CAN: Constrained Attention Networks for Multi-Aspect Sentiment Analysis

Mengting Hu[1]*, **Shiwan Zhao**[2], **Li Zhang**[2], **Keke Cai**[2], **Zhong Su**[2], **Renhong Cheng**[1], **Xiaowei Shen**[2]
[1] Nankai University, [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, compared to the sentence level classification. A sentence usually contains one or more aspects. To detect the sentiment towards a particular aspect in a sentence, previous studies have developed various methods for generating aspect-specific sentence representations. However, these studies handle each aspect of a sentence separately. In this paper, we argue that multiple aspects of a sentence are usually orthogonal based on the observation that different aspects concentrate on different parts of the sentence. To force the orthogonality among aspects, we propose constrained attention networks (CAN) for multi-aspect sentiment analysis, which handles multiple aspects of a sentence simultaneously. Experimental results on two public datasets demonstrate the effectiveness of our approach. We also extend our approach to multi-task settings, outperforming the state-of-the-arts significantly.

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 is a fine-grained sentiment analysis task, which aims at detecting the sentiment towards a particular aspect in a sentence. A multi-aspect sentence (*i.e.*, the sentence contains more than one aspect) can be categorized as **overlapping** or **non-overlapping**. A sentence is annotated as non-overlapping only if any two of its aspects have no overlap. One key observation is 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 in the left part of the sentence while *service* in the right part. 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 *"great"* is relevant to the aspect *food* and *"dreadful"* to *service*. The distribution of the opinion expression of each aspect is **sparse**.

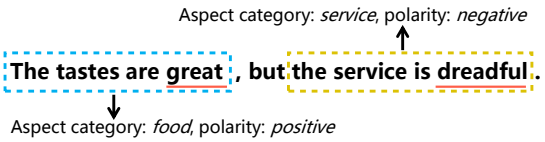


Figure 1: Example of a non-overlapping sentence.

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; Wang and Lu, 2018) have developed various attention-based methods for generating aspect-specific sentence representations. To name a few, (Wang et al., 2016) proposes attention-based LSTMs for aspect level sentiment classification. The attention mechanism can concentrate on different parts of a sentence when different aspects are taken as input. (Wang and Lu, 2018) proposes a segmentation attention based LSTM model which can effectively capture the structural dependencies between the target and the sentiment expressions with a linear-chain conditional random field (CRF) layer. However, all these works are single-aspect sentiment analysis, which deals with aspects in a sentence one at a time, ignoring the orthogonality among multiple aspects.

Therefore, we propose a model for multi-aspect sentiment analysis, which handles multiple aspects of a sentence simultaneously. Specifically, we introduce orthogonal regularization for attention weights among multiple non-overlapping as-

* This work was done when Mengting Hu was a research intern at IBM Research - China.

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). The implementation of adding regularization terms to attention weights of multiple aspects is similar to adding the penalization term in self-attention in (Lin et al., 2017). The details will be introduced in the model section.

In addition to aspect level sentiment classification (**ALSC**), aspect category detection (**ACD**) is another task of aspect based sentiment analysis. We introduce ACD as an auxiliary task to assist the ALSC task, benefiting from the shared context of the two tasks. Aspect category detection (Zhou et al., 2015; Schouten et al., 2018) is a task which 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 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 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. To the best of our knowledge, this is the first work for multi-aspect sentiment analysis.
- 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 classification is a fine-grained sentiment analysis task. Earlier methods are usually based on explicit features (Liu et al., 2010; Vo and Zhang, 2015). (Liu et al., 2010) uses different linguistic features for sentiment classification. (Vo and Zhang, 2015) studies aspect-based Twitter sentiment classification by applying automatic features, which are obtained from unsupervised learning methods. With the rapid development of deep learning technologies, many end-to-end neural networks are implemented to solve this fine-grained task. (Wang et al., 2016) proposes an attention-based LSTM network for aspect-level sentiment classification. (Tay et al., 2018) introduces a word aspect fusion attention layer to learn attentive representations. (Ma et al., 2017) proposes the interactive attention networks to generate the representations for targets and contexts separately. (Tay et al., 2017) proposes dyadic memory networks for aspect based sentiment analysis. (Cheng et al., 2017; Ruder et al., 2016) both propose hierarchical neural network models for aspect level sentiment classification. (Ma et al., 2018) proposes a two-step attention model for targeted aspect-based sentiment analysis. (Wang and Lu, 2018) proposes a segmentation attention based LSTM model for aspect level sentiment classification. However, all these works can be categorized as single-aspect sentiment analysis, which deals with aspects in a sentence separately, ignoring the orthogonality among multiple aspects.

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) learns both main task and auxiliary task jointly with shared representations, achieving improved performance in question answering. (Toshniwal et al., 2017) uses 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) uses 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,

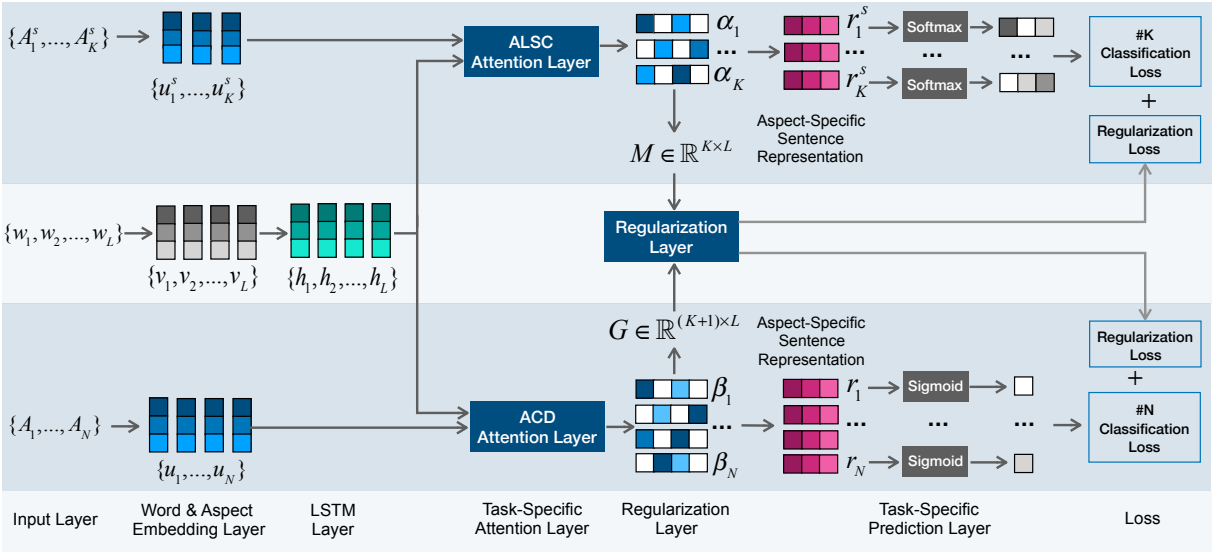


Figure 2: Network Architecture. The aspect categories are embedded as vectors. The model encodes the sentence using LSTM. Based on its hidden states, aspect-specific sentence representations for ALSC and ACD tasks are learned via constrained attention. Then aspect level sentiment prediction and aspect category detection are made.

$A = \{A_1, ..., A_N\}$. Given a sentence $S = \{w_1, w_2, ..., w_L\}$, which contains K aspects $A^s = \{A^s_1, ..., A^s_K\}$, $A^s \subseteq A$, 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.

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 Input and Embedding Layers

Traditionally, aspect based 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, each of which is associated with a single aspect. 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, $\langle S, A^s_1, p_1\rangle$ and $\langle S, A^s_2, p_2\rangle$, will be constructed.

In this paper, our model is for multi-aspect sentiment analysis, handling multiple aspects of a sentence together. For the sentence S with two aspects A^s_1 and A^s_2, the input to our model is $\langle S, [A^s_1, A^s_2], [p_1, p_2]\rangle$, as a single instance.

With embedding matrices, the input sentence $\{w_1, w_2, ..., w_L\}$ is converted to a sequence of vectors $\{v_1, v_2, ..., v_L\}$, and the K aspects of the sentence are transformed to vectors $\{u^s_1, ..., u^s_K\}$, which is a subset of $\{u_1, ..., u_N\}$, the vectors of all aspect categories. The embedding dimension is d.

3.2 LSTM Layer

The word embeddings of the sentence are then fed into an LSTM network (Hochreiter and Schmidhuber, 1997), which outputs hidden states $H = \{h_1, h_2, ..., h_L\}$. At each time step l, the hidden state h_l of the LSTM is computed by:

$$h_l = LSTM(h_{l-1}, v_l) \quad (1)$$

The size of the hidden state is also set to be d.

3.3 Task-Specific Attention Layer

Our multi-task learning framework supports both ALSC and ACD tasks. The two 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 (Bahdanau et al., 2015) to compute the attention. Particularly, given the sentence S with K aspects, $A^s = \{A^s_1, ..., A^s_K\}$, for each aspect A^s_k, its attention weights are calculated by:

$$\alpha_k = softmax(z^{a\mathrm{T}} tanh(W^a_1 H + W^a_2 (u^s_k \otimes e_L))) \quad (2)$$

where u^s_k is the embedding of the aspect A^s_k, $e_L \in \mathbb{R}^L$ is a vector of 1s, $u^s_k \otimes e_L$ is the op-

eration repeatedly concatenating u_k^s for L times. $W_1^a \in \mathbb{R}^{d\times d}$, $W_2^a \in \mathbb{R}^{d\times d}$ 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:

$$\beta_n = softmax(z^{b^{\mathrm{T}}} tanh(W_1^b H + W_2^b(u_n \otimes e_L))) \quad (3)$$

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).

3.4 Regularization Layer

Multi-aspect sentiment analysis 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 (α_k or β_n) to concentrate on less words. For simplicity, we use α_k as an example, $\alpha_k = \{\alpha_{k1}, \alpha_{k2}, ..., \alpha_{kL}\}$. To make α_k sparse, the sparse regularization term is defined as:

$$R_s = \mid \sum_{l=1}^{L} \alpha_{kl}^2 - 1 \mid \quad (4)$$

where $\sum_{l=1}^{L} \alpha_{kl} = 1$ and $\alpha_{kl} > 0$. Since α_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 4 will force the sparsity of α_k. It has the similar effect as minimizing the entropy of α_k, which leads to placing more probabilities on less words.

Orthogonal Regularization This regularization term forces orthogonality between attention weight vectors of different 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 $\{A_1^s, ..., A_K^s\}$ and their attention weight vectors are $\{\alpha_1, ..., \alpha_K\}$. We pack them together as a two-dimensional attention matrix $M \in \mathbb{R}^{K\times L}$ to calculate the orthogonal regularization term.

$$R_o = \parallel M^{\mathrm{T}} M - I \parallel_2 \quad (5)$$

where I is an identity matrix. In the resulted matrix of $M^{\mathrm{T}}M$, 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 $M^{\mathrm{T}}M$ are subtracted by 1, which are the same as R_s defined in Equation 4. 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 $\{\beta_1, ..., \beta_N\}$ as a matrix. The sentence S contains K aspects. For simplicity, let $\{\beta_1, ..., \beta_K\}$ be the attention vectors of the K aspects mentioned, while $\{\beta_{K+1}, ..., \beta_N\}$ be the attention vectors of the $N - K$ aspects not mentioned. We compute the average of the $N - K$ attention vectors, denoted by β_{avg}. We then construct the attention matrix $G = \{\beta_1, ..., \beta_K, \beta_{avg}\}$, $G \in \mathbb{R}^{(K+1)\times L}$. The reason why we calculate β_{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.

3.5 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

final aspect-specific sentence representation.

$$r_k^s = tanh(W_1^r \bar{h}_k + W_2^r h_L) \quad (6)$$

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

$$\hat{y_k} = softmax(W_p^a r_k^s + b_p^a) \quad (7)$$

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.

$$r_n = \bar{h}_n = \sum_{l=1}^{L} \beta_{nl} h_l \quad (8)$$

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.

$$\hat{y_n} = sigmoid(W_p^b r_n + b_p^b) \quad (9)$$

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

3.6 Loss

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

$$L_a = -\sum_{k=1}^{K}\sum_{c} y_{kc} \log \hat{y_{kc}} \quad (10)$$

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:

$$L_b = -\sum_{n=1}^{N}[y_n \log \hat{y_n} + (1-y_n)\log(1-\hat{y_n})] \quad (11)$$

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

$$L = L_a + \frac{1}{N} L_b + \lambda R \quad (12)$$

where R is the regularization term mentioned previously, which can be R_s or R_o. λ 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.

Dataset	#Single	#Multi			#Total
		OL	*NOL*	*Total*	
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

Table 1: The numbers of single- and multi-aspect sentences. *OL* and *NOL* denote the overlapping and non-overlapping multi-aspect sentences, respectively.

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 (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. 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.

4.2 Comparison Methods

- **LSTM**: We implement the vanilla LSTM networks to model the sentence and use the average of all 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 2. We do not concatenate the aspect embedding to the hidden state as in (Wang et al., 2016) and gain small performance improvement. We use this

modified version of AT-LSTM 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.

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-R_s**: Add sparse regularization R_s to AT-LSTM to constrain the attention weights of each single aspect.
- **AT-CAN-R_o**: Add orthogonal regularization R_o to AT-CAN-R_s to constrain the attention weights of multiple non-overlapping aspects.
- **ATAE-CAN-R_s**: Add R_s to ATAE-LSTM.
- **ATAE-CAN-R_o**: Add R_o to ATAE-CAN-R_s.

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-R_s**: Add R_s to the ALSC task in M-AT-LSTM.
- **M-CAN-R_o**: Add R_o to the ALSC task in M-CAN-R_s.
- **M-CAN-2R_s**: Add R_s to both tasks in M-AT-LSTM.
- **M-CAN-2R_o**: Add R_o to both tasks in M-CAN-2R_s.

4.4 Implementation Details

We implement all models in Pytorch. We set $\lambda = 1$ 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) and the aspect embedding matrix is initialized by sampling from a uniform distribution $U(-\varepsilon, \varepsilon)$, $\varepsilon = 0.01$. These two embedding matrices are updated during training. The hidden cell dimension of LSTM and the aspect embedding dimension are both 300. Parameters are initialized by sampling from the uniform distribution $U(-\varepsilon, \varepsilon)$. 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. The reported results are the evaluation running against the test set using the saved model. To reduce the randomness of results, we train each model three times and report their averaged scores.

It is worth pointing out that in the testing stage, we infer all sentences in the same way, ignoring the difference between single-aspect and multi-aspect, between overalpping and non-overlapping.

Model	Rest14		Rest15	
	3-way	Binary	3-way	Binary
LSTM	80.61	86.66	73.14	73.27
AT-LSTM	81.66	87.13	75.15	76.40
ATAE-LSTM	82.08	87.72	74.32	76.79
AT-CAN-R_s	81.97	88.08	75.74	80.05
AT-CAN-R_o	82.60	88.67	75.03	81.10
ATAE-CAN-R_s	82.29	87.37	76.09	80.83
ATAE-CAN-R_o	**82.91**	**89.02**	**77.28**	**82.66**

Table 2: Results of the ALSC task in terms of accuracy (%). All methods are run in single-task settings.

Model	Rest14		Rest15	
	3-way	Binary	3-way	Binary
M-AT-LSTM	82.39	88.31	75.38	80.44
M-CAN-R_s	83.02	89.14	76.57	80.57
M-CAN-R_o	83.23	89.37	76.45	81.36
M-CAN-2R_s	83.54	89.02	76.92	**81.49**
M-CAN-2R_o	**84.28**	**90.08**	**77.87**	81.23

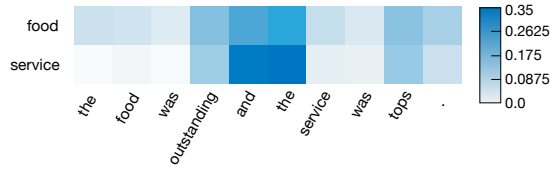
Table 3: Results of the ALSC task in terms of accuracy (%). All methods are run in multi-task settings.

Model	Precision	Recall	F1
M-AT-LSTM	0.8626	0.8553	0.8589
M-CAN-2R_s	0.8698	0.8595	0.8645
M-CAN-2R_o	**0.8907**	**0.8627**	**0.8765**

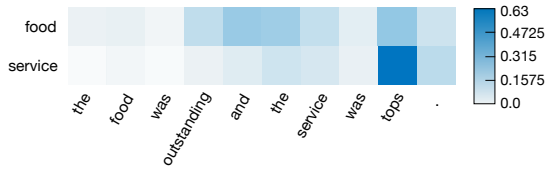
Table 4: Results of the ACD task on the Rest14 dataset.

4.5 Results

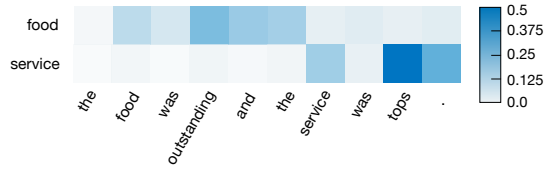
Table 2 and 3 show our experimental results on the two public datasets for single-task and multi-task settings respectively. In both tables, "3-way" stands for 3-class classification (positive, neutral,



(a) AT-LSTM



(b) M-AT-LSTM



(c) M-CAN-2R_o

Figure 3: Visualization of attention weights of different aspects in the ALSC task. Three different models are compared.

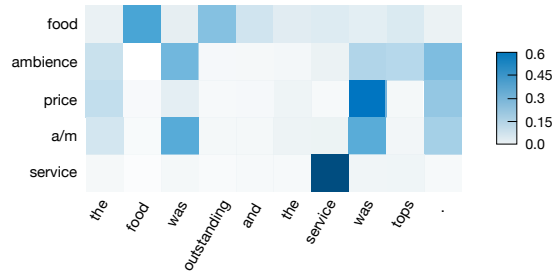


Figure 4: Visualization of attention weights of different aspects in the ACD task from M-CAN-2R_o. The a/m is short for anecdotes/miscellaneous.

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 aspect level sentiment classification 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. Specifically, AT-CAN-R_s and AT-CAN-R_o outperform AT-LSTM in 7 of 8 results; ATAE-CAN-R_s and ATAE-CAN-R_o also outperform ATAE-LSTM in 7 of 8 results. For example, in the Rest15 dataset, ATAE-CAN-R_o outperforms ATAE-LSTM by up to 7.64% in the Binary 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

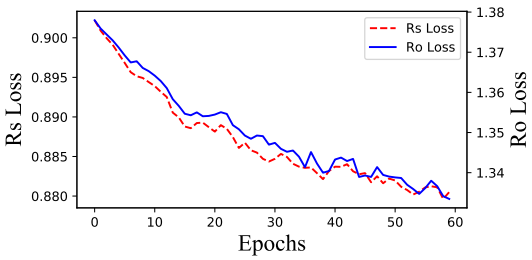


Figure 5: The regularization loss curves of R_s and R_o during the training of AT-CAN-R_o.

for non-overlapping multi-aspect sentences. Finally, the LSTM method outputs the worst results in all cases, because it can not distinguish different aspects. We do not add regularization terms to the LSTM method since no attention weights are computed in this method.

Multi-task Settings Table 3 shows experimental results of aspect level sentiment classification 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 Rest14 dataset, M-AT-LASTM outperforms AT-LSTM by 1.35%, and M-CAN-2R_o further outperforms M-AT-LSTM by 2.00% (outperforms AT-LSTM by 3.39%).

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 all metrics.

4.6 Attention Visualizations

Figure 3 depicts the attention weights from AT-LSTM, M-AT-LSTM and M-CAN-2R_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 subfigure (a), the attention weights of aspect *food*

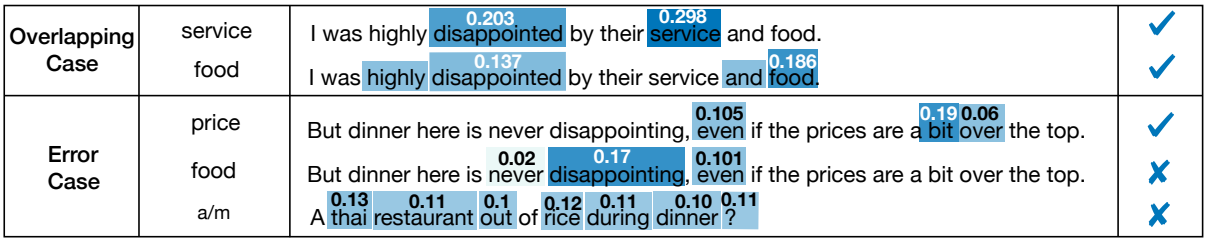


Figure 6: Examples of overlapping case and error case. The a/m is short for anecdotes/miscellaneous.

and *service* are both high in words *"outstanding"*, *"and"*, and *"the"*, but actually, the word *"outstanding"* is used to describe the aspect *food* rather than *service*. The same situation occurs with the word *"tops"*, which should associate with *service* rather than *food*. The attention mechanism alone is not good enough to locate aspect-specific opinion words and generate aspect-specific sentence representations in the ALSC task.

As shown in subfigure (b), the issue is mitigated in M-AT-LSTM. Multi-task learning can learn better hidden states of the sentence, and better aspect embeddings. However, it is still not good enough. For instance, the attention weights of the word *"tops"* are both high for the two aspects, and the weights are overlapped in the middle part of the sentence.

As shown in subfigure (c), M-CAN+2R_o 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-2R_o. 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 also plot the regularization loss curves in Figure 5, which shows that both R_s and R_o decrease during the training of AT-CAN-R_o.

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 disappointing"*. 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 cases, the aspect *price* attends on right words and be classified correctly. While 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 Conclusion

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-arts significantly.

References

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

Rich Caruana. 1997. Multitask learning. *Machine Learning*, 28(1):41–75.

Jiajun Cheng, Shenglin Zhao, Jiani Zhang, Irwin King, Xin Zhang, and Hui Wang. 2017. Aspect-level sentiment classification with heat (hierarchical attention) network. In *(CIKM 2017)*, pages 97–106.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Sepp Hochreiter and Jrgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Binxuan Huang, Yanglan Ou, and Kathleen M Carley. 2018. Aspect level sentiment classification with attention-over-attention neural networks. *arXiv preprint arXiv:1804.06536*.

Yanzhou Huang and Tao Zhong. 2018. Multitask learning for neural generative question answering. *Machine Vision & Applications*, pages 1–9.

Zhouhan Lin, Minwei Feng, Cicero Nogueira dos Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. 2017. A structured self-attentive sentence embedding. In *(ICLR 2017)*.

Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Feifan Liu, Dong Wang, Bin Li, and Yang Liu. 2010. Improving blog polarity classification via topic analysis and adaptive methods. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 309–312.

Dehong Ma, Sujian Li, Xiaodong Zhang, Houfeng Wang, Dehong Ma, Sujian Li, Xiaodong Zhang, and Houfeng Wang. 2017. Interactive attention networks for aspect-level sentiment classification. In *(IJCAI 2017)*, pages 4068–4074.

Yukun Ma, Haiyun Peng, and Erik Cambria. 2018. Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm. In *(AAAI 2018)*.

Tetsuya Nasukawa and Jeonghee Yi. 2003. Sentiment analysis:capturing favorability using natural language processing. In *International Conference on Knowledge Capture (K-CAP)*, pages 70–77.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *(EMNLP 2014)*, pages 1532–1543.

Maria Pontiki, Dimitris Galanis, John Pavlopoulos, Harris Papageorgiou, Ion Androutsopoulos, and Suresh Manandhar. 2014. Semeval-2014 task 4: Aspect based sentiment analysis. *(SemEval 2014)*, pages 27–35.

Sebastian Ruder, Parsa Ghaffari, and John G. Breslin. 2016. A hierarchical model of reviews for aspect-based sentiment analysis. In *(EMNLP 2016)*, pages 999–1005.

Kim Schouten, Onne Van Der Weijde, Flavius Frasincar, and Rommert Dekker. 2018. Supervised and unsupervised aspect category detection for sentiment analysis with co-occurrence data. *IEEE Transactions on Cybernetics*, 48(4):1263–1275.

Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Learning to attend via word-aspect associative fusion for aspect-based sentiment analysis. In *(AAAI 2018)*.

Yi Tay, Luu Anh Tuan, and Siu Cheung Hui. 2017. Dyadic memory networks for aspect-based sentiment analysis. In *(CIKM 2017)*, pages 107–116.

Shubham Toshniwal, Hao Tang, Liang Lu, and Karen Livescu. 2017. Multitask learning with low-level auxiliary tasks for encoder-decoder based speech recognition. *INTERSPEECH*, pages 3532–3536.

Duy Tin Vo and Yue Zhang. 2015. Target-dependent twitter sentiment classification with rich automatic features. In *(IJCAI 2015)*, pages 1347–1353.

Bailin Wang and Wei Lu. 2018. Learning latent opinions for aspect-level sentiment classification. In *(AAAI 2018)*.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016. Attention-based lstm for aspect-level sentiment classification. In *(EMNLP 2016)*, pages 606–615.

Jianfei Yu and Jing Jiang. 2016. Learning sentence embeddings with auxiliary tasks for cross-domain sentiment classification. In *(EMNLP 2016)*, pages 236–246.

Xinjie Zhou, Xiaojun Wan, and Jianguo Xiao. 2015. Representation learning for aspect category detection in online reviews. In *(AAAI 2015)*, pages 417–423.