DOER: Dual Cross-Shared RNN for Aspect Term-Polarity Co-Extraction

Huaishao Luo[1], **Tianrui Li**[1*], **Bing Liu**[2], **Junbo Zhang**[3,4,5]
[1]School of Information Science and Technology, Southwest Jiaotong University, China
huaishaoluo@gmail.com, trli@swjtu.edu.cn
[2]Department of Computer Science, University of Illinois at Chicago, USA
liub@uic.edu
[3]JD Intelligent Cities Business Unit & [4]JD Intelligent Cities Research, China
[5]Institute of Artificial Intelligence, Southwest Jiaotong University, China
msjunbozhang@outlook.com

Abstract

This paper focuses on two related subtasks of aspect-based sentiment analysis, namely aspect term extraction and aspect sentiment classification, which we call *aspect term-polarity co-extraction*. The former task is to extract aspects of a product or service from an opinion document, and the latter is to identify the polarity expressed in the document about these extracted aspects. Most existing algorithms address them as two separate tasks and solve them one by one, or only perform one task, which can be complicated for real applications. In this paper, we treat these two tasks as two sequence labeling problems and propose a novel Dual crOss-sharEd RNN framework (DOER) to generate all aspect term-polarity pairs of the input sentence simultaneously. Specifically, DOER involves a dual recurrent neural network to extract the respective representation of each task, and a cross-shared unit to consider the relationship between them. Experimental results demonstrate that the proposed framework outperforms state-of-the-art baselines on three benchmark datasets.

1 Introduction

Aspect terms extraction (ATE) and aspect sentiment classification (ASC) are two fundamental, fine-grained subtasks of aspect-based sentiment analysis. Aspect term extraction is the task of extracting the attributes (or aspects) of an entity upon which opinions have been expressed, and aspect sentiment classification is the task of identifying the polarities expressed on these extracted aspects in the opinion text (Hu and Liu, 2004). Consider the example in Figure 1, which contains comments that people expressed about the aspect terms "operating system", "preloaded software", "keyboard", "bag", "price", and "service" labeled with their polarities, respectively. The polarities contain four classes, e.g., positive (PO), conflict (CF), neutral (NT)[1], and negative (NG).

I love the **[operating system]**$_{\textbf{positive}}$ and the **[preloaded software]**$_{\textbf{positive}}$.

No backlit **[keyboard]**$_{\textbf{conflict}}$, but not an issue for me.

You may need to special order a **[bag]**$_{\textbf{neutral}}$.

The **[price]**$_{\textbf{positive}}$ is reasonable although the **[service]**$_{\textbf{negative}}$ is poor.

Figure 1: Aspect terms extraction and aspect sentiment classification.

To facilitate practical applications, our goal is to solve ATE and ASC simultaneously. For easy description and discussion, these two subtasks are referred to as aspect term-polarity co-extraction. Both ATE and ASC have attracted a great of attention among researchers, but they are rarely solved together at the same time due to some challenges:

1) *ATE and ASC are quite different tasks.* ATE is an extraction or sequence labeling task (Jakob and Gurevych, 2010; Wang et al., 2016a), while ASC is a classification task (Jiang et al., 2011; Wagner et al., 2014; Tang et al., 2016a,b; Tay et al., 2018). Thus, they are naturally treated as two separate tasks, and solved one by one in a pipeline manner. However, this two-stage framework is complicated and difficult to use in applications because it needs to train two models separately. There is also the latent error propagation when an aspect term is used to classify its corresponding polarity. Thus, due to the different natures of the two tasks, most current works focus either on extracting aspect terms (Yin et al., 2016; Luo et al., 2018; Xu et al., 2018) or on classifying aspect sentiment (Ma et al., 2017; Wang and Lu, 2018). A possible idea to bridge the difference between the two tasks is to change ASC to a sequence labeling task. Then, ATE and ASC

[*] Tianrui Li is the corresponding author.

[1] Neutral means no sentiment is expressed, and we also regard it as a polarity as in many prior works.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 591–601
Florence, Italy, July 28 - August 2, 2019. ©2019 Association for Computational Linguistics

have the same formulation.

2) *The number of aspect term-polarity pairs in a sentence is arbitrary.* Considering the examples depicted in Figure 1, we can observe that some sentences contain two term-polarity pairs and some sentences contain one pair. Moreover, each aspect term can consist of any number of words, which makes the co-extraction task difficult to solve.

Some existing research has treated ATE and ASC as two sequence labeling tasks and dealt with them together. Mitchell et al. (2013) and Zhang et al. (2015) compared pipelined, joint, and collapsed approaches to extracting named entities and their sentiments. They found that the joint and collapsed approaches are superior to the pipelined approach. Li and Lu (2017) proposed a collapsed CRF model. The difference with the standard CRF is that they expanded the node type at each word to capture sentiment scopes. Another interesting work comes from Li et al. (2019), where the authors proposed a unified model with the collapsed approach to do aspect term-polarity co-extraction. We can intuitively explain the pipelined, joint, and collapsed approaches through Figure 2. The pipelined approach first labels the given sentence using aspect term tags, e.g., "B" and "I" (the Beginning and Inside of an aspect term) and then feeds the aspect terms into a classifier to obtain their corresponding polarities. The collapsed approach uses collapsed labels as the tags set, e.g., "B-PO" and "I-PO". Each tag indicates the aspect term boundary and its polarity. The joint approach jointly labels each sentence with two different tag sets: aspect term tags and polarity tags.

We believe that the joint approach is more feasible than the collapsed approach when integrating with neural networks because the combined tags of the latter may easily make the learned representation confused. As an example in Figure 2, the "operating system" is an aspect term. Its polarity "positive" actually comes from the word "love". They should be learned separately because the meanings of these two groups of words are different. That means that using "B-PO I-PO" to extract the meaning of "operating system" and "love" simultaneously is difficult in training (this will be clearer later). In contrast, the joint approach has separate representations for ATE and ASC and separate labels. Thus, an extra sentiment lexicon can improve the representation of ASC individually, and the interaction of ATE and ASC can further enhance the performance of each other.

Input	I	love	the	**operating**	**system**	and	the	**preloaded**	**software**	.
Joint	O	O	O	B	I	O	O	B	I	O
	O	O	O	PO	PO	O	O	PO	PO	O
Collapsed	O	O	O	B-PO	I-PO	O	O	B-PO	I-PO	O

Figure 2: A labeling example of aspect terms and their polarities.

In this paper, we propose a novel Dual crOss-sharEd RNN framework (DOER) to generate all aspect term-polarity pairs of a given sentence. DOER mainly contains a dual recurrent neural network (RNN) and a cross-shared unit (CSU). The CSU is designed to take advantage of the interactions between ATE and ASC. Apart from them, two auxiliary tasks, *aspect length enhancement* and *sentiment enhancement*, are integrated to improve the representation of ATE and ASC. An extra RNN cell called the *Residual Gated Unit* (ReGU) is also proposed to improve the performance of aspect term-polarity co-extraction. The ReGU utilizes a gate to transfer the input to the output like skip connection (He et al., 2016), and thus, is capable of training deeper and obtaining more useful features. In a word, DOER generates aspect terms and their polarities simultaneously by an end-to-end method instead of building two separate models, which saves time and gives a unified solution to practical applications.

Our contributions are summarized as follows:

- A novel framework DOER is proposed to address the aspect term-polarity co-extraction problem in an end-to-end fashion. A cross-shared unit (CSU) is designed to leverage the interaction of the two tasks.

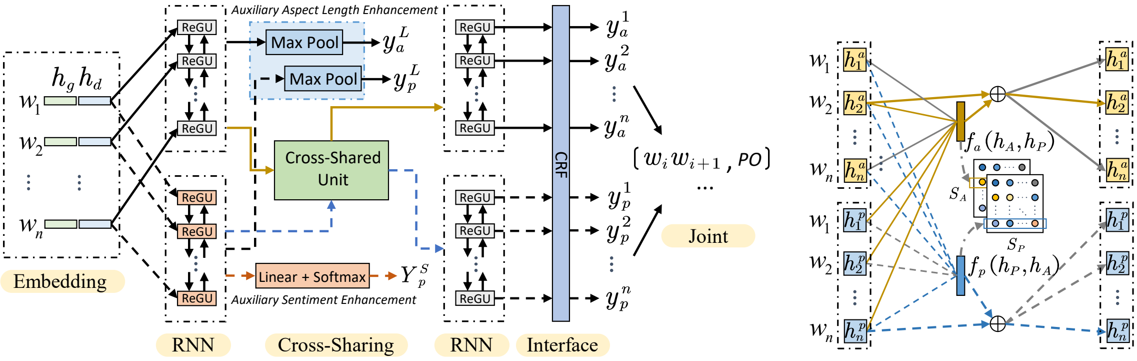
- Two auxiliary tasks are designed to enhance the labeling of ATE and ASC, and an extra RNN cell ReGU is proposed to improve the capability of feature extraction.

2 Methodology

The proposed framework is shown in Figure 3a. We will first formulate the aspect term-polarity co-extraction problem and then describe this framework in detail in this section.

2.1 Problem Statement

This paper deals with aspect term-polarity co-extraction, in which the aspect terms are explicitly



(a) Dual cross-shared RNN framework (DOER)

(b) Cross-shared unit (CSU)

Figure 3: An illustration of the proposed DOER framework.

mentioned in the text. We solve it as two sequence labeling tasks. Formally, given a review sentence S with n words from a particular domain, denoted by $S = \{w_i | i = 1, \ldots, n\}$. For each word w_i, the objective of ATE is to assign it a tag $t_i^a \in T^a$, and likewise, the objective of ASC is to assign a tag $t_i^p \in T^p$, where $T^a = \{\text{B, I, O}\}$ and $T^p = \{\text{PO, NT, NG, CF, O}\}$. The tags B, I and O in T^a stand for the beginning of an aspect term, the inside of an aspect term, and other words, respectively. The tags PO, NT, NG, and CF indicate polarity categories: positive, neutral, negative, and conflict, respectively. The tag O in T^p means other words like that in T^a. Figure 2 shows a labeling example of the first sentence in Figure 1.

2.2 Model Overview

We discuss the proposed framework DOER in detail below.

Word Embedding. Instead of adopting standard techniques to generate the embedding of each word w_i by concatenating word embedding and char embedding, we use the double embeddings proposed in (Xu et al., 2018) as the initial word embeddings. The double embeddings contain two types: general-purpose embeddings and domain-specific embeddings, which are distinguished by whether the embeddings are trained by an in-domain corpus or not. Formally, each word w_i will be initialized with a feature vector $h_{w_i} \in \mathbb{R}^{d_G + d_D}$, where d_G and d_D are the first dimension size of the general-purpose embeddings $\text{G} \in \mathbb{R}^{d_G \times |V|}$ and the domain-specific embeddings $\text{D} \in \mathbb{R}^{d_D \times |V|}$, respectively. $|V|$ is the size of the vocabulary. Hence, h_{w_i} is generated by $h_{w_i} = G(w_i) \oplus D(w_i)$, where $\oplus$ means the concatenation operation. h_g and h_d in Figure 3a denote $G(w_i)$ and $D(w_i)$, respectively. All the out-of-vocabulary words are randomly initialized, and all sentences are padded (or tailored when testing) and initialized with zeros to the max length of the training sentences.

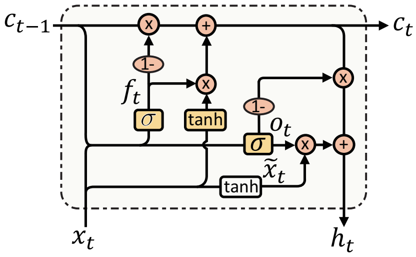


Figure 4: Residual gated unit (ReGU).

Stacked Dual RNNs. The main architecture of DOER is a stacked dual RNNs, one stacked RNN for ATE, and one stacked RNN for ASC. Each layer of RNNs is a bidirectional ReGU (BiReGU). As shown in Figure 4, ReGU has two gates to control the flow of input and hidden state. Given input x_t at time t and the previous memory cell c_{t-1}, the new memory cell c_t is calculated via the following equation:

$$c_t = (1 - f_t) \odot c_{t-1} + f_t \odot \tanh(W_i x_t), \qquad (1)$$

and the new hidden state h_t is then computed as

$$h_t = (1 - o_t) \odot c_t + o_t \odot \tilde{x}_t, \qquad (2)$$

where $f_t = \sigma(W_f x_t + U_f c_{t-1})$ is a forget gate, $o_t = \sigma(W_o x_t + U_o c_{t-1})$ is a residual gate, and $\tilde{x}_t$ is x_t or $\tanh(W_x x_t)$ according to whether the size of x_t is equal to c_t or not. f_t controls the information flow from the previous timestamp to the next timestamp. o_t controls the information flow from the previous layer to the next layer. σ denotes the logistic function, *tanh* means the hyperbolic tangent function, and $\odot$ is element-wise multiplication. W_* of size $d \times d_I$ and U_* of size $d \times d$ are weight matrices, where $* \in \{i, f, o, x\}$. The bias vectors are omitted for simplicity. The size of d_I changes with the dimension of the input. Its value is $d_G + d_D$ when it is the first layer of the stacked BiReGU.

BiReGU owns two directional representations of the input like Bidirectional LSTM (Graves and Schmidhuber, 2005). We concatenate the hidden states generated by ReGU in both directions belonging to the same input as the output vector, which is expressed as $h_t = \overrightarrow{h}_t \oplus \overleftarrow{h}_t$, where $\oplus$ again means concatenation. $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$ have the same formulation as Eq. (2) but different propagation directions. Thus, the size of h_t is $2d$, and the size of d_I will also become $2d$ when stacking a new BiReGU layer. We refer the outputs of dual BiReGU as h_A and h_P separately to differentiate ATE and ASC.

Cross-Shared Unit. When generating the representation after BiReGU, the information of ATE and ASC is separated from each other. However, the fact is that the labels of ATE and the labels of ASC have strong relations. For instance, if the label of ATE is O, the label for ASC should be O as well, and if the label of ASC is PO, the label for ATE should be B or I. Besides, both the labels of ATE and the labels of ASC have the information to imply the boundary of each aspect term.

The cross-shared unit (CSU) is used to consider the interaction of ATE and ASC. We first compute the composition vector $\alpha_{ij}^{M} \in \mathbb{R}^K$ through the following tensor operator:

$$\alpha_{ij}^{M} = f_m\left(h_i^m, h_j^{\overline{m}}\right) = \tanh\left(\left(h_i^m\right)^{\top} G^m h_j^{\overline{m}}\right), \tag{3}$$

where $M \in \{A, P\}$, $m \in \{a, p\}$, $h_i^m \in h_M$, and $G^m \in \mathbb{R}^{K \times 2d \times 2d}$ are 3-dimensional tensors. K is a hyperparameter. A, a and P, p are indexes of ATE and ASC, respectively, $\overline{m} = p, M = A$ if $m = a$, and $\overline{m} = a, M = P$ if $m = p$. Such tensor operators can be seen as multiple bilinear terms, which have the capability of modeling more complicated compositions between two vectors (Socher et al., 2013; Wang et al., 2017).

After obtaining the composition vectors, the attention score S_{ij}^{M} is calculated as:

$$S_{ij}^{M} = v_m^{\top} \alpha_{ij}^{M}, \tag{4}$$

where $v_m \in \mathbb{R}^K$ is a weight vector used to weight each value of the composition vector, $M \in \{A, P\}$, and $m \in \{a, p\}$. Thus, S_{ij}^{M} is a scalar. All these scalars S_{ij}^{A} and S_{ij}^{P} are gathered in two matrices S_A and S_P, respectively. A higher score S_{ij}^{A} indicates a higher correlation between aspect term i and the polarity representation captured from j-th word. Likewise, a higher score S_{ij}^{P} indicates a higher correlation between aspect polarity i and the representation of aspect term captured from j-th word. We use their related representations to enhance the original ATE and ASC features through:

$$h_M = h_M + \text{softmax}_r\left(S^M\right) h_{\overline{M}}, \tag{5}$$

where softmax_r is a row-based softmax function, $M \in \{A, P\}$, $\overline{M} = P$ if $M = A$, and $\overline{M} = A$ if $M = P$. Such an operation can make ATE and ASC get enhanced information from each other. The process is shown in Figure 3b.

Interface. To generate the final ATE tags and ASC tags, either a dense layer plus a softmax function or a Conditional Random Fields (CRF) can be used. According to the comparison in (Reimers and Gurevych, 2017), using a CRF instead of a softmax classifier as the last layer can obtain a performance increase for tasks with a high dependency between tags. Thus, we use the linear-chain CRF as our inference layer. Its log-likelihood is computed as follows:

$$L\left(W_c, b_c\right) = \sum_i \log\ p\left(y|h; W_c, b_c\right). \tag{6}$$

where $p\left(y|h; W_c, b_c\right)$ is the probability function of CRF, and W_c and b_c are the weight and bias, respectively. The Viterbi algorithm is used to generate the final labels of ATE and ASC.

Joint Output. After generating the labels for ATE and ASC in the inference layer, the last step is to obtain the aspect term-polarity pairs. It is convenient to get the aspect terms of the given sentence according to the meaning of the elements in T^a. To generate the polarity of each aspect term, we use the aspect term as the boundary of polarity labels, and then count the number of each polarity category within the boundary and adopt the label that has the maximum number or the first label (if all the numbers of each polarity category are equal) as the final polarity. For example, the final polarity of "PO NT" is "PO", the final polarity of "PO PO" is also "PO", and the final polarity of "PO NT NT" is "NT". This method is simple and effective in our experiments.

Auxiliary Aspect Term Length Enhancement. Although CRF is capable of considering the correlation of two adjacent labels, there are generated discontinuous labels, especially for a long target aspect term. To alleviate the influence resulted from the length of the aspect term, we designed an auxiliary task to predict the average length of aspect terms in each sentence when training the model. The computational process of the prediction in ATE is as follows:

$$z_{u_A} = \sigma\left(W_{u_A}^{\top}\tilde{h}_A\right), \tag{7}$$

where $\tilde{h}_A \in \mathbb{R}^{2d}$ is the result of max-pooling of $h_A^{l_1}$, which is generated by the first RNN layer, $W_{u_A} \in \mathbb{R}^{2d}$ is a weight parameter. We calculate the prediction loss through the mean squared error (MSE):

$$\mathcal{L}_{u_A} = \|z_{u_A} - \hat{z}_u\|^2, \tag{8}$$

where $\hat{z}_u$ is the average length of aspect terms in a sentence after global normalization on the training dataset.

ASC has a similar prediction process to ATE after the first layer of the stacked RNNs, but it has different weight W_{u_P} and hidden feature $\tilde{h}_P$ than W_{u_A} and $\tilde{h}_A$. The prediction loss is denoted by $\mathcal{L}_{u_P}$.

Auxiliary Sentiment Lexicon Enhancement. As previously discussed, the polarity of an aspect term is usually inferred from its related opinion words. Thus, we also use a sentiment lexicon to guide ASC. Specifically, we train an auxiliary word-level classifier on the branch of ASC for discriminating positive words and negative words based on the sentiment labels $\hat{Y}_p^S$. This means that we use a sentiment lexicon to map each word of a sentence to a sentiment label in training. For each feature of ASC h_i^{p,l_1} generated by the first RNN layer, we use a linear layer and the softmax function to get its sentiment label:

$$z_i^s = \text{softmax}\left(W_s^{\top}h_i^{p,l_1}\right), \tag{9}$$

where $W_s \in \mathbb{R}^{2d\times c}$ is a weight parameter, $c = 3$ means the sentiment label is one of the three elements in the set {positive, negative, none}. We use the cross-entropy error to calculate the loss of each sentence:

$$\mathcal{L}_s = -\frac{1}{n}\sum_{i=1}^{n}\left(\mathbb{I}\left(\hat{y}_i^S\right)\left(\log\left(z_i^s\right)\right)^{\top}\right), \tag{10}$$

where $\mathbb{I}(\hat{y}_i^S)$ means the one-hot vector of $\hat{y}_i^S \in \hat{Y}_p^S$.

Datasets		Train	Dev	Test	Total
$\mathbb{S}_\mathrm{L}$	#PO	941	32	340	1,313
	#NT	446	4	169	619
	#NG	820	17	126	963
	#CF	41	1	16	58
$\mathbb{S}_\mathrm{R}$	#PO	3,262	126	1,490	4,878
	#NT	674	13	250	937
	#NG	1,205	46	500	1,751
	#CF	88	0	14	102
$\mathbb{S}_\mathrm{T}$	#PO	-			698
	#NT	-			2,254
	#NG	-			271

Table 1: Datasets from SemEval and Twitter.

2.3 Joint Loss

On the whole, the proposed framework DOER has two branches: one for ATE labeling and the other for ASC labeling. Each of them is differentiable, and thus can be trained with gradient descent. We equivalently use the negative of $L(W_c, b_c)$ in Eq. (6) as the error to do minimization via back-propagation through time (BPTT) (Goller and Kuchler, 1996). Thus, the loss is as follows:

$$\mathcal{L} = -\sum_i \log\ p\left(y|h; W_c, b_c\right), \tag{11}$$

Then, the losses from both tasks and the auxiliary tasks are constructed as the joint loss of the entire model:

$$\mathcal{J}(\Theta) = (\mathcal{L}_a + \mathcal{L}_p) + (\mathcal{L}_{u_A} + \mathcal{L}_{u_P} + \mathcal{L}_s) + \frac{\lambda}{2}\|\Theta\|^2, \tag{12}$$

where $\mathcal{L}_a$ and $\mathcal{L}_p$, which have the same formulation as Eq. (11), denote the loss for aspect term and polarity, respectively. Θ represents the model parameters containing all weight matrices W, U, v and bias vectors b. λ is a regularization parameter.

3 Experiments

3.1 Datasets

We conduct experiments on two datasets from the SemEval challenges and one English Twitter dataset. The details of these benchmark datasets are summarized in Table 1. $\mathbb{S}_\mathrm{L}$ comes from SemEval 2014 (Pontiki et al., 2014), which contains laptop reviews, and $\mathbb{S}_\mathrm{R}$ are restaurant reviews merged from SemEval 2014, SemEval 2015 (Pontiki et al., 2015), and SemEval 2016 (Pontiki et al., 2016). We keep the official data division of these datasets for the training set, validation set, and testing set. The reported results of $\mathbb{S}_{\mathrm{L}}$ and $\mathbb{S}_{\mathrm{R}}$ are averaged scores of 10 runs. $\mathbb{S}_{\mathrm{T}}$ consists of English tweets. Due to lack of standard train-test split, we report the ten-fold cross-validation results of $\mathbb{S}_{\mathrm{T}}$ as done in (Mitchell et al., 2013; Zhang et al., 2015; Li et al., 2019). For the auxiliary task of sentiment lexicon enhancement, we exploit a sentiment lexicon [2] to generate the label when training the model. The evaluation metric is F1 score based on the exact match of aspect term and its polarity.

3.2 Word Embeddings

To initialize the domain-specific word embeddings, we train the word embeddings by CBOW (Mikolov et al., 2013) using Amazon reviews[3] and Yelp reviews[4], which are in-domain corpora for laptop and restaurant respectively. Thus, for $\mathbb{S}_{\mathrm{L}}$, we use Amazon embedding, and for $\mathbb{S}_{\mathrm{R}}$, we use Yelp embedding. The Amazon review dataset contains 142.8M reviews, and the Yelp review dataset contains 2.2M restaurant reviews. The embeddings from all these datasets are trained by Gensim[5] which contains the implementation of CBOW. The parameter *min_count* is set to 10 and *iter* is set to 200. We use Amazon embedding as the domain-specific word embeddings of $\mathbb{S}_{\mathrm{T}}$ as Amazon corpora is large and comprehensive although not in the same domain. The general-purpose embeddings are initialized by Glove.840B.300d embeddings (Pennington et al., 2014). Its corpus is crawled from the Web.

3.3 Settings

In our experiments, the regularization parameter λ is empirically set as 0.001, and d_G and d_D as 300 and 100, respectively. The hidden state size of d of ReGU is 300. The hyperparameter K is set to 5. We use Adam (Kingma et al., 2014) as the optimizer with the learning rate of 0.001 and the batch size of 16. We also employ dropout (Srivastava et al., 2014) on the outputs of the embedding layer and two BiReGU layers. The dropout rate is 0.5. To avoid the exploding gradient problem, we clip the gradient norm within 5. The maximum number of epochs is set to 50. The word embeddings are fixed during the training process. We implemented DOER using the TensorFlow library (Abadi et al., 2016), and all computations are done on an NVIDIA Tesla K40 GPU.

3.4 Baseline Methods

To validate the performance of the proposed model DOER [6] on the aspect term-polarity co-extraction task, a comparative experiment is conducted with the following baseline models:

- **CRF-{pipelined, joint, collapsed}**: They leverage linguistically informed features with CRF to perform the sequence labeling task using the pipelined, joint, or collapsed approach[7] (Mitchell et al., 2013).
- **NN+CRF-{pipelined, joint, collapsed}**: An improvement of (Mitchell et al., 2013) that concatenates target word embedding and context four-word embeddings besides using linguistically informed features plus CRF to finish the sequence labeling task (Zhang et al., 2015). Instead of using the officially released code[8] due to the outdated library, we reproduce the results with the original settings.
- **Sentiment-Scope**: A collapsed CRF model[9] (Li and Lu, 2017), which expands the node types of CRF to capture sentiment scopes. The discrete features used in this model are exactly the same as the above two groups of models.
- **DE-CNN+TNet**: DE-CNN[10] (Xu et al., 2018) and TNet (Li et al., 2018) are the current state-of-the-art models for ATE and ASC, respectively. DE-CNN+TNet combines them in a pipelined manner. We use the official TNet-AS variant[11] as our TNet implementation.
- **LSTM+CRF-{LSTMc, CNNc}**: They all use BiLSTM plus CRF for sequence labeling.

[2] http://mpqa.cs.pitt.edu/ (the lexicon of (Hu and Liu, 2004) https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html can be used as well.

[3] http://jmcauley.ucsd.edu/data/amazon/

[4] https://www.yelp.com/academic_dataset

[5] https://radimrehurek.com/gensim/

[6] The code of DOER is available at https://github.com/ArrowLuo/DOER

[7] http://www.m-mitchell.com/code/

[8] https://github.com/SUTDNLP/OpenTargetedSentiment

[9] https://github.com/leodotnet/sentimentscope

[10] https://github.com/howardhsu/DE-CNN

[11] https://github.com/lixin4ever/TNet

	Model	$\mathbb{S}_{\mathrm{L}}$	$\mathbb{S}_{\mathrm{R}}$	$\mathbb{S}_{\mathrm{T}}$
Pipeline Baselines	CRF-pipeline	51.08	54.78	31.91
	NN+CRF-pipeline	53.36	60.78	45.08
	DE-CNN+TNet	56.47	67.54	48.74
Collapsed Baselines	CRF-collapsed	49.24	59.52	32.00
	NN+CRF-collapsed	50.64	61.74	45.52
	Sentiment-Scope	50.27	62.01	45.91
	LSTM+CRF-LSTMc	54.43	65.93	46.57
	LSTM+CRF-CNNc	54.71	66.36	47.35
	LM-LSTM-CRF	56.39	67.56	48.46
	E2E-TBSA	57.99	69.91	49.13
Joint Baselines	CRF-joint	50.73	59.75	32.42
	NN+CRF-joint	52.81	60.27	44.69
Ours	S-BiLSTM	56.83	71.22	48.94
	S-**BiReGU**	57.82	71.47	49.11
	S-**BiReGU+CSU**	58.99	72.19	49.89
	S-**BiReGU+CSU+AuL**	59.06	72.32	51.06
	S-**BiReGU+CSU+AuS**	60.11	72.64	51.13
	DOER	**60.35**	**72.78**	**51.37**

Table 2: F1 score (%) comparison of all systems for aspect term-polarity pair extraction.

The difference is that LSTM+CRF-LSTMc (Lample et al., 2016) encodes char embedding by BiLSTM, while LSTM+CRF-CNNc (Ma and Hovy, 2016) uses CNN.

- **LM-LSTM-CRF**: It is a language model enhanced LSTM-CRF model proposed in (Liu et al., 2018), which achieved competitive results on several sequence labeling tasks[12].
- **E2E-TBSA**: It is an end-to-end model of the collapsed approach proposed to address ATE and ASC simultaneously[13] (Li et al., 2019).
- **S-BiLSTM**: It is a stacked BiLSTM model with two layers that adopts the joint approach and has the same Embeddings, Interface, Joint Output layers as DOER.
- **S-BiReGU**: It is similar to S-BiLSTM but uses a ReGU cell instead of an LSTM cell.

We use two abbreviations AuL and AuS for the ablation study. **AuL** denotes the auxiliary task of aspect term length enhancement, and **AuS** denotes the auxiliary task of sentiment lexicon enhancement. All baselines have publicly available codes, and we ran these officially released codes to reproduce the baseline results except the NN+CRF variants due to the outdated library as discussed in the bullet point for these baseline systems.

3.5 Results and Analysis

Comparison Results. The comparison results are shown in Table 2, which are F1 scores of aspect term-polarity pairs. As the results show, our DOER obtains consistent improvement over baselines. Compared to the best pipelined model, the proposed framework outperforms DE-CNN+TNet by 3.88%, 5.24%, and 2.63% on $\mathbb{S}_{\mathrm{L}}$, $\mathbb{S}_{\mathrm{R}}$, and $\mathbb{S}_{\mathrm{T}}$, respectively. It indicates that an elaborated joint model can achieve better performance than pipeline approaches on aspect term-polarity co-extraction task. Besides, seven collapsed models are also introduced to the comparison. Compared to the best of these collapsed approaches, DOER improves by 2.36%, 2.87%, and 2.24% over E2E-TBSA on $\mathbb{S}_{\mathrm{L}}$, $\mathbb{S}_{\mathrm{R}}$, and $\mathbb{S}_{\mathrm{T}}$, respectively. This result shows the potential of a joint model which considers the interaction between the two relevant tasks. Comparing with existing works based on the joint approach, i.e., CRF-joint and NN+CRF-joint, DOER makes substantial gains over them as well. The improvements over DE-CNN+TNet and E2E-TBSA are statistically significant ($p < 0.05$).

Ablation Study. To test the effectiveness of

[12] https://github.com/LiyuanLucasLiu/LM-LSTM-CRF

[13] https://github.com/lixin4ever/E2E-TBSA

each component of DOER, we conduct an ablation experiment with results shown in the last block of Table 2. The fact that S-BiReGU gives superior performance compared to S-BiLSTM indicates the effectiveness of ReGU in our task. This residual architecture enables information transfer to the next layers more effective. With the help of CSU, S-BiReGU+CSU achieves better performance than without it. We believe the interaction of information between ATE and ASC is essential to improve each other. Although the samples with long aspect terms are rare, the auxiliary task of aspect term length can improve the performance. Another auxiliary task of sentiment lexicon can also enhance the representation of the proposed framework. As a whole of S-BiReGU, CSU, AuL, and AuS, the proposed DOER achieves superior performance. It mainly benefits from the enhanced features by the two auxiliary tasks and the interaction of two separate routes of ATE and ASC.

Results on ATE. Table 3 shows the results of aspect term extraction only. DE-CNN is the current state-of-the-art model on ATE as mentioned above. Comparing with it, DOER achieves new state-of-the-art scores. DOER* denotes the DOER without ASC part. As the table shows, DOER achieves better performance than DOER*, which indicates the interaction between ATE and ASC can yield better performance for ATE than only conduct a single task.

Model	$\mathbb{S}_{\mathrm{L}}$	$\mathbb{S}_{\mathrm{R}}$	$\mathbb{S}_{\mathrm{T}}$
DE-CNN	81.26	78.98	63.23
DOER*	82.11	79.98	68.99
DOER	82.61	81.06	71.35

Table 3: F1 score (%) comparison only for aspect term extraction.

Case Study. Table 4 shows some examples of S-BiLSTM, S-BiReGU+CSU, and DOER. As observed in the first and second rows, S-BiReGU+CSU and DOER predict the aspect term-polarity pair correctly but S-BiLSTM does not. With the constraint of CSU, the error words can be avoided as shown in the second row. The two auxiliary tasks work well on the CSU. They can capture a better sentiment representation, e.g., the third row, and alleviate the misjudgment on the long aspect terms, e.g., the last row.

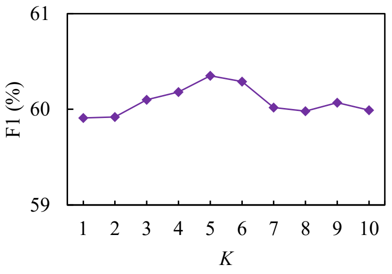
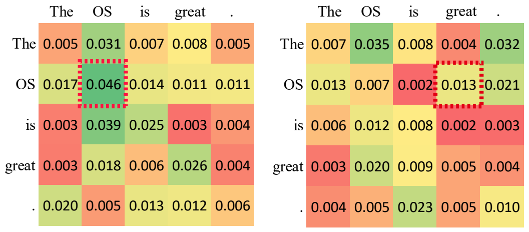


Figure 5: F1 scores on $\mathbb{S}_{\mathrm{L}}$ with different K.

Impact of K. We investigate the impact of hyperparameter K of the CSU on the final performance. The experiment is conducted on $\mathbb{S}_{\mathrm{L}}$ by varying K from 1 to 10 with the step of 1. As shown in Figure 5, value 5 is the best choice for the proposed method to address our task. Due to the performance demonstrated in the figure, K is set to 5 cross all experiments for simplicity.

Visualization of Attention Scores in CSU. We also try to visualize the attention scores S_A and S_P to explore the effectiveness of CSU. As shown in Figure 6, S_A and S_P have different values, which indicate that both ATE and ASC indeed interact with each other. The red dashed rectangle in Figure 6a shows that the model learns to focus on itself when labeling the word "OS" in the ATE task. Likewise, the red dashed rectangle in Figure 6b shows that the model learns to focus on the word "great" instead of itself when labeling the word "OS" in the ASC task. The fact that the polarity on the target aspect "OS" is positive, which is inferred from the "great", verifies that the system is doing the right job. In summary, we can conclude that the attention scores learned by CSU benefit the labeling process.



	The	OS	is	great	.
The	0.005	0.031	0.007	0.008	0.005
OS	0.017	0.046	0.014	0.011	0.011
is	0.003	0.039	0.025	0.003	0.004
great	0.003	0.018	0.006	0.026	0.004
.	0.020	0.005	0.013	0.012	0.006

(a) S_A

	The	OS	is	great	.
The	0.007	0.035	0.008	0.004	0.032
OS	0.013	0.007	0.002	0.013	0.021
is	0.006	0.012	0.008	0.002	0.003
great	0.003	0.020	0.009	0.005	0.004
.	0.004	0.005	0.023	0.005	0.010

(b) S_P

Figure 6: Visualization of S_A and S_P in CSU.

4 Related Work

Our work spans two major topics of aspect-based sentiment analysis: aspect term extraction and aspect sentiment classification. Each of them has

Input	S-BiLSTM	S-BiReGU+CSU	DOER
I like the [lighted screen]$_{\text{PO}}$ at night.	None (✗)	[lighted screen]$_{\text{PO}}$	[lighted screen]$_{\text{PO}}$
It is a great [size]$_{\text{PO}}$ and amazing [windows 8]$_{\text{PO}}$ included!	[size]$_{\text{PO}}$, [windows 8 included]$_{\text{PO}}$ (✗)	[size]$_{\text{PO}}$, [windows 8]$_{\text{PO}}$	[size]$_{\text{PO}}$, [windows 8]$_{\text{PO}}$
I tried several [monitors]$_{\text{NT}}$ and several [HDMI cables]$_{\text{NT}}$ and this was the case each time.	[HDMI cables]$_{\text{NG}}$ (✗)	None (✗), [HDMI cables]$_{\text{NT}}$	[monitors]$_{\text{NT}}$, [HDMI cables]$_{\text{NT}}$
The [2.9 ghz dual-core i7 chip]$_{\text{PO}}$ really out does itself.	[dual-core i7 chip]$_{\text{PO}}$ (✗)	[dual-core i7 chip]$_{\text{PO}}$ (✗)	[2.9 ghz dual-core i7 chip]$_{\text{PO}}$

Table 4: Case analysis on S-BiLSTM, S-BiReGU+CSU, and DOER. ✗ means wrong prediction.

been studied by many researchers. Hu and Liu (2004) extracted aspect terms using frequent pattern mining. Qiu et al. (2011) and Liu et al. (2015) proposed to use rule-based approach exploiting either hand-crafted or automatically generated rules about some syntactic relationships. Mei et al. (2007), He et al. (2011) and Chen et al. (2014) used topic modeling based on Latent Dirichlet Allocation (Blei et al., 2003). All of the above methods are unsupervised. For supervised methods, the ATE task is usually treated as a sequence labeling problem solved by CRF. For the ASC task, a large body of literature has tried to utilize the relation or position between the aspect terms and the surrounding context words as the relevant information or context for prediction (Tang et al., 2016a; Laddha and Mukherjee, 2016). Convolution neural networks (CNNs) (Poria et al., 2016; Li and Xue, 2018), attention network (Wang et al., 2016b; Ma et al., 2017; He et al., 2017), and memory network (Wang et al., 2018) are also active approaches.

However, the above methods are proposed for either the ATE or the ASC task. Lakkaraju et al. (2014) proposed to use hierarchical deep learning to solve these two subtasks. Wu et al. (2016) utilized cascaded CNN and multi-task CNN to address aspect extraction and sentiment classification. Their main idea is to directly map each review sentence into pre-defined aspect terms by using classification and then classifying the corresponding polarities. We believe the pre-defined aspect terms are in general insufficient for most analysis applications because they will almost certainly miss many important aspects in review texts.

This paper regards ATE and ASC as two parallel sequence labeling tasks and solves them simultaneously. Comparing with the methods that address them one by one using two separate models, our framework is easy to use in practical applications by outputting all the aspect term-polarity pairs of input sentences at once. Similar to our work, Mitchell et al. (2013) and Zhang et al. (2015) are also about performing two sequence labeling tasks, but they extract named entities and their sentiment classes jointly. We have a different objective and utilize a different model. Li et al. (2019) have the same objective as us. The main difference is that their approach belongs to a collapsed approach but ours is a joint approach. The model proposed by (Li and Lu, 2017) is also a collapsed approach based on CRF. Its performance is heavily dependent on manually crafted features.

5 Conclusion

In this paper, we introduced a co-extraction task involving aspect term extraction and aspect sentiment classification for aspect-based sentiment analysis and proposed a novel framework DOER to solve the problem. The framework uses a joint sequence labeling approach and focuses on the interaction between two separate routes for aspect term extraction and aspect sentiment classification. To enhance the representation of sentiment and alleviate the difficulty of long aspect terms, two auxiliary tasks were also introduced in our framework. Experimental results on three benchmark datasets verified the effectiveness of DOER and showed that it significantly outperforms the baselines on aspect term-polarity co-extraction.

Acknowledgments

This work is supported by the National Key R&D Program of China (No. 2017YFB1401401).

References

Martín Abadi, Ashish Agarwal, Paul Barham, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. In *JMLR*, pages 993–1022.

Zhiyuan Chen, Arjun Mukherjee, and Bing Liu. 2014. Aspect extraction with automated prior knowledge learning. In *ACL*, pages 347–358.

Christoph Goller and Andreas Kuchler. 1996. Learning task-dependent distributed representations by backpropagation through structure. In *ICNN*, pages 347–352.

Alex Graves and Jürgen Schmidhuber. 2005. Framewise phoneme classification with bidirectional LSTM and other neural network architectures. *Neural Networks*, 18(5-6):602–610.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*, pages 770–778.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2017. An unsupervised neural attention model for aspect extraction. In *ACL*, pages 388–397.

Yulan He, Chenghua Lin, and Harith Alani. 2011. Automatically extracting polarity-bearing topics for cross-domain sentiment classification. In *ACL*.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *KDD*, pages 168–177.

Niklas Jakob and Iryna Gurevych. 2010. Extracting opinion targets in a single-and cross-domain setting with conditional random fields. In *EMNLP*, pages 1035–1045.

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *ACL*.

Diederik Kingma, Jimmy Ba, Diederik Kingma, and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Abhishek Laddha and Arjun Mukherjee. 2016. Extracting aspect specific opinion expressions. In *EMNLP*, pages 627–637.

Himabindu Lakkaraju, Richard Socher, and Chris Manning. 2014. Aspect specific sentiment analysis using hierarchical deep learning. In *NIPS Workshop on Deep Learning and Representation Learning*.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*.

Hao Li and Wei Lu. 2017. Learning latent sentiment scopes for entity-level sentiment analysis. In *AAAI*, pages 3482–3489.

Tao Li and Wei Xue. 2018. Aspect based sentiment analysis with gated convolutional networks. In *ACL*, pages 2514–2523.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018. Transformation networks for target-oriented sentiment classification. In *ACL*, pages 946–956.

Xin Li, Lidong Bing, Piji Li, and Wai Lam. 2019. A unified model for opinion target extraction and target sentiment prediction. *AAAI*.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank Fangzheng Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *AAAI*, pages 5253–5260.

Qian Liu, Zhiqiang Gao, Bing Liu, and Yuanlin Zhang. 2015. Automated rule selection for aspect extraction in opinion mining. In *IJCAI*, pages 1291–1297.

Huaishao Luo, Tianrui Li, Bing Liu, Bin Wang, and Herwig Unger. 2018. Improving aspect term extraction with bidirectional dependency tree representation. *arXiv preprint arXiv:1805.07889*.

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

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. In *ACL*, pages 1064–1074.

Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*, pages 171–180.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Margaret Mitchell, Jacqui Aguilar, Theresa Wilson, and Benjamin Van Durme. 2013. Open domain targeted sentiment. In *EMNLP*, pages 1643–1654.

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

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *SemEval@NAACL-HLT*, pages 486–495.

Maria Pontiki, Dimitris Galanis, Haris Papageorgiou, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *SemEval@NAACL-HLT*, pages 19–30.

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

Soujanya Poria, Erik Cambria, and Alexander Gelbukh. 2016. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, 108:42–49.

Guang Qiu, Bing Liu, Jiajun Bu, and Chun Chen. 2011. Opinion word expansion and target extraction through double propagation. *Computational Linguistics*, 37(1):9–27.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *EMNLP*, pages 338–348.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, pages 1631–1642.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *JMLR*, 15(1):1929–1958.

Duyu Tang, Bing Qin, Xiaocheng Feng, and Ting Liu. 2016a. Effective lstms for target-dependent sentiment classification. In *COLING*, pages 3298–3307.

Duyu Tang, Bing Qin, and Ting Liu. 2016b. Aspect level sentiment classification with deep memory network. In *EMNLP*, pages 214–224.

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

Joachim Wagner, Piyush Arora, Santiago Cortes, Utsab Barman, Dasha Bogdanova, Jennifer Foster, and Lamia Tounsi. 2014. Dcu: Aspect-based polarity classification for semeval task 4. In *SemEval*.

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

Shuai Wang, Sahisnu Mazumder, Bing Liu, Mianwei Zhou, and Yi Chang. 2018. Target-sensitive memory networks for aspect sentiment classification. In *ACL*, pages 957–967.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016a. Recursive neural conditional random fields for aspect-based sentiment analysis. In *EMNLP*, pages 616–626.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *AAAI*, pages 3316–3322.

Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. 2016b. Attention-based lstm for aspect-level sentiment classification. In *EMNLP*.

Haibing Wu, Yiwei Gu, Shangdi Sun, and Xiaodong Gu. 2016. Aspect-based opinion summarization with convolutional neural networks. In *IJCNN*, pages 3157–3163.

Hu Xu, Bing Liu, Lei Shu, and Philip S. Yu. 2018. Double embeddings and cnn-based sequence labeling for aspect extraction. In *ACL*, pages 592–598.

Yichun Yin, Furu Wei, Li Dong, Kaimeng Xu, Ming Zhang, and Ming Zhou. 2016. Unsupervised word and dependency path embeddings for aspect term extraction. In *IJCAI*, pages 2979–2985.

Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015. Neural networks for open domain targeted sentiment. In *EMNLP*, pages 612–621.