constrained attention networks (CAN).

There have been some works on introducing sparsity in attention weights in machine translation (Malaviya et al., 2018) and orthogonal constraints in domain adaptation (Bousmalis et al., 2016). In this paper, we add both sparse and orthogonal regularizations in a unified form inspired by the work (Lin et al., 2017). The details will be introduced in Section 3.

In addition to aspect level sentiment classification (ALSC), aspect category detection (ACD) is another task of aspect based sentiment analysis. ACD (Zhou et al., 2015; Schouten et al., 2018) aims to identify the aspect categories discussed in a given sentence from a predefined set of aspect categories (e.g., price, food, service). Take Figure 1 as an example, aspect categories food and service are mentioned. We introduce ACD as an auxiliary task to assist the ALSC task, benefiting from the shared context of the two tasks. We also apply our attention constraints to the ACD task. By applying attention weight constraints to both ALSC and ACD tasks in an end-to-end network, we can further evaluate the effectiveness of CAN in multi-task settings.

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

• We propose CAN for multi-aspect sentiment analysis. Specifically, we introduce orthogonal and sparse regularizations to constrain the attention weight allocation, helping learn better aspect-specific sentence representations.

• We extend CAN to multi-task settings by introducing ACD as an auxiliary task, and applying CAN on both ALSC and ACD tasks.

• Extensive experiments are conducted on public datasets. Results demonstrate the effectiveness of our approach for aspect level sentiment classification.

2 Related Work

Aspect level sentiment analysis is a fine-grained sentiment analysis task. Earlier methods are usually based on explicit features (Liu et al., 2010; Vo and Zhang, 2015). With the development of deep learning technologies, various neural attention mechanisms have been proposed to solve this fine-grained task (Wang et al., 2016; Ruder et al., 2016; Ma et al., 2017; Tay et al., 2017; Cheng et al., 2017; Chen et al., 2017; Tay et al., 2018; Ma et al., 2018; Wang and Lu, 2018; Wang et al., 2018). To name a few, Wang et al. (2016) propose an attention-based LSTM network for aspect level sentiment classification. Ma et al. (2017) use the interactive attention networks to generate the representations for targets and contexts separately. Cheng et al. (2017); Ruder et al. (2016) both propose hierarchical neural network models for aspect level sentiment classification. Wang and Lu (2018) employ a segmentation attention based LSTM model for aspect level sentiment classification. All these works can be categorized as single-aspect sentiment analysis, which deals with aspects in a sentence separately, without considering the relationship between aspects.

More recently, a few works have been proposed to take the relationship among multiple aspects into consideration. Hazarika et al. (2018) make simultaneous classification of all aspects in a sentence using recurrent networks. Majumder et al. (2018) employ memory network to model the dependency of the target aspect with the other aspects in the sentence. Fan et al. (2018) design an aspect alignment loss to enhance the difference of the attention weights towards the aspects which have the same context and different sentiment polarities. In this paper, we introduce orthogonal regularization to constrain the attention weights of multiple non-overlapping aspects, as well as sparse regularization on each single aspect.

Multi-task learning Caruana (1997) solves multiple learning tasks at the same time, achieving improved performance by exploiting commonalities and differences across tasks. Multi-task learning has been used successfully in many machine learning applications. Huang and Zhong (2018) learn both main task and auxiliary task jointly with shared representations, achieving improved performance in question answering. Toshniwal et al. (2017) use low-level auxiliary tasks for encoder-decoder based speech recognition, which suggests that the addition of auxiliary tasks can help in either optimization or generalization. Yu and Jiang (2016) use two auxiliary tasks to help induce a
sentence embedding that works well across domains for sentiment classification. In this paper, we adopt the multi-task learning approach by using ACD as the auxiliary task to help the ALSC task.

3 Model

We first formulate the problem. There are totally $N$ predefined aspect categories in the dataset, $A = \{A_1, \ldots, A_N\}$. Given a sentence $S = \{w_1, w_2, \ldots, w_L\}$, which contains $K$ aspects $A^s = \{A^s_1, \ldots, A^s_K\}$, $A^s \subseteq A$, the multi-task learning is to simultaneously solve the ALSC and ACD tasks, namely, the ALSC task predicts the sentiment polarity of each aspect $A^s_k \in A^s$, and the auxiliary ACD task checks each aspect $A_n \in A$ to see whether the sentence $S$ mentions it. Note that we only focus on aspect-category instead of aspect-term (Xue and Li, 2018) in this paper.

We propose CAN for multi-aspect sentiment analysis, supporting both ALSC and ACD tasks by a multi-task learning framework. The network architecture is shown in Figure 2. We will introduce all components sequentially from left to right.

3.1 Embedding and LSTM Layers

Traditional single-aspect sentiment analysis handles each aspect separately, one at a time. In such settings, a sentence $S$ with $K$ aspects will be copied to form $K$ instances. For example, a sentence $S$ contains two aspects: $A^s_1$ with polarity $p_1$ and $A^s_2$ with polarity $p_2$. Two instances, $(S, A^s_1, p_1)$ and $(S, A^s_2, p_2)$, will be constructed.

Our multi-aspect sentiment analysis method handles multiple aspects together and takes the single instance $(S, [A^s_1, A^s_2], [p_1, p_2])$ as input.

The input sentence $\{w_1, w_2, \ldots, w_L\}$ is first converted to a sequence of vectors $\{v_1, v_2, \ldots, v_L\}$, and the $K$ aspects of the sentence are transformed to vectors $\{u^s_1, \ldots, u^s_K\}$, which is a subset of $\{v_1, \ldots, v_N\}$, the vectors of all aspect categories. The word embeddings of the sentence are then fed into an LSTM network (Hochreiter and Schmidhuber, 1997), which outputs hidden states $H = \{h_1, h_2, \ldots, h_L\}$. The sizes of the embedding and the hidden state are both set to be $d$.

3.2 Task-Specific Attention Layer

The ALSC and ACD tasks share the hidden states from the LSTM layer, while compute their own attention weights separately. The attention weights are then used to compute aspect-specific sentence representations.

**ALSC Attention Layer** The key idea of aspect level sentiment classification is to learn different attention weights for different aspects, so that different aspects can concentrate on different parts of the sentence. We follow the approach in the work (Bahdanau et al., 2015) to compute the attention. Particularly, given the sentence $S$ with $K$ aspects, $A^s = \{A^s_1, \ldots, A^s_K\}$, for each aspect $A^s_k$, its attention weights are calculated by:

$$\alpha_k = \text{softmax}(z^a_k \text{tanh}(W^a_k h + W^b_k (u^k_k \otimes e_L)))$$  \hspace{1cm} (1)

where $u^k_k$ is the embedding of the aspect $A^s_k$, $e_L \in \mathbb{R}^L$ is a vector of 1s, $u^k_k \otimes e_L$ is the op-
eration repeatedly concatenating $u_k^b$ for $L$ times. $W_1^a \in \mathbb{R}^{d \times d}$, $W_2^b \in \mathbb{R}^{d \times d}$ and $z^a \in \mathbb{R}^d$ are the weight matrices.

**ACD Attention Layer** We treat the ACD task as multi-label classification problem for the set of $N$ aspect categories. For each aspect $A_n \in A$, its attention weights are calculated by:

$$\beta_n = \text{softmax}(z^b \tanh(W_1^b H + W_2^b (u_n \otimes e_L)))$$  \hspace{1cm} (2)

where $u_n$ is the embedding of the aspect $A_n$, $W_1^b \in \mathbb{R}^{d \times d}$, $W_2^b \in \mathbb{R}^{d \times d}$ and $z^b \in \mathbb{R}^d$ are the weight matrices.

The ALSC and ACD tasks use the same attention mechanism, but they do not share parameters. The reason to use separated parameters is that, for the same aspect, the attention of ALSC concentrates more on opinion words, while ACD focuses more on aspect target terms (see the attention visualizations in Section 4.6).

### 3.3 Regularization Layer

We simultaneously handles multiple aspects by adding constraints to their attention weights. **Note that this layer is only available in the training stage**, in which the ground-truth aspects are known for calculating the regularization loss, and then influence parameter updating in back propagation. While in the testing/inference stage, the true aspects are unknown and the regularization loss is not calculated so that this layer is omitted from the architecture.

In this paper, we introduce two types of regularizations: the sparse regularization on each single aspect; the orthogonal regularization on multiple non-overlapping aspects.

**Sparse Regularization** For each aspect, the sparse regularization constrains the distribution of the attention weights ($\alpha_k$ or $\beta_n$) to concentrate on less words. For simplicity, we use $\alpha_k$ as an example, $\alpha_k = \{\alpha_{k1}, \alpha_{k2}, \ldots, \alpha_{KL}\}$. To make $\alpha_k$ sparse, the sparse regularization term is defined as:

$$R_s = \sum_{l=1}^{L} \alpha_{kl}^2 - 1 \rightleftharpoons$$

where $\sum_{l=1}^{L} \alpha_{kl} = 1$ and $\alpha_{kl} > 0$. Since $\alpha_k$ is normalized as a probability distribution, $L_1$ norm is always equal to 1 (the sum of the probabilities) and does not work as sparse regularization as usual. Minimizing Equation 3 will force the sparsity of $\alpha_k$. It has the similar effect as minimizing the entropy of $\alpha_k$, which leads to placing more probabilities on less words.

**Orthogonal Regularization** This regularization term forces orthogonality among attention weight vectors of multiple aspects, so that different aspects attend on different parts of the sentence with less overlap. Note that we only apply this regularization to non-overlapping multi-aspect sentences. Assume that the sentence $S$ contains $K$ non-overlapping aspects $\{A_1, \ldots, A_K\}$ and their attention weight vectors are $\{\alpha_1, \ldots, \alpha_K\}$. We pack them together as a two-dimensional attention matrix $M \in \mathbb{R}^{K \times L}$ to calculate the orthogonal regularization term.

$$R_o = \| M^T M - I \|_2$$  \hspace{1cm} (4)

where $I$ is an identity matrix. In the resulted matrix of $M^T M$, each non-diagonal element is the dot product between two attention weight vectors, minimizing the non-diagonal elements will force orthogonality between corresponding attention weight vectors. The diagonal elements of $M^T M$ are subtracted by 1, which are the same as $R_s$ defined in Equation 3. As a whole, $R_o$ includes both sparse and orthogonal regularization terms.

Note that in the ACD task, we do not pack all the $N$ attention vectors $\{\beta_1, \ldots, \beta_N\}$ as a matrix. The sentence $S$ contains $K$ aspects. For simplicity, let $\{\beta_1, \ldots, \beta_K\}$ be the attention vectors of the $K$ aspects mentioned, while $\{\beta_{K+1}, \ldots, \beta_N\}$ be the attention vectors of the $N-K$ aspects not mentioned. We compute the average of the $N-K$ attention vectors, denoted by $\beta_{avg}$. We then construct the attention matrix $G = \{\beta_1, \ldots, \beta_K, \beta_{avg}\}$. $G \in \mathbb{R}^{(K+1) \times L}$. The reason why we calculate $\beta_{avg}$ is that if an aspect is not mentioned in the sentence, its attention weights often attend to meaningless stop words, such as “to”, “the”, “was”, etc. We do not need to distinguish among the $N-K$ aspects not mentioned, therefore they can share stop words in the sentence by being averaged as a whole, which keeps the $K$ aspects mentioned away from such stop words.

### 3.4 Task-Specific Prediction Layer

Given the attention weights of each aspect, we can generate aspect-specific sentence representation, and then make prediction for the ALSC and ACD tasks respectively.

**ALSC Prediction** The weighted hidden state is combined with the last hidden state to generate the
initial aspect-specific sentence representation.

\[ r_k^a = \tanh(W_1^a \bar{h}_k + W_2^a h_L) \]  

(5)

where \( W_1^a \in \mathbb{R}^{d \times d} \) and \( W_2^a \in \mathbb{R}^{d \times d} \). \( \bar{h}_k = \sum_{l=1}^L \alpha_{kl} h_l \) is the weighted hidden state for aspect \( k \). \( r_k^a \) is then used to make sentiment polarity prediction.

\[ y_k = \text{softmax}(W_p^a r_k^a + b_p^a) \]  

(6)

where \( W_p^a \in \mathbb{R}^{d \times c} \) and \( b_p^a \in \mathbb{R}^c \) are the parameters of the projection layer, and \( c \) is the number of classes.

For the sentence \( S \) with \( K \) aspects mentioned, we make \( K \) predictions simultaneously. That is why we call our approach multi-aspect sentiment analysis.

**ACD Prediction** We directly use the weighted hidden state as the sentence representation for ACD prediction.

\[ r_n = \bar{h}_n = \sum_{l=1}^L \beta_{nl} h_l \]  

(7)

We do not combine with the last hidden state \( h_L \) since the aspect may not be mentioned by the sentence. We make \( N \) predictions for all predefined aspect categories.

\[ y_n = \text{sigmoid}(W_p^b r_n + b_p^b) \]  

(8)

where \( W_p^b \in \mathbb{R}^{d \times 1} \) and \( b_p^b \) is a scalar.

### 3.5 Loss

For the task ALSC, the loss function for the \( K \) aspects of the sentence \( S \) is defined by:

\[ L_a = -\sum_{k=1}^K \sum_{c=1}^c y_{kc} \log \hat{y}_{kc} \]

where \( c \) is the number of classes. For the task ACD, as each prediction is binary classification problem, the loss function for the \( N \) aspects of the sentence \( S \) is defined by:

\[ L_b = -\sum_{n=1}^N [y_n \log \hat{y}_n + (1 - y_n) \log(1 - \hat{y}_n)] \]

We jointly train our model for the two tasks. The parameters in our model are then trained by minimizing the combined loss function:

\[ L = L_a + \frac{1}{N} L_b + \lambda \beta \]  

(9)

where \( R \) is the regularization term mentioned previously, which can be \( R_a \) or \( R_o \). \( \lambda \) is the hyper-parameter used for tuning the impact from regularization loss to the overall loss. To avoid \( L_b \) overwhelming the overall loss, we divide it by the number of aspect categories.

### 4 Experiments

#### 4.1 Datasets

We conduct experiments on two public datasets from SemEval 2014 task 4 (Pontiki et al., 2014) and SemEval 2015 task 12 (Pontiki et al., 2015) (denoted by Rest14 and Rest15 respectively). These two datasets consist of restaurant customer reviews with annotations identifying the mentioned aspects and the sentiment polarity of each aspect. To apply orthogonal regularization, we manually annotate the multi-aspect sentences with overlapping or non-overlapping\(^1\). We randomly split the original training set into training, validation sets in the ratio 5:1, where the validation set is used to select the best model. We count the sentences of single-aspect and multi-aspect separately. Detailed statistics are summarized in Table 1. Particularly, 85.23% and 83.73% of the multi-aspect sentences are non-overlapping in Rest14 and Rest15, respectively.

\begin{table}[h]
\centering
\begin{tabular}{|c|c|c|c|c|}
\hline
Dataset & \#Single & \#Multi & \#Total & \#Total \\
\hline
Rest14_Train & 2053 & 67 & 415 & 482 & 2535 \\
Rest14_Val & 412 & 19 & 75 & 94 & 506 \\
Rest14_Test & 611 & 27 & 162 & 189 & 800 \\
Rest15_Train & 622 & 47 & 262 & 309 & 931 \\
Rest15_Val & 137 & 13 & 39 & 52 & 189 \\
Rest15_Test & 390 & 30 & 162 & 192 & 582 \\
\hline
\end{tabular}
\caption{The numbers of single- and multi-aspect sentences. OL and NOL denote the overlapping and non-overlapping multi-aspect sentences, respectively.}
\end{table}

#### 4.2 Comparison Methods

Since we focus on aspect-category sentiment analysis, many works (Ma et al., 2017; Li et al., 2018; Fan et al., 2018) which focus on aspect-term sentiment analysis are excluded.

- **LSTM**: We implement the vanilla LSTM to model the sentence and use the average of all

\(^1\)We will release the annotated dataset later.
<table>
<thead>
<tr>
<th>Model</th>
<th>Rest14</th>
<th></th>
<th>Rest15</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3-way</td>
<td>Binary</td>
<td>3-way</td>
<td>Binary</td>
</tr>
<tr>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Acc</td>
<td>F1</td>
</tr>
<tr>
<td>LSTM</td>
<td>80.92</td>
<td>68.30</td>
<td>85.83</td>
<td>80.88</td>
</tr>
<tr>
<td>AT-LSTM</td>
<td>81.24</td>
<td>69.19</td>
<td>73.37</td>
<td>51.74</td>
</tr>
<tr>
<td>ATAE-LSTM</td>
<td>82.18</td>
<td>69.18</td>
<td>74.56</td>
<td>51.40</td>
</tr>
<tr>
<td>GCAE</td>
<td>82.08</td>
<td>70.20</td>
<td>76.69</td>
<td>53.00</td>
</tr>
<tr>
<td>AT-CAN-Rs</td>
<td>82.28</td>
<td>70.94</td>
<td>75.62</td>
<td>53.56</td>
</tr>
<tr>
<td>AT-CAN-Ro</td>
<td>82.81</td>
<td>71.32</td>
<td>76.92</td>
<td>55.67</td>
</tr>
<tr>
<td>ATAE-CAN-Rs</td>
<td>82.18</td>
<td>72.19</td>
<td>77.28</td>
<td>52.45</td>
</tr>
<tr>
<td>ATAE-CAN-Ro</td>
<td>82.81</td>
<td>73.23</td>
<td>78.58</td>
<td>54.72</td>
</tr>
</tbody>
</table>

Table 2: Results of the ALSC task in single-task settings in terms of accuracy (%) and Macro-F1 (%).

hidden states as the sentence representation. In this model, aspect information is not used.

- **AT-LSTM** (Wang et al., 2016): It adopts the attention mechanism in LSTM to generate a weighted representation of a sentence. The aspect embedding is used to compute the attention weights as in Equation 1. We do not concatenate the aspect embedding to the hidden state as in the work (Wang et al., 2016) and gain small performance improvement. We use this modified version in all experiments.

- **ATAE-LSTM** (Wang et al., 2016): This method is an extension of AT-LSTM. In this model, the aspect embedding is concatenated to each word embedding of the sentence as the input to the LSTM layer.

- **GCAE** (Xue and Li, 2018): This state-of-the-art method is based on the convolutional neural network with gating mechanisms, which is for both aspect-category and aspect-term sentiment analysis. We compare with its aspect-category sentiment analysis task.

### 4.3 Our Methods

To verify the performance gain of introducing constraints on attention weights, we first create several variants of our model for single-task settings.

- **AT-CAN-Rs**: Add sparse regularization $R_s$ to AT-LSTM to constrain the attention weights of each single aspect.

- **AT-CAN-Ro**: Add orthogonal regularization $R_o$ to AT-CAN-Rs to constrain the attention weights of multiple non-overlapping aspects.

- **ATAE-CAN-Rs**: Add $R_s$ to ATAE-LSTM.

- **ATAE-CAN-Ro**: Add $R_o$ to ATAE-CAN-Rs.

We then extend attention constraints to multi-task settings, creating variants by different options: 1) no constraints, 2) adding regularizations only to the ALSC task, 3) adding regularizations to both tasks.

- **M-AT-LSTM**: This is the basic multi-task model without regularizations.

- **M-CAN-Rs**: Add $R_s$ to the ALSC task in M-AT-LSTM.

- **M-CAN-Ro**: Add $R_o$ to the ALSC task in M-CAN-Rs.

- **M-CAN-2Rs**: Add $R_s$ to both tasks in M-AT-LSTM.

- **M-CAN-2Ro**: Add $R_o$ to both tasks in M-CAN-2Rs.

### 4.4 Implementation Details

We set $\lambda = 0.1$ with the help of the validation set. All models are optimized by the Adagrad optimizer (Duchi et al., 2011) with learning rate 0.01. Batch size is 25. We apply a dropout of $p = 0.7$ after the embedding and LSTM layers. All words in the sentences are initialized with 300 dimension Glove Embeddings (Pennington et al., 2014). The aspect embedding matrix and parameters are initialized by sampling from a uniform distribution $U(-\varepsilon, \varepsilon)$, $\varepsilon = 0.01$. $d$ is set as 300. The models are trained for 100 epochs, during which the model with the best performance on the validation set is saved. We also apply early stopping in training, which means that the training will stop if the performance on validation set does not improve in 10 epochs.
<table>
<thead>
<tr>
<th>Model</th>
<th>Rest14 3-way</th>
<th>Rest14 Binary</th>
<th>Rest15 3-way</th>
<th>Rest15 Binary</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Acc</td>
<td>F1</td>
<td>Acc</td>
<td>F1</td>
</tr>
<tr>
<td>M-AT-LSTM</td>
<td>82.60</td>
<td>71.44</td>
<td>88.55</td>
<td>83.76</td>
</tr>
<tr>
<td>M-CAN-(R_s)</td>
<td>83.65</td>
<td>73.97</td>
<td>89.26</td>
<td>85.43</td>
</tr>
<tr>
<td>M-CAN-(R_o)</td>
<td>83.12</td>
<td>72.29</td>
<td>89.61</td>
<td>85.18</td>
</tr>
<tr>
<td>M-CAN-2(R_s)</td>
<td>83.23</td>
<td>72.81</td>
<td>89.37</td>
<td>85.42</td>
</tr>
<tr>
<td>M-CAN-2(R_o)</td>
<td>84.28</td>
<td>74.45</td>
<td>89.96</td>
<td>86.16</td>
</tr>
</tbody>
</table>

Table 3: Results of the ALSC task in multi-task settings in terms of accuracy (%) and Macro-F1 (%).

<table>
<thead>
<tr>
<th>Model</th>
<th>Rest14</th>
<th>Rest15</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>M-AT-LSTM</td>
<td>0.8626</td>
<td>0.8553</td>
</tr>
<tr>
<td>M-CAN-2(R_s)</td>
<td>0.8698</td>
<td>0.8595</td>
</tr>
<tr>
<td>M-CAN-2(R_o)</td>
<td><strong>0.8907</strong></td>
<td><strong>0.8627</strong></td>
</tr>
</tbody>
</table>

Table 4: Results of the ACD task. Rest14 has 5 aspect categories while Rest15 has 13 ones.

**4.5 Results**

Table 2 and 3 show our experimental results on the two public datasets for single-task and multi-task settings respectively. In both tables, “3-way” stands for 3-class classification (positive, neutral, and negative), and “Binary” for binary classification (positive and negative). The best scores are marked in bold.

**Single-task Settings** Table 2 shows our experimental results of ALSC in single-task settings. Firstly, we observe that by introducing attention regularizations (either \(R_s\) or \(R_o\)), most of our proposed methods outperform their counterparts. Particularly, AT-CAN-\(R_s\) and AT-CAN-\(R_o\) out-perform AT-LSTM in all results; ATAE-CAN-\(R_s\) and ATAE-CAN-\(R_o\) also outperform ATAE-LSTM in 15 of 16 results. For example, in the Rest15 dataset, ATAE-CAN-\(R_o\) outperforms ATAE-LSTM by 5.39% of accuracy and 6.46% of the F1 score in the 3-way classification. Secondly, regularization \(R_o\) achieves better performance improvement than \(R_s\) in all results. This is because \(R_o\) includes both orthogonal and sparse regularizations for non-overlapping multi-aspect sentences. Thirdly, our approaches, especially ATAE-CAN-\(R_o\), outperform the state-of-the-art baseline model GCMAE. Finally, the LSTM method outputs the worst results in all cases, because it can not distinguish different aspects.

**Multi-task Settings** Table 3 shows experimental results of ALSC in multi-task settings. We first observe that the overall results in multi-task settings outperform the ones in single-task settings, which demonstrates the effectiveness of multi-task learning by introducing the auxiliary ACD task to help the ALSC task. Second, in almost all cases, applying attention regularizations to both tasks gains more performance improvement than only to the ALSC task, which shows that our attention regularization approach can be extended to different tasks which involving aspect level attention weights, and works well in multi-task settings. For example, for the Binary classification in the Rest15 dataset, M-AT-LASTM outperforms AT-LSTM by 3.57% of accuracy and 4.96% of the F1 score, and M-CAN-2\(R_o\) further outperforms M-AT-LASTM by 3.28% of accuracy and 4.0% of the F1 score.

Table 4 shows the results of the ACD task in multi-task settings. Our proposed regularization terms can also improve the performance of ACD. Regularization \(R_o\) achieves the best performance in almost all metrics.

**4.6 Attention Visualizations**

Figure 3 depicts the attention weights from AT-LSTM, M-AT-LASTM and M-CAN-2\(R_o\) methods, which are used to predict the sentiment polarity in the ALSC task. The subfigure (a), (b) and (c) show the attention weights of the same sentence, for the aspect food and service respectively. We observe that the attention weights of each word associated with each aspect are quite different for different methods. For AT-LSTM method in sub-
As shown in subfigure (c), M-CAN+2Ro generates the best attention weights. The attention weights of the aspect food are almost orthogonal to the weights of service. The aspect food concentrates on the first part of the sentence while service on the second part. Meanwhile, the key opinion words “outstanding” and “tops” get highest attention weights in the corresponding aspects.

We also visualize the attention for the auxiliary task ACD. Figure 4 depicts the attention weights from the method M-CAN-2Ro. There are five predefined aspect categories (food, ambience, price, anecdotes/miscellaneous, service) in the dataset, two of which are mentioned in the sentence. In the ACD task, we need to calculate the attention weights for all the five aspect categories, and then generate aspect-specific sentence representations to determine whether the sentence contains each aspect. As shown in Figure 4, attention weights for aspects food and service are pretty good. The aspect food concentrates on words “food” and “outstanding”, and the aspect service focuses on the word “service”. It is interesting that for aspects which are not mentioned in the sentence, their attention weights often attend to meaningless stop words, such as “the”, “was”, etc. We do not distinguish these aspects and just treat them as a whole.

We plot the regularization loss curves in Figure 5, which shows that both Rs and Ro decrease during the training of AT-CAN-Ro.

**4.7 Case Studies**

**Overlapping Case** We only add sparse regularization to overlapping sentences in which multiple aspects share the same opinion snippet. As shown in Figure 6, the sentence contains two aspects food
and *service*, both described by the opinion snippet “highly disappointed”. Our method can locate the aspect terms and shared opinion words for both aspects, and then classify the sentiment correctly.

**Error Case** With the help of attention visualization, we can conduct error analysis of our model conveniently. As shown in Figure 6, for the first sentence in error case, the aspect *food* attends on the right word “disappointing”, but fails to include the negation word “never”. This may be caused by the inaccurate sentence representation or aspect embedding. We can not rebuild the connection between the aspect and the word by our regularizations. The second sentence is negative but classified as neutral. The attention weights distribute evenly since the sentence does not contain any explicit opinion words. Since there is no tendency which words to concentrate, our model can not adjust the attention weights and help on such cases.

### 5 Conclusions
We propose constrained attention networks for multi-aspect sentiment analysis, which handles multiple aspects of a sentence simultaneously. Specifically, we introduce orthogonal and sparse regularizations on attention weights. Furthermore, we introduce an auxiliary task ACD for promoting the ALSC task, and apply CAN on both tasks. Experimental results demonstrate that our approach outperforms the state-of-the-art methods.

### 6 Acknowledgement
This work is supported by National Science and Technology Major Project, China (Grant No. 2018YFB0204304).

### References
Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In ICLR.

Konstantinos Bousmalis, George Trigeorgis, Nathan Silberman, Dilip Krishnan, and Dumitru Erhan. 2016. Domain separation networks. In NIPS.


Feifan Fan, Yansong Feng, and Dongyan Zhao. 2018. Multi-grained attention network for aspect-level sentiment classification. In EMNLP.


Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In ACL.


4608


