Table 1: Comparison of different methods by their capability in utilizing interactive relations.

Interactive Relations		$\mathbf{R}_1$	$\mathbf{R}_2$	$\mathbf{R}_3$	$\mathbf{R}_4$
Separate/ Auxiliary	Wang et al. (2017)	✓	✗	✗	✗
	Xu et al. (2018)	✗	✗	✗	✗
	Li et al. (2018b)	✓	✗	✗	✗
	Hu et al. (2019)	✗	✗	✗	✗
Unified	Wang et al. (2018a)	✗	✗	✗	✗
	Li et al. (2019)	✗	✗	✗	✗
	Luo et al. (2019)	✗	✗	✗	✓
	He et al. (2019)	✗	✗	✓	✓
	Ours	✓	✓	✓	✓

- $\mathbf{R}_2$ indicates the triadic relation between SC and $\mathbf{R}_1$. One critical problem in SC is to determine the dependency between the aspect and its context. For example, the context "*small and cramped*" plays an important role in predicting the polarity of "*place*". Such a dependency is highly in accordance with $\mathbf{R}_1$ which emphasizes the interaction between the aspect and opinion terms. Hence SC and $\mathbf{R}_1$ can help refine the selection process for each other.
- $\mathbf{R}_3$ indicates the dyadic relation between SC and OE. The specific opinion terms generally convey specific polarities. For example, "*fantastic*" is often positive. The opinion terms extracted in OE should be paid more attention when predicting the sentiment polarity in SC.
- $\mathbf{R}_4$ indicates the dyadic relation between SC and AE. In the complete ABSA task, the aspect terms are unknown and SC will assign a polarity to every word. The aspect terms, e.g., "*place*" and "*food*", will have their corresponding polarities, while other words are considered as the background ones without sentiment. That is to say, the results from AE should be helpful in supervising the training of SC.

When reviewing the literature on the ABSA task, we find that existing separate methods either do not utilize any relations, or only utilize $\mathbf{R}_1$ by treating OE as an auxiliary task of AE. Meanwhile, the unified methods at most explicitly utilize $\mathbf{R}_3$ and $\mathbf{R}_4$. In view of this, we propose a novel Relation-Aware Collaborative Learning (RACL) framework to fully exploit the interactive relations in the complete ABSA task. We compare our model with existing methods by their capability in utilizing interactive relations in Table 1.

RACL is a multi-layer multi-task learning framework with a relation propagation mechanism to mutually enhance the performance of subtasks. For multi-task learning, RACL adopts the shared-private scheme (Collobert and Weston, 2008; Liu et al., 2017). Subtasks AE, OE, and SC first jointly train the low-level shared features, and then they train their high-level private features independently. In this way, the shared and private features can embed the task-invariant and task-oriented knowledge respectively. For relation propagation, RACL improves the model capacity by exchanging informative clues among three subtasks. Moreover, RACL can be stacked to multiple layers to perform collaborative learning at different semantic levels. We conduct extensive experiments on three datasets. Results demonstrate that RACL significantly outperforms the state-of-the-art methods for both the single subtasks and the complete ABSA task.

2 Related Work

Aspect-based sentiment analysis (ABSA) is first proposed by Hu and Liu (2004) and has been widely studied in recent years (Zhang et al., 2018). We organize existing studies by how the subtasks are performed and combined to perform ABSA.

Separate Methods Most existing studies treat ABSA as a two-step task containing aspect term extraction (AE) and aspect-based sentiment classification (SC), and develop separate methods for AE (Popescu and Etzioni, 2005; Wu et al., 2009; Li et al., 2010; Qiu et al., 2011; Liu et al., 2012; Chen et al., 2014; Chernyshevich, 2014; Toh and Wang, 2014; Vicente et al., 2015; Liu et al., 2015, 2016; Yin et al., 2016; Wang et al., 2016; Li and Lam, 2017; Clercq et al., 2017; He et al., 2017; Xu et al., 2018; Yu et al., 2019), and SC (Jiang et al., 2011; Mohammad et al., 2013; Kiritchenko et al., 2014; Dong et al., 2014; Vo and Zhang, 2015; Ma et al., 2017; Wang et al., 2018b; Zhu and Qian, 2018; Chen and Qian, 2019; Zhu et al., 2019), respectively. Some of them resort to the auxiliary task opinion term extraction (OE) and exploit their relation for boosting the performance of AE. For the complete ABSA task, results from two steps must be merged together *in a pipeline manner*. In this way, the relation between AE/OE and SC is totally neglected, and the errors from the upstream AE/OE will be propagated to the downstream SC. The overall performance of ABSA task is not promising for pipeline methods.

Unified Methods Recently, several studies attempt to solve ABSA task in a unified framework. The unified methods fall into two types: *collapsed tagging* (Mitchell et al., 2013; Zhang et al., 2015; Wang et al., 2018a; Li et al., 2019) and *joint training* (He et al., 2019; Luo et al., 2019). The former combines the labels of AE and SC to construct collapsed labels like {*B-senti, I-senti, O*}. The subtasks need to share all trainable features without distinction, which is likely to confuse the learning process. Moreover, the relations among subtasks cannot be explicitly modeled for this type of methods. Meanwhile, the latter constructs a multi-task learning framework where each subtask has independent labels and can have shared and private features. This allows the interactive relations among different subtasks to be modeled explicitly for the joint training methods. However, none of existing studies along this line has fully exploited the power of such relations.

We differentiate our work from aforementioned methods in that we propose a unified framework which exploits all dyadic and triadic relations among subtasks to enhance the learning capability.

3 Methodology

3.1 Task Definition

Given a sentence $S_e = \{w_1, ..., w_i, ..., w_n\}$, we formulate subtasks AE, OE, and SC as three sequence labeling problems.

- AE aims to predict a tag sequence $Y^A = \{y_1^A, ..., y_i^A, ..., y_n^A\}$ for aspect extraction, where $y_i^A \in \{B, I, O\}$ denotes the *beginning of, inside of*, and *outside of* an aspect term.
- OE aims to predict a tag sequence $Y^O = \{y_1^O, ..., y_i^O, ..., y_n^O\}$ for opinion extraction, where $y_i^O \in \{B, I, O\}$ denotes the *beginning of, inside of*, and *outside of* an opinion term.
- SC aims to predict a tag sequence $Y^S = \{y_1^S, ..., y_i^S, ..., y_n^S\}$ for sentiment classification, where $y_i^S \in \{pos, neu, neg\}$ denotes the *positive, neutral*, and *negative* sentiment polarities towards each word.

3.2 Model Architecture

Our proposed RACL is a unified multi-task learning framework which enables propagating the interactive relations (denoted as the same $\mathbf{R_1}..\mathbf{R_4}$ as those in Figure 1) for improving the ABSA performance, and it can be stacked to multiple layers to interact subtasks at different semantic levels. We present the overall architecture of RACL in Figure 2(a) and details of a single layer in Figure 2(b).

In particular, a single RACL layer contains three modules: AE, OE, and SC, where each module is designed for the corresponding subtask. These modules receive a shared representation of the input sentence, then encode their task-oriented features. After that, they propagate relations $\mathbf{R_1}..\mathbf{R_4}$ for collaborative learning by exchanging informative clues to further enhance the task-oriented features. Finally, three modules will make predictions for the corresponding tag sequences Y^A, Y^O, and Y^S based on the enhanced features.

In the following, we first illustrate the relation-aware collaborative learning in one layer, then show the stacking and the training of the entire RACL.

3.3 Relation-Aware Collaborative Learning

Input Word Vectors Given a sentence S_e, we can map the word sequence in S_e with either pre-trained word embeddings (e.g., GloVe) or pre-trained language encoders (e.g., BERT) to generate a sequence of word vectors $\mathbf{E}$=$\{\boldsymbol{e_1}, ..., \boldsymbol{e_i}, ...,\boldsymbol{e_n}\} \in \mathbb{R}^{d_w \times n}$, where d_w is the dimension of word vectors. We will examine the effects of these two types of word vectors in the experiments.

Multi-task Learning with Shared-Private Scheme To perform multi-task learning, different subtasks should focus on the different characteristics of a shared training sample. Inspired by the shared-private scheme (Collobert and Weston, 2008; Liu et al., 2017), we extract both the shared and private features to embed task-invariant and task-oriented knowledge for the AE, OE, and SC modules.

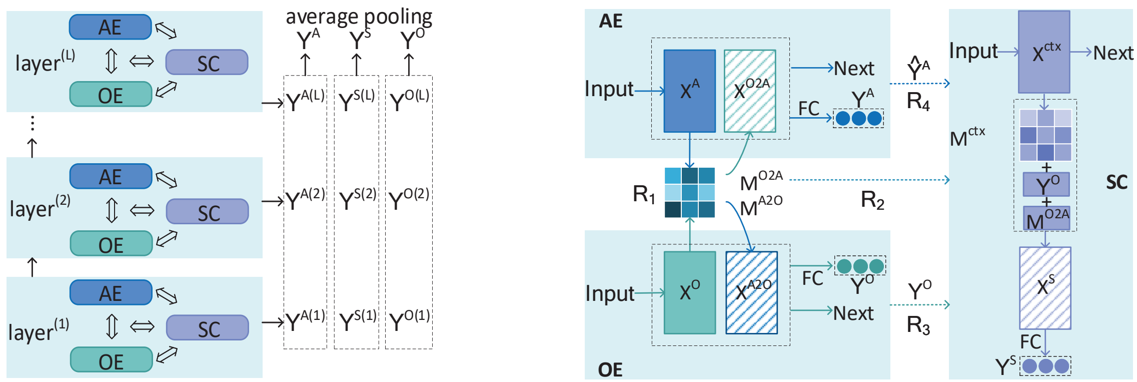
To encode the shared task-invariant features, we simply feed each $\boldsymbol{e_i}$ in $\mathbf{E}$ into a fully-connected layer and generate a transformed vector $\boldsymbol{h_i} \in \mathbb{R}^{d_h}$. We then obtain a sequence of shared vectors $\mathbf{H}$=$\{\boldsymbol{h_1}, ..., \boldsymbol{h_i}, ...,\boldsymbol{h_n}\} \in \mathbb{R}^{d_h \times n}$ for each sentence which will be jointly trained by all subtasks.

Upon the shared task-invariant features $\mathbf{H}$, the AE, OE, and SC modules will encode the task-oriented private features for the corresponding subtasks. We choose a simple CNN as the encoder function F due to its high computation efficiency.

For subtasks AE and OE, the key features for determining the existence of aspect and opinion terms are the representations of the original and adjacent words. Therefore, we construct two encoders to extract local *AE-oriented features* $\mathbf{X}^A$ and *OE-oriented features* $\mathbf{X}^O$:

$$\begin{aligned} F^A &: \mathbf{H} \rightarrow \mathbf{X}^A, \mathbf{X}^A \in \mathbb{R}^{d_c \times n}, \\ F^O &: \mathbf{H} \rightarrow \mathbf{X}^O, \mathbf{X}^O \in \mathbb{R}^{d_c \times n} \end{aligned} \tag{1}$$

For subtask SC, the process of feature generation is different from that in AE/OE. In order to determine the sentiment polarity towards an aspect term, we need to extract related semantic information from its context. The critical problem in SC is to determine the dependency between an aspect



(a) Architecture with L stacked RACL layers.

(b) Details of a single RACL layer.

Figure 2: The proposed RACL framework.

term and its context. Moreover, in the complete ABSA task, the aspect terms are unknown in SC and it needs to assign a polarity to every word in S_e. Based on these observations, we first encode the contextual features $\mathbf{X}^{ctx}$ from $\mathbf{H}$:

$$F^{ctx}: \mathbf{H} \rightarrow \mathbf{X}^{ctx}, \mathbf{X}^{ctx} \in \mathbb{R}^{d_h \times n} \qquad (2)$$

Then we treat the shared vector $\boldsymbol{h_i}$ as the query aspect and compute the semantic relation between the query and contextual features using the attention mechanism:

$$\begin{aligned} ds_{i,j}^{(i\neq j)} &= ((\boldsymbol{h_i})^T \times \mathbf{X}_j^{ctx}) \cdot [log_2(2+|i-j|)]^{-1}, \\ \mathbf{M}_{i,j}^{ctx} &= \frac{exp(ds_{i,j})}{\sum_{k=1}^{n} exp(ds_{i,k})}, \end{aligned} \qquad (3)$$

where $ds_{i,j}^{(i\neq j)}$ denotes the dependency strength between the i-th query word and the j-th context word, and $\mathbf{M}_{i,j}^{ctx}$ is the normalized attention weight of $ds_{i,j}^{(i\neq j)}$. We add a coefficient $[log_2(2+|i-j|)]^{-1}$ based on the absolute distance between two words. The rationale is that the adjacent context words should contribute more to the sentiment polarity. Finally, for the aspect query w_i, we can obtain the global *SC-oriented features* $\mathbf{X}_i^S$ by a weighted sum of all contextual features (except the one for w_i):

$$\mathbf{X}_i^S = \sum_{j=1}^{n} (\mathbf{M}_{i,j}^{ctx} \cdot \mathbf{X}_j^{ctx}) \qquad (4)$$

Propagating Relations for Collaborative Learning After encoding task-oriented features, we propagate the interactive relations ($\mathbf{R_1}..\mathbf{R_4}$) among subtasks to mutually enhance the AE, OE, and SC modules.

(1) $\mathbf{R_1}$ is the dyadic relation between AE and OE, which indicates that AE and OE might hold informative clues to each other. In order to model $\mathbf{R_1}$, we want the AE-oriented features $\mathbf{X}^A$ and the OE-oriented features $\mathbf{X}^O$ to exchange useful information based on their semantic relations. Take the subtask AE as an example, the semantic relation between the word in AE and that in OE is defined as follows:

$$\begin{aligned} sr_{i,j}^{(i\neq j)} &= (\mathbf{X}_i^A)^T \times \mathbf{X}_j^O, \\ \mathbf{M}_{i,j}^{O2A} &= \frac{exp(sr_{i,j})}{\sum_{k=1}^{n} exp(sr_{i,k})} \end{aligned} \qquad (5)$$

For the word w_i in AE, we can obtain the useful clues $\mathbf{X}_i^{O2A}$ from OE by applying a weighted sum of semantic relations to all words in OE (except the word w_i itself), i.e.,

$$\mathbf{X}_i^{O2A} = \sum\nolimits_{j=1}^{n} (\mathbf{M}_{i,j}^{O2A} \cdot \mathbf{X}_j^O) \qquad (6)$$

We then concatenate the original AE-oriented features $\mathbf{X}^A$ and the useful clues $\mathbf{X}^{O2A}$ from OE as the final features for AE, and feed them into a fully-connected layer to predict the tags of aspect terms:

$$Y^A = softmax(\mathbf{W}^A(\mathbf{X}^A \oplus \mathbf{X}^{O2A})), \qquad (7)$$

where $\mathrm{W}^A \in \mathbb{R}^{3\times 2d_c}$ is a transformation matrix, $\mathrm{Y}^A \in \mathbb{R}^{3\times n}$ is the predicted tag sequence of AE.

For subtask OE, we use the transposed matrix of $sr_{i,j}^{(i\neq j)}$ in Eq. 5 to compute the corresponding $\mathbf{M}^{A2O}$. In this way, the semantic relation between AE and OE will be consistent without regard to the direction. Then we can obtain the useful clues $\mathbf{X}^{A2O}$ from AE and generate the predicted tag sequence $Y^O \in \mathbb{R}^{3\times n}$ in a similar way, i.e.,

$$Y^O = softmax(\mathbf{W}^O(\mathbf{X}^O \oplus \mathbf{X}^{A2O})) \qquad (8)$$

Additionally, each w_i cannot be an aspect term and an opinion term at the same time, so we add a regularization hinge loss to constrain Y^A and Y^O:

$$\mathcal{L}^R = \sum\nolimits_{i=1}^{n} max(0, P_{y_i^A \in \{B,I\}} + P_{y_i^O \in \{B,I\}} - 1.0), \quad (9)$$

where P denotes the probability under the given conditions.

(2) $\mathbf{R_2}$ is the triadic relation between SC and $\mathbf{R_1}$. Remember that the dependency between the aspect term and its context is critical for subtask SC, and we have already calculated this dependency using the normalized attention weight $\mathbf{M}^{ctx}$. Hence we can model $\mathbf{R_2}$ by propagating $\mathbf{R_1}$ to $\mathbf{M}^{ctx}$. We use $\mathbf{M}^{O2A}$ as the representative of $\mathbf{R_1}$, and add it on $\mathbf{M}^{ctx}$ to denote the influence from $\mathbf{R_1}$ to SC. More formally, we define $\mathbf{R_2}$ as the following operation:

$$\mathbf{M}_{i,j}^{ctx} \leftarrow \mathbf{M}_{i,j}^{ctx} + \mathbf{M}_{i,j}^{O2A} \tag{10}$$

Actually, $\mathbf{M}^{O2A}$ characterizes the dependency between aspect terms and contexts in the view of term extraction while $\mathbf{M}^{ctx}$ characterizes it in the view of sentiment classification. The dual-view relation $\mathbf{R_2}$ can help refine the selection processes for both extraction and classification subtasks.

(3) $\mathbf{R_3}$ is the dyadic relation between SC and OE, which indicates that the extracted opinion terms should be paid more attention when predicting the sentiment polarity. In order to model $\mathbf{R_3}$, similarly to the method for $\mathbf{R_2}$, we update $\mathbf{M}^{ctx}$ in SC using the generated tag sequence Y^O from OE:

$$\mathbf{M}_{i,j}^{ctx} \leftarrow \mathbf{M}_{i,j}^{ctx} + P_{y_j^O \in \{B,I\}} \cdot [log_2(2+|i-j|)]^{-1} \tag{11}$$

By doing this, the opinion terms can get larger weights in the attention mechanism. Consequently, they will contribute more to the prediction of the sentiment polarity.

After getting the interacted values for $\mathbf{M}^{ctx}$, we can recompute the SC-oriented features $\mathbf{X}^S$ in Eq.4 accordingly. Then we concatenate $\mathbf{H}$ and $\mathbf{X}^S$ as the final features for SC and feed them into a fully-connected layer to predict sentiment polarities for the candidate aspect terms:

$$Y^S = softmax(\mathbf{W}^S(\mathbf{H} \oplus \mathbf{X}^S)), \tag{12}$$

where $\mathrm{W}^S \in \mathbb{R}^{3\times 2d_h}$ is a transformation matrix, $Y^S \in \mathbb{R}^{3\times n}$ is the predicted tag sequence of SC.

(4) $\mathbf{R_4}$ is the dyadic relation between SC and AE, which indicates that the results from AE are helpful in supervising the training of SC. Clearly, only aspect terms have sentiment polarities. Although SC needs to assign a polarity to every word, we know the ground truth aspect terms in AE during the training process. Therefore, we directly use the ground truth tag sequence $\hat{Y}^A$ of AE to refine the labeling process in SC. Specifically, only the predicted tags towards true aspect terms would be counted in the training procedure:

$$y_i^S \leftarrow \mathbf{I}(\hat{y}_i^A) \cdot y_i^S, \tag{13}$$

where $\mathbf{I}(\hat{y}_i^A)$ equals to 1 if w_i is an aspect term and to 0 if not. Notice that this approach is only used in the training procedure.

3.4 Stacking RACL to Multiple Layers

When using one single RACL layer, AE, OE, and SC modules only extract corresponding features in a relatively low linguistic level, which may be insufficient to serve as the evidence to label each word. Hence we stack RACL to multiple layers to obtain high-level semantic features for subtasks, which helps to conduct deep collaborative learning.

Specifically, we first encode features $\mathbf{X}^{ctx(1)}$, $\mathbf{X}^{A(1)} \oplus \mathbf{X}^{O2A(1)}$, and $\mathbf{X}^{O(1)} \oplus \mathbf{X}^{A2O(1)}$ in layer$^{(1)}$. Then in layer$^{(2)}$, we input these features for SC, AE, and OE to generate $\mathbf{X}^{ctx(2)}$, $\mathbf{X}^{A(2)}$, and $\mathbf{X}^{O(2)}$. In this way, we can stack RACL to L layers. We then conduct average pooling on results from all layers to obtain the final prediction:

$$Y^T = avg([Y^{T(1)}, Y^{T(2)}, ..., Y^{T(L)}]), \tag{14}$$

where $T \in \{A, O, S\}$ denotes the specific subtask, and L is the number of layers. This shortcut-like architecture can enforce the features in the low layers to be meaningful and informative, which in turn helps the high layers to make better predictions.

3.5 Training Procedure

After generating the tag sequences Y^A, Y^O, and Y^S for the sentence S_e, we compute the cross-entropy loss of each subtask:

$$\mathcal{L}^T = -\sum\nolimits_{i=1}^{n} \sum\nolimits_{j=1}^{J} \hat{y_{ij}^T} \cdot log(y_{ij}^T), \tag{15}$$

where $T \in \{A, O, S\}$ denotes the subtask, n is the length of S_e, J is the category of labels, y_i^T and $\hat{y_i^T}$ are the predicted tags and ground truth labels.

The final loss $\mathcal{L}$ of RACL is the combination of the loss for subtasks and the loss for regularization, i.e., $\mathcal{L} = \sum \mathcal{L}^T + \lambda \cdot \mathcal{L}^R$, where λ is a coefficient. We then train all parameters with back propagation.

4 Experiments

4.1 Datasets and Settings

Datasets We evaluate RACL on three real-world ABSA datasets from SemEval 2014 (Pontiki et al., 2014) and 2015 (Pontiki et al., 2015), which include reviews from two domains: restaurant and laptop. Original datasets only have ground truth labels for aspect terms and corresponding sentiment polarities, while labels for opinion terms are annotated by two previous works (Wang et al., 2016, 2017). All datasets have a fixed training/test split. We further randomly sample 20% training data as the development set to tune hyper-parameters, and only use the remaining 80% for training. The statistics for datasets are summarized in Table 2.

Table 2: The statistics of datasets.

Datasets	Type	Sentence	Aspect	Opinion
Restaurant14	train	3044	3699	3484
	test	800	1134	1008
Laptop14	train	3048	2373	2504
	test	800	654	674
Restaurant15	train	1315	1199	1210
	test	685	542	510

Settings We examine RACL with two types of word vectors: the *pre-trained word embedding* and *pre-trained language encoder*. In the word embedding implementation, we follow the previous studies (Xu et al., 2018; He et al., 2019; Luo et al., 2019) and use two types of embeddings, i.e., general-purpose and domain-specific embeddings. The former is from GloVe vectors with 840B tokens (Pennington et al., 2014), and the latter is trained on a large domain-specific corpus using fastText and published by Xu et al. (2018). Two types of embeddings are concatenated as the word vectors. In the language encoder implementation, we follow Hu et al. (2019) by using the BERT$_{Large}$ (Devlin et al., 2019) as the backbone and fine-tuning it during the training process. We denote these two implementations as **RACL-GloVe** and **RACL-BERT** [1].

For RACL-GloVe, we set the dimension d_w=400, d_h=400, d_c=256 and the coefficient λ=1e-5. Other hyper-parameters are tuned on the development set. The kernel size K of CNN and the layer number L is set to $\{3,3,5\}$ and $\{4,3,4\}$ for three datasets, respectively. We train the model for fixed epochs using Adam optimizer (Kingma and Ba, 2015) with learning rate 1e-4 and batch size 8. For RACL-BERT, we set d_w to 1024 and learning rate to 1e-5 for fine-tuning BERT, and other hyper-parameters are directly inherited from RACL-GloVe.

We use four metrics for evaluation, i.e., *AE-*F_1, *OE-*F_1, *SC-*F_1, and *ABSA-*F_1. The first three denote the F$_1$-score of each subtask, while the last one measures the overall performance for complete ABSA [2]. To compute ABSA-F$_1$, the result for an aspect term would be considered as correct only when both AE and SC results are correct. The model achieving the minimum loss on the development set is used for evaluation on the test set.

Baselines To demonstrate the effectiveness of RACL for the complete ABSA task, we compare it with the following *pipeline* and *unified* baselines. The hyper-parameters for baselines are set to the optimal values as reported in their papers.

- **{CMLA, DECNN}+ {TNet, TCap}**: CMLA (Wang et al., 2017) and DECNN (Xu et al., 2018) are the state-of-the-art methods for AE, while TNet (Li et al., 2018a) and T(rans)Cap (Chen and Qian, 2019) are the top-performing methods for SC. We then construct four *pipeline* baselines through combination.
- **MNN** (Wang et al., 2018a): is a *unified* method utilizing the collapsed tagging scheme for AE and SC.
- **E2E-ABSA** (Li et al., 2019): is a *unified* method using the collapsed tagging scheme for AE and SC, and it introduces the auxiliary OE task without explicit interaction.
- **DOER** (Luo et al., 2019): is a multi-task *unified* method which jointly trains AE and SC, and it explicitly models the relation $\mathbf{R_4}$.
- **IMN-D** (He et al., 2019): is a *unified* method involving joint training for AE and SC with separate labels. The OE task is fused into AE to construct five-class labels. It explicitly models relations $\mathbf{R_3}$ and $\mathbf{R_4}$ [3].
- **SPAN-BERT** (Hu et al., 2019): is a *pipeline* method using BERT$_{Large}$ as the backbone. A multi-target extractor is used for AE, then a polarity classifier is used for SC.
- **IMN-BERT**: is an extension of the best *unified* baseline IMN-D with BERT$_{Large}$. By doing this, we wish to conduct convincing comparisons for the BERT-style methods. The input dimension and learning rate of IMN-BERT are the same as our RACL-BERT, and other hyper-parameters are inherited from IMN-D .

4.2 Comparison Results

The comparison results for all methods are shown in Table 3. The methods are divided into three groups: M1$\sim$M4 are *GloVe-based pipeline* methods, M5$\sim$M9 are *GloVe-based unified* methods, and M10$\sim$M12 are *BERT-based* methods.

Firstly, among all GloVe-based methods (M1$\sim$M9), we can observe that RACL-GloVe consistently outperforms all baselines in terms of

[1] Our code and data are available at https://github.com/NLPWM-WHU/RACL.

[2] Following He et al. (2019), if an aspect term contains multiple words, we use the predicted sentiment of the first word as the SC result. Moreover, aspect terms with *conflict* sentiment labels are ignored when computing *SC-*F_1 and *ABSA-*F_1. The same goes for all baseline methods.

[3] For a fair comparison, we remove the auxiliary document-level datasets in TransCap and IMN-D, and only use the same aspect-level datasets as ours.

Table 3: Comparison of different methods. We separate the GloVe-based (M1∼M9) and BERT-based (M10∼M12) methods for a fair comparison. The best scores are in bold, and second best ones are underlined. Results of M5, M6 and M8 are taken from He et al. (2019), while other results are the average scores of 5 runs with random initialization. "-" denotes that the method does not contain the subtask OE.

Model		Restaurant14 (Res14)				Laptop14 (Lap14)				Restaurant15 (Res15)			
		AE-F_1	OE-F_1	SC-F_1	ABSA-F_1	AE-F_1	OE-F_1	SC-F_1	ABSA-F_1	AE-F_1	OE-F_1	SC-F_1	ABSA-F_1
M1	CMLA+TNet	81.91	83.84	69.69	64.49	77.49	76.06	68.30	55.94	67.73	70.56	62.27	55.00
M2	CMLA+TCap	81.91	83.84	71.32	65.68	77.49	76.06	69.49	56.30	67.73	70.56	63.32	55.47
M3	DECNN+TNet	82.79	-	70.45	65.80	79.38	-	68.69	57.39	68.52	-	62.41	55.69
M4	DECNN+TCap	82.79	-	71.77	66.84	79.38	-	69.61	57.71	68.52	-	63.60	56.22
M5	MNN	83.05	84.55	68.45	63.87	76.94	77.77	65.98	53.80	70.24	69.38	57.90	56.57
M6	E2E-TBSA	83.92	84.97	68.38	66.60	77.34	76.62	68.24	55.88	69.40	71.43	58.81	57.38
M7	DOER	84.63	-	64.50	68.55	80.21	-	60.18	56.71	67.47	-	36.76	50.31
M8	IMN-D	84.01	**85.64**	71.90	68.32	78.46	78.14	69.92	57.66	69.80	72.11	60.65	57.91
M9	RACL-GloVe	**85.37**	85.32	**74.46**	**70.67**	**81.99**	**79.76**	**71.09**	**60.63**	**72.82**	**78.06**	**68.69**	**60.31**
M10	SPAN-BERT	**86.71**	-	71.75	73.68	**82.34**	-	62.50	61.25	**74.63**	-	50.28	62.29
M11	IMN-BERT	84.06	85.10	75.67	70.72	77.55	**81.00**	**75.56**	61.73	69.90	73.29	70.10	60.22
M12	RACL-BERT	86.38	**87.18**	**81.61**	**75.42**	81.79	79.72	73.91	**63.40**	73.99	**76.00**	**74.91**	**66.05**

the overall metric ABSA-F_1, and achieves 2.12%, 2.92%, and 2.40% absolute gains over the strongest baselines on three datasets. The results prove that jointly training all subtasks and comprehensively modeling the interactive relations are critical for improving the performance of the complete ABSA task. Moreover, RACL-GloVe also achieves the best or second best results on all subtasks. This further demonstrates that the learning process of each subtask can be enhanced by the collaborative learning. Another observation from M1∼M9 is that the unified methods (M5∼M9) perform better than the pipeline ones (M1∼M4).

Secondly, among the GloVe-based unified methods, RACL-GloVe, IMN-D, and DOER perform better than MNN and E2E-TBSA in general. This can be due to the fact that the former three methods explicitly model interactive relations among subtasks while the latter two do not. We notice that DOER gets a poor SC-F_1 score. The reason might be that it utilizes an auxiliary sentiment lexicon to enhance the words with *"positive"* and *"negative"* sentiment. It is hard for DOER to handle words with *"neutral"* sentiment and this results in a low SC-F_1 score.

Thirdly, the BERT-based methods (M10∼M12) achieve a better performance than GloVe-based methods by utilizing the large-scale external knowledge encoded in the pre-trained BERT_{Large} backbone. Specifically, SPAN-BERT is a strong baseline in subtask AE by reducing the search space with a multi-target extractor. However, its performance on SC drops a lot because it cannot capture the dependency between the extracted aspect terms in AE and the opinion terms in SC without interactions among subtasks. IMN-BERT achieves relatively high scores on OE and SC, but its performance on AE is the worst among three without the guidance from the relations $\mathbf{R_1}$ and $\mathbf{R_2}$. In contrast, RACL-BERT gets significantly better overall scores than SPAN-BERT and IMN-BERT on all three datasets. This again shows the superiority of our RACL framework for the complete ABSA task by using all interactive relations.

5 Analysis

5.1 Ablation Study

To investigate the effects of different relations on RACL -GloVe/-BERT, we conduct the following ablation study. We sequentially remove each interactive relation and obtain four simplified variants.

As expected, all simplified variants in Table 4 have a performance decrease of ABSA-F_1. The results clearly demonstrate the effectiveness of the proposed relations. Moreover, we find that the relations play more important roles on small datasets than on large ones. The reason might be that it is hard to train a complicated model on small datasets, and the relations can absorb external knowledge from other subtasks.

Table 4: Ablation study. ↓ denotes a performance drop of RACL-GloVe/RACL-BERT.

	- $\mathbf{R_1}$	- $\mathbf{R_2}$	- $\mathbf{R_3}$	- $\mathbf{R_4}$
Res14	0.98/2.13↓	1.91/2.16↓	1.76/1.52↓	1.86/2.94↓
Lap14	1.05/1.44↓	0.96/0.59↓	2.08/0.44↓	2.17/2.23↓
Res15	1.88/5.19↓	1.15/3.72↓	1.82/4.33↓	2.74/6.46↓

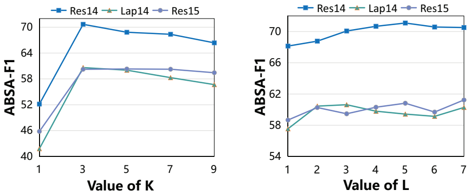
5.2 Effects of Hyper-Parameters

There are two key hyper-parameters in our model: the kernel size K of the CNN encoder and the layer number L. To investigate their impacts, we first vary K in the range of [1, 9] stepped by 2 while fixing L to the values in section 4.1, and then vary L in the range of [1, 7] stepped by 1 while fixing K.

Table 5: Case study. The columns "AE+SC" and "OE" denote the results generated by corresponding subtasks, where "None" denotes that no aspect/opinion terms are extracted. Words in blue and italic are annotated opinion terms, and those in red are annotated aspect terms with the subscripts denoting their sentiment polarities.

Examples	PIPELINE		IMN-D		RACL-GloVe	
	AE+SC	OE	AE+SC	OE	AE+SC	OE
S1. The [OS]$_{pos}$ is *easy*, and offers all kinds of surprises.	[OS]$_{pos}$	easy, offers(✗)	[OS]$_{pos}$	easy, offers(✗)	[OS]$_{pos}$	easy
S2. So much *faster* and *sleeker* [looking]$_{pos}$.	None(✗)	faster, sleeker	None(✗)	faster, sleeker	[looking]$_{pos}$	faster, sleeker
S3. [Dessert]$_{pos}$ was also to *die for*!	[Dessert]$_{neu}$(✗)	die for	[Dessert]$_{neu}$(✗)	die for	[Dessert]$_{pos}$	die for
S4. [Sushi]$_{pos}$ so *fresh* that it crunches in your mouth.	[Sushi]$_{neg}$(✗)	fresh	[Sushi]$_{pos}$	fresh	[Sushi]$_{pos}$	fresh

We only present the ABSA-F$_1$ results for RACL-GloVe in Figure 3 since the hyper-parameters of RACL-BERT are inherited from RACL-GloVe.



(a) Effects of K. (b) Effects of L.

Figure 3: Effects of hyper-parameters.

In Figure 3(a), K=1 yields extremely poor performance because the raw features are generated only by the current word. Increasing K to 3 or 5 can widen the receptive field and remarkably boosts the performance. However, when further increasing K to 7 or 9, many irrelevant words are added as noises and thus deteriorate the performance. In Figure 3(b), increasing L can, to some extent, expand the learning capability and achieve high performance. However, too many layers introduce excessive parameters and make the learning process over complicated.

5.3 Case Study

This section details the analysis on results of several examples by different methods for a case study. We choose CMLA+TCap (denoted as PIPELINE), IMN-D, and RACL-GloVe as three competitors. We do not include the BERT-based methods as we wish to investigate the power of the models without the external resources.

S1 and S2 verify the effectiveness of relation $\mathbf{R_1}$. In S1, due to the existence of the conjunction "*and*", two baselines incorrectly extract "*offers*" as an opinion term as "*easy*". In contrast, RACL-GloVe can successfully filter out "*offers*" in OE by using $\mathbf{R_1}$. The reason is that "*offers*" has never co-occured as an opinion term with the aspect term "*OS*" in the training set, and $\mathbf{R_1}$ which connects the AE and OE subtasks will treat them as irrelevant terms. This information will be passed to OE subtask during the testing phase. Similarly, in S2, both baselines fail to recognize "*looking*" as an aspect term, because it might be the present participle of "*look*" without opinion information. Instead, RACL-GloVe correctly labels it as $\mathbf{R_1}$ provides useful clues from opinion terms "*faster*" and "*sleeker*".

S3 shows the superiority of relation $\mathbf{R_2}$ which is critical to connect the three subtasks but has never been employed in previous studies. Both baselines successfully extract "*Dessert*" and "*die for*" for AE and OE, but assign the incorrect "*neutral*" sentiment polarity even if IMN-D has emphasized the opinion terms. The reason is that these two terms have not co-occurred in the training samples, and it is hard for SC to recognize their dependency. In contrast, since "*Dessert*" and "*die for*" are typical words in AE and OE, RACL-GloVe is able to encode their dependency in $\mathbf{R_1}$. By propagating $\mathbf{R_1}$ to SC using $\mathbf{R_2}$, RACL-GloVe can assign a correct polarity for "*Dessert*". To take a close look, we visualize the averaged predicted results (left) and the attention weights (right) of all layers in Figure 4. Clearly, the original attention $\mathbf{M}^{ctx-before}$ of "*Dessert*" does not concentrate on "*die for*". After getting enhanced by $\mathbf{M}^{O2A}$ and OE, $\mathbf{M}^{ctx-after}$ successfully highlights the opinion words and SC makes a correct prediction.

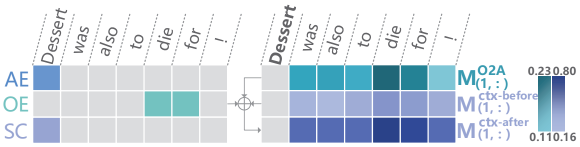


Figure 4: Visualization of the example S3.

S4 shows the benefits from relation $\mathbf{R_3}$. IMN-D and RACL-GloVe assign a correct polarity towards "*Sushi*" in SC since they both get the guidance from "*fresh*" in OE, while PIPELINE gets lost in contexts and makes a false prediction without the help of the opinion term. Notice that S1∼S4 simultaneously demonstrate the necessity for $\mathbf{R_4}$, since RACL-GloVe is not biased by background words and can make correct sentiment predictions in all examples.

5.4 Analysis on Computational Cost

To demonstrate that our RACL model does not incur the high computational cost, we compare it with two strong baselines DOER and IMN-D in terms of the parameter number and running time. We run three models on the Restaurant 2014 dataset with the same batch size 8 in a single 1080Ti GPU, and present the results in Table 6. Obviously, our proposed RACL has similar computational complexity with IMN-D, and they are both much simpler than DOER.

Table 6: Computational cost of different methods.

Model	Parameter Number	Runtime per Epoch
DOER	9,855,057	116s
IMN-D	4,129,713	5s
RACL-GloVe	5,087,568	5s

6 Conclusion

In this paper, we highlight the importance of interactive relations in the complete ABSA task. In order to exploit these relations, we propose a Relation-Aware Collaborative Learning (RACL) framework with multi-task learning and relation propagation techniques. Experiments on three real-world datasets demonstrate that our RACL framework with its two implementations outperforms the state-of-the-art pipeline and unified baselines for the complete ABSA task.

Acknowledgments

We thank the anonymous reviewers for their valuable comments. The work described in this paper is supported by the NSFC projects (61572376, 91646206), and the 111 project (B07037).

References

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

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *ACL*, pages 547–556.

Maryna Chernyshevich. 2014. IHS r&d belarus: Cross-domain extraction of product features using CRF. In *SemEval@COLING*, pages 309–313.

Orphée De Clercq, Els Lefever, Gilles Jacobs, Tijl Carpels, and Véronique Hoste. 2017. Towards an integrated pipeline for aspect-based sentiment analysis in various domains. In *WASSA@EMNLP*, pages 136–142.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*, pages 160–167.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186.

Li Dong, Furu Wei, Chuanqi Tan, Duyu Tang, Ming Zhou, and Ke Xu. 2014. Adaptive recursive neural network for target-dependent twitter sentiment classification. In *ACL*, pages 49–54.

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.

Ruidan He, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier. 2019. An interactive multi-task learning network for end-to-end aspect-based sentiment analysis. In *ACL*, pages 504–515.

Minghao Hu, Yuxing Peng, Zhen Huang, Dongsheng Li, and Yiwei Lv. 2019. Open-domain targeted sentiment analysis via span-based extraction and classification. In *ACL*, pages 537–546.

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

Long Jiang, Mo Yu, Ming Zhou, Xiaohua Liu, and Tiejun Zhao. 2011. Target-dependent twitter sentiment classification. In *ACL*, pages 151–160.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In *SemEval@COLING*, pages 437–442.

Fangtao Li, Chao Han, Minlie Huang, Xiaoyan Zhu, Yingju Xia, Shu Zhang, and Hao Yu. 2010. Structure-aware review mining and summarization. In *COLING*, pages 653–661.

Xin Li, Lidong Bing, Wai Lam, and Bei Shi. 2018a. 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. In *AAAI*, pages 6714–6721.

Xin Li, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang. 2018b. Aspect term extraction with history attention and selective transformation. In *IJCAI*, pages 4194–4200.

Xin Li and Wai Lam. 2017. Deep multi-task learning for aspect term extraction with memory interaction. In *EMNLP*, pages 2886–2892.

Kang Liu, Liheng Xu, and Jun Zhao. 2012. Opinion target extraction using word-based translation model. In *EMNLP*, pages 1346–1356.

Pengfei Liu, Shafiq R. Joty, and Helen M. Meng. 2015. Fine-grained opinion mining with recurrent neural networks and word embeddings. In *EMNLP*, pages 1433–1443.

Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. In *ACL*, pages 1–10.

Qian Liu, Bing Liu, Yuanlin Zhang, Doo Soon Kim, and Zhiqiang Gao. 2016. Improving opinion aspect extraction using semantic similarity and aspect associations. In *AAAI*, pages 2986–2992.

Huaishao Luo, Tianrui Li, Bing Liu, and Junbo Zhang. 2019. DOER: dual cross-shared RNN for aspect term-polarity co-extraction. In *ACL*, pages 591–601.

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

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

Saif Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. 2013. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *NAACL-HLT*, pages 321–327.

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*, pages 486–495.

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

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *EMNLP*, pages 339–346.

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.

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

Zhiqiang Toh and Wenting Wang. 2014. DLIREC: aspect term extraction and term polarity classification system. In *SemEval@COLING*, pages 235–240.

Iñaki San Vicente, Xabier Saralegi, and Rodrigo Agerri. 2015. Elixa: A modular and flexible ABSA platform. In *SemEval@NAACL-HLT*, pages 748–752.

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

Feixiang Wang, Man Lan, and Wenting Wang. 2018a. Towards a one-stop solution to both aspect extraction and sentiment analysis tasks with neural multi-task learning. In *IJCNN*, pages 1–8.

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

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. 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.

Yuanbin Wu, Qi Zhang, Xuanjing Huang, and Lide Wu. 2009. Phrase dependency parsing for opinion mining. In *EMNLP*, pages 1533–1541.

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.

Jianfei Yu, Jing Jiang, and Rui Xia. 2019. Global inference for aspect and opinion terms co-extraction based on multi-task neural networks. *TASLP*, 27(1):168–177.

Lei Zhang, Shuai Wang, and Bing Liu. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.*, 8(4).

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

Peisong Zhu, Zhuang Chen, Haojie Zheng, and Tieyun Qian. 2019. Aspect aware learning for aspect category sentiment analysis. *TKDD*, 13(6):55:1–55:21.

Peisong Zhu and Tieyun Qian. 2018. Enhanced aspect level sentiment classification with auxiliary memory. In *COLING*, pages 1077–1087.