Information Sciences 586 (2022) 209–223

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Relation construction for aspect-level sentiment classification

Jiandian Zeng [a], Tianyi Liu [b], Weijia Jia [b,c], Jiantao Zhou [a,*]

[a] State Key Laboratory of Internet of Things for Smart City, Department of Computer and Information Science, University of Macau, China
[b] School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China
[c] BNU-UIC Joint AI Research Institute, Beijing Normal University, Guangdong, China

ARTICLE INFO

*Article history:*
Received 28 December 2020
Received in revised form 22 November 2021
Accepted 26 November 2021
Available online 2 December 2021

*Keywords:*
Aspect-level
Sentiment analysis
Aspect relations
Graph convolutional network

ABSTRACT

Aspect-level sentiment classification aims to obtain fine-grained sentiment polarities of different aspects in one sentence. Most existing approaches handle the classification by acquiring the importance of context words towards each given aspect individually, and ignore the benefits brought by aspect relations. Since the sentiment of one aspect can be deduced through their relationship according to other aspects, in this paper, we propose a novel relation construction multi-task learning network (RMN), which is the first attempt to extract aspect relations as an auxiliary classification task. RMN generates aspect representations through graph convolution networks with a semantic dependency graph and utilizes the bi-attention mechanism to capture the relevance between the aspect and the context. Unlike conventional multi-task learning methods that need extra datasets, we construct an auxiliary relation-level classification task that extracts aspect relations from the original dataset with shared parameters. Extensive experiments on five public datasets from SemEval 14, 15, 16 and MAMS show that our RMN improves about 0.09% to 0.8% on accuracy and about 0.04% to 1.19% on F1 score, compared to several comparative baselines.


## 1. Introduction

As an important Nature Language Processing (NLP) task, sentiment classification is widely applied in online review analysis [1], text mining [2] and emotion recognition [3]. Document-level [4,5] and sentence-level sentiment analyses [6,7] aim to predict the overall sentiment of the input text, which can be considered as coarse-grained sentiment. Given a set of aspects in one sentence, aspect-level sentiment analysis aims at identifying sentiment polarities towards these specific entities [8]. For example, in the sentence "*The menu is limited but almost all of the dishes are excellent*", although the overall sentiment of the sentence is *neutral*, the polarity of "*menu*" is *negative* while the one for "*dish*" is *positive*.

For the aspect-level sentiment analysis, traditional methods mainly constructed sentiment lexicon dictionary [9] or adopted a feature-based SVM [10] for classification. However, such kind of lexicon dictionary and feature-based methods are labor-intensive. Recently, with the ability to handle complex structure sentences, neural network-based approaches have become the mainstream of sentiment classification. Besides, the attention mechanism has been proved effective for capturing potential semantic relations between the aspect and the context [8,11,12]. Apart from the attention mechanism, Zhang et al. [13] applied Graph Convolutional Networks (GCNs) to extract syntactical dependencies among the context, and presented its advantages when handling long sentences.

* Corresponding author.
*E-mail addresses:* yb87470@um.edu.mo (J. Zeng), liutianyi@sjtu.edu.cn (T. Liu), jiawj@sjtu.edu.cn (W. Jia), jtzhou@um.edu.mo (J. Zhou).

https://doi.org/10.1016/j.ins.2021.11.081

However, the above studies all associated a single aspect with its contextual words individually, and ignored the benefits brought by multiple relations among aspects. One aspect can be better classified when considering its aspect relations. For instance, as shown in Fig. 1, given the sentence "*Nice food. The price is reasonable although the service is poor.*", we can obtain that "*food*" and "*price*" are in *positive* semantic while "*service*" is *negative*. Meanwhile, three relations are extracted based on three aspects, namely "*food-price*", "*food-service*" and "*price-service*". Suppose that the semantic word "*reasonable*" is masked, and the polarity of "*price*" is unknown. If taken relations "*food-price*" and "*price-service*" into consideration, we can easily deduce that the polarity of "*price*" is *negative*. Therefore, for the sake of better aspect-level sentiment analysis, two challenges should be addressed: 1) how to model relations among aspects; and 2) how to utilize these aspect relations for the aspect-level sentiment classification.

For the first challenge, several methods [14,15] adopted the attention mechanism or GCNs over aspects to capture potential relations. However, they exaggerated aspect information since the aspect with neutral polarity may generate noises if considered. To address this issue, we only consider explicit aspect relations ("*similar*" and "*opposite*") to avoid neutral aspect noises. Specifically, we model aspect relations by the subtraction between two aspect representations, then auto-annotate their labels based on aspect polarities. For the second challenge, we construct an auxiliary task named relation-level classification, and train the basic aspect classification task and the auxiliary task simultaneously in a multi-task framework to explore the benefits of the aspect relations. Our contributions are summarized as follows:

- To the best of our knowledge, this is the first attempt to extract aspect relations as an auxiliary task for aspect-level sentiment analysis, and we explore the benefits brought by aspect relations.
- We propose a novel relation construction multi-task learning network that utilizes the bi-attention mechanism to capture bidirectional semantic information between the context and the aspect, and we adopt aspect disagreement regularization to better identify aspect-specific features from overlapped representations.
- Extensive experiments on several benchmark datasets validate the effectiveness of the proposed model compared to several comparative baselines, and show the ability to handle relatively small datasets.

The rest of this paper is organized as follows: Section 2 reviews related works. The detailed modules of the proposed method are presented in Section 3. Experimental results and further analyses are shown in Section 4. Section 5 concludes the paper and discusses the future research.

## 2. Related works

### 2.1. Sentiment classification

Sentiment analysis has been served as an essential role in NLP tasks, and can be divided into three levels: document-level [4,5], sentence-level [7,16], and aspect-level [17–19]. In the document-level sentiment classification, Dou et al. [4] proposed a deep memory network to predict the sentiment polarity of a whole document. In the sentence-level classification, Liu et al. [20] investigated domain representations of multitask learning for the multi-domain sentiment analysis towards each sentence. Besides, a three-way enhanced convolutional neural network model was proposed in [21] to make the sentence-level sentiment decision.

Unlike the document-level and the sentence-level that predict the overall sentiment towards the whole text, the aspect-level sentiment analysis aims to identify the polarities of different aspects based on their contextual words [22,23]. Previous studies [8,24,25] mainly utilized LSTMs to explore potential connections between an aspect and its context. Liu et al. [26] proposed a bidirectional attention mechanism to capture the semantic relevance from both directions. Though the attention

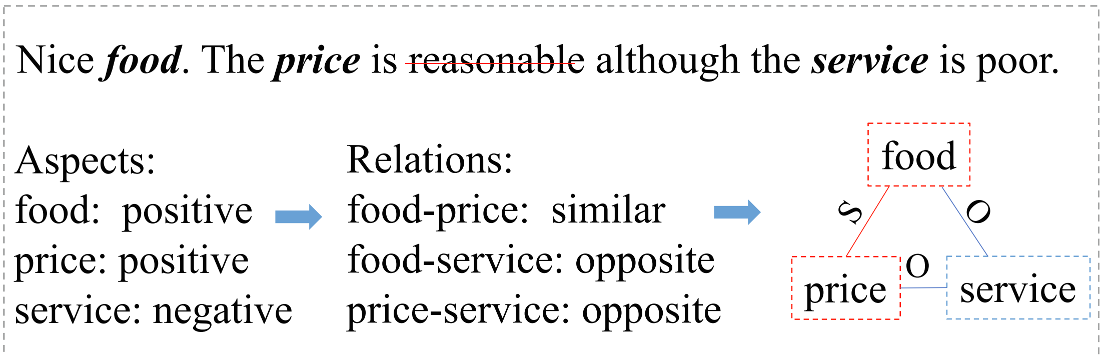


**Fig. 1.** An example for illustrating multiple relations among aspects: 1) similar relation means that two aspects are similar semantics. 2) opposite relation means that two aspects are opposite semantics.

mechanism has been proved to be useful, some unrelated words may be mistakenly selected for calculating the weights for a given aspect. To solve the issue, Zhang et al. [13] applied GCNs to obtain long-away aspect-relevant information, showing the effectiveness of using semantic graph when generating aspect representations.

However, the above methods handled each given aspect individually, and neglected the relations among multiple aspects. Recently, He et al. [27] learned the probability distribution over aspects and applied syntax-based attention mechanism to generate aspect-specific representations. Hazarika et al. [28] utilized inter-aspect dependencies along with temporal dependency processing of their corresponding sentence representations. In addition to the attention mechanism, graph-based structures [15,29] were adopted. Zhao et al. [15] applied multi-GCNs to model the dependencies of aspects. Wang et al. [29] proposed a relational graph attention network to model connections between aspects and opinion words. Zhang et al. [30] designed a lexical graph to capture the global word co-occurrence information in training samples, and considered different types of relations in syntactic graphs. Besides, Yuan et al. [31] assigned different weights to edges in a graph, and applied a graph attention network with memory fusion to learn and exploit multi-word relations.

Although the above graph-based methods are promising, the uncertainly relationships between the neutral aspect and other aspects may generate noises and lead to ambiguous semantic of the classification.

### *2.2. Multi-task learning*

By learning different tasks synchronously with shared representations, Multi-Task Learning (MTL) can improve model generalization by capturing correlations between related tasks. In recent years, MTL has been applied in many NLP tasks [32,34]. Liu et al. [34] applied MTL for the implicit discourse relation classification by synthesizing the discourse analysis tasks within different corpora. He et al. [32] explored the knowledge from document-level corpus for training aspect-level sentiment classifiers. Besides, a message passing mechanism was applied in [35] to learn interactions between the aspect and the opinion term co-extraction. Chen et al. [33] utilized the abundant document-labeled data and developed a transfer learning framework to transfer knowledge from the document-level task to the aspect-level task. However, the above methods needed extra corpus for training the auxiliary task.

Table 1 summaries the selected DNN models. As will be clear soon, our model differs from the above approaches in several ways: 1) we construct more specific aspect relations for auxiliary sentiment classification; and 2) our model needs no extra corpus since it can extract relations from the original datasets with shared parameters.

## 3. Problem definition and methodology

In this section, we first give the problem definition and the associated notations. Afterwards, we introduce the detailed methodology and workflows. The overall architecture of our RMN is presented in Fig. 2, where aspect representations are extracted from $L$-layer GCNs, context representations are encoded by position encoder, and the final aspect-specific representations are generated by the bi-attention module.

### *3.1. Task definition and annotation*

Taking aspect relations into consideration, we define two tasks: *aspect-level classification* task and *relation-level classification* task.

**Table 1**
Summary of the selected DDN models.

| Method | RNN | CNN | Attention | Graph | Multi-tasks | Datasets |
|---|---|---|---|---|---|---|
| TD-LSTM [24] | ✓ | | ✓ | | | Twitter |
| ATAE-LSTM [8] | ✓ | | ✓ | | | SemEval |
| AB-LSTM [25] | ✓ | | ✓ | | | Twitter |
| ABCDM [16] | ✓ | ✓ | ✓ | | | Twitter, Review |
| BILSTM-ATT-G [26] | ✓ | | ✓ | | | Review |
| ASGCN [13] | ✓ | | ✓ | ✓ | | Twitter, SemEval |
| He et al. [27] | ✓ | | ✓ | | | SemEval |
| Hazarika et al. [28] | ✓ | | ✓ | | | SemEval |
| ADGCN [15] | ✓ | | ✓ | ✓ | | SemEval |
| R-GAT [29] | ✓ | | ✓ | ✓ | | Twitter, SemEval |
| Bi-GCN [30] | ✓ | | ✓ | ✓ | | Twitter, SemEval |
| PRET + MULT [32] | ✓ | | ✓ | | ✓ | SemEval |
| Transcap [33] | ✓ | | ✓ | | ✓ | SemEval |
| RMN (ours) | ✓ | | ✓ | ✓ | ✓ | SemEval, MAMS |

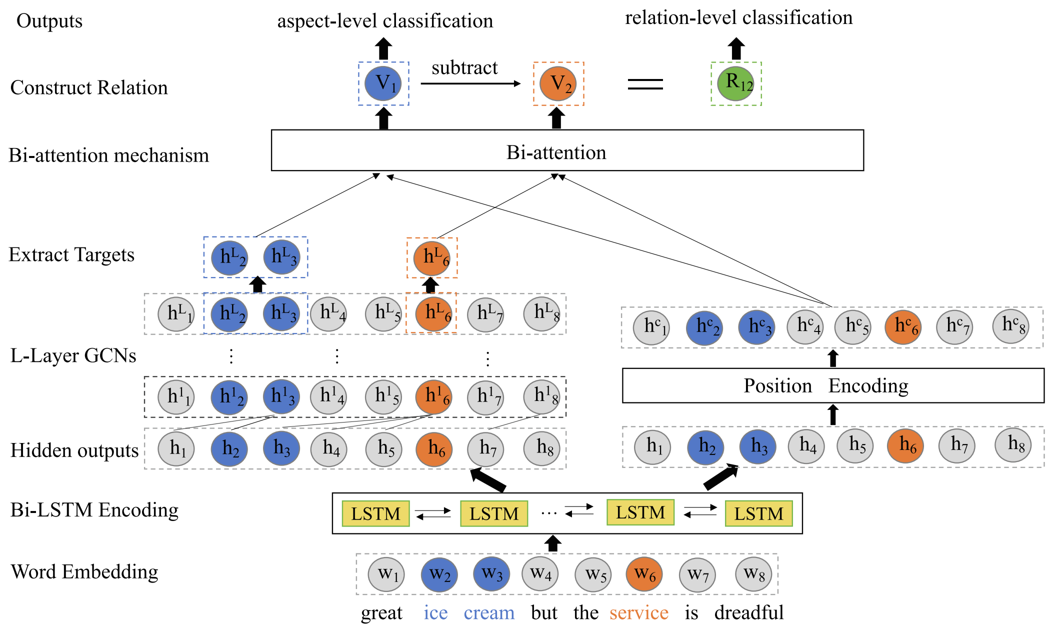


**Fig. 2.** The architecture of RMN. The input sentence is duplicated parallelly after Bi-LSTM encoding: one goes through *L*-layer GCNs for aspect representations, the other one generates the context representation with position encoding. Then, the aspect-specific representation is obtained for aspect-level classification by the bi-attention module. Finally, the aspect relation is obtained for relation-level classification by the subtraction of two aspect-specific representations.

#### 3.1.1. Aspect-level classification task

Given a sentence $S = [w_1, w_2, \ldots, w_n]$ with $n$ words and $m$ aspects $T = [A_1, A_2, \ldots, A_m]$, where the $i$-th aspect $A_i = [w_j, w_{j+1}, \ldots, w_k]$ is the sub-sequence of $S$. The task is to classify sentiment polarity (*negative, neutral, positive*) of each aspect $A_i$.

#### 3.1.2. Relation-level classification task

For this task, we generally define three relations, namely *"similar"*, *"unrelated"* and *"opposite"*. As shown in Table 2, the relation label is auto-annotated based on the polarities of two aspects, and no extra corpus is needed since all relations are extracted from the original dataset. For *"unrelated"* relation, it means that there is at least one *neutral* aspect. The task is to classify aspect relations according to their polarities in one sentence.

### 3.2. Word encoding

Distributed representations of words are unambiguous to express the semantic meanings in many NLP tasks [8,26], which provides convenience to the neural network to perform mathematical operations. Formally, given a sentence $S = [w_1, w_2, \ldots, w_n]$, we employ the pre-trained model, GloVe [36], to map the $i$-th word vector $w_i \in \mathbb{R}^d$, where $d$ represents the embedding dimension.

To further encode relation features inside the context, we adopt the Bidirectional Long Short-Term Memory network(Bi-LSTM) [37] as our basic relation encoder, which can access the future context as well as the past. In this way, each sentence is fed into Bi-LSTM to extract contextual information, where the forward hidden state $\overrightarrow{h} \in \mathbb{R}^{d'}$ and the backward hidden state

**Table 2**
Relation Annotation.

| Aspect 1 | Aspect 2 | Relation |
|---|---|---|
| negative | negative | similar |
| neutral | neutral | similar |
| positive | positive | similar |
| negative | positive | opposite |
| negative | neutral | unrelated |
| positive | neutral | unrelated |

$\overleftarrow{h} \in \mathbb{R}^{d'}$, and $d'$ is the dimension of hidden state. Then, the encoded representation of each word is formed by concatenating two hidden states:

$$h_i = \left[\overrightarrow{h_i} || \overleftarrow{h_i}\right], \; h_i \in \mathbb{R}^{2d'}, \tag{1}$$

where || denotes the vertical concatenating operation. Therefore, the sentence with Bi-LSTM encoding can be denoted as: $S = [h_1, h_2, \ldots, h_n]$.

### 3.3. GCNs for aspect representations

Previous studies [8,26,15] fed aspect words into LSTM networks or average over LSTM hidden outputs to obtain aspect representations, which will lose syntactically relevant words to the aspect, especially in long sentences. Inspired by Zhang et al. [13], we take the syntactic structure of sentence into consideration and apply GCNs with dependency graph to acquire aspect representations.

With the ability to handle the graph-structured data, GCNs perform well on acquiring neighboring information. As Fig. 3 presents, given a dependency graph[1] on one sentence, we can obtain the corresponding undirectional adjacency matrix $A \in \mathbb{R}^{n \times n}$. In one layer GCN, the $i$-th node $h_i$ is updated by:

$$\begin{aligned} h_i^{(1)} &= ReLU\left(\sum_{j=1}^{n}(A_{ij} + I_{ij})Wh_j + b\right) \\ &= ReLU\left(\sum_{j=1}^{n}A_{ij}Wh_j + Wh_i + b\right), \end{aligned} \tag{2}$$

where $W \in \mathbb{R}^{2d' \times 2d'}$ is a weight matrix, $b \in \mathbb{R}^{2d'}$ is a bias term, and $I \in \mathbb{R}^{n \times n}$ is the identity matrix considering self-connections.

With $L$-layer GCNs, features from $L$ hops away can be propagated to a target node. In this way, the syntactically relevant words can be transformed through the dependency graph based GCNs. The representation of the $i$-th node in the $l$-th layer can be denoted as:

$$h_i^{(l)} = ReLU\left(\sum_{j=1}^{n}A_{ij}W^{l-1}h_j^{l-1} + Wh_i^{l-1} + b^{l-1}\right), \tag{3}$$

Based on $h_i^l$, the $i$-th aspect representation can be obtained through its original sequence index:

$$A_i = \left[h_j^l, h_{j+1}^l, \ldots, h_k^l\right], 0 \leqslant j \leqslant k \leqslant n, \tag{4}$$

where $j$ and $k$ are the starting and ending indexes of the aspect.

### 3.4. Position encoding for context representations

Intuitively, a closer context word has bigger influence on aspect words [38]. For instance, given the sentence "*The food is decent though not worth the price*", the emotional word "*decent*" should be assigned larger weight for "*food*" while less weight for "*price*". We adopt the absolute distance as a metric and assign larger weights to those closer words. Specifically, we first calculate the position weight $p_i$ of the $i$-th word:

$$p_i = \begin{cases} 1 - \frac{j-i}{n}, & -s < i - j < 0 \\ 1, & j \leqslant i \leqslant k \\ 1 - \frac{i-k}{n}, & 0 < i - k < s \\ 0, & otherwise \end{cases} \tag{5}$$

where $s$ is the pre-defined distance parameter, and $n$ is the length of sentence. Then, the context representation can be calculated as:

$$h_i^c = p_i \cdot h_i, \tag{6}$$

At last, we can obtain the corresponding context representation: $C = [h_1^c, h_2^c, \ldots, h_n^c]$.

[1] Parsed by Spacy: https://spacy.io/.

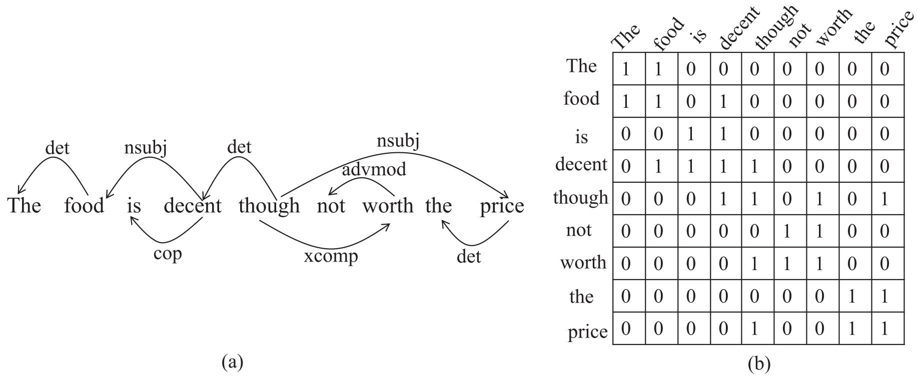


**Fig. 3.** An illustration of the dependency graph. (a) The universal dependencies of sentence, where directed edges between words mean that they are syntactically related. (b) The corresponding adjacent matrix, where '1' denotes two connected words.

### 3.5. Bidirectional matching mechanism

In this paper, we adopt the bidirectional matching mechanism to capture multiple semantic information between the aspect and the context. As can be seen in Fig. 4, we first calculate the matching matrix:

$$D_{ij} = C_i * A_j^T, D_{ij} \in \mathbb{R}^{n \times m}, \tag{7}$$

where $m$ and $n$ are the lengths of aspect and context representation. Each element in $D_{ij}$ denotes the importance of semantically matching between the context and the aspect. Then, *context to aspect attention* and *aspect to context attention* modules are applied to explore the interactive information.

**Context to Aspect Attention.** With aspect word representations, the attention mechanism aims to generate attention vector using the column-wise weight. Specifically, the column-wise weight is calculated by:

$$\begin{aligned} f_t &= tanh\left(D_{ij}^T \cdot W_t\right) \\ a_t &= softmax\left(w_t \cdot f_t^T\right), \end{aligned} \tag{8}$$

where $W_t \in \mathbb{R}^{n \times 2d'}$ and $w_t \in \mathbb{R}^{2d'}$ are weight matrices. Then, the context to aspect attention vector $V_t$ is obtained:

$$V_t = a_t \cdot A_j. \tag{9}$$

The aspect information with similar semantic to the context is addressed by assigning higher weights in $a_t$.

**Aspect to Context Attention.** Similar to the *context to aspect attention*. The row-wise weight is calculated by:

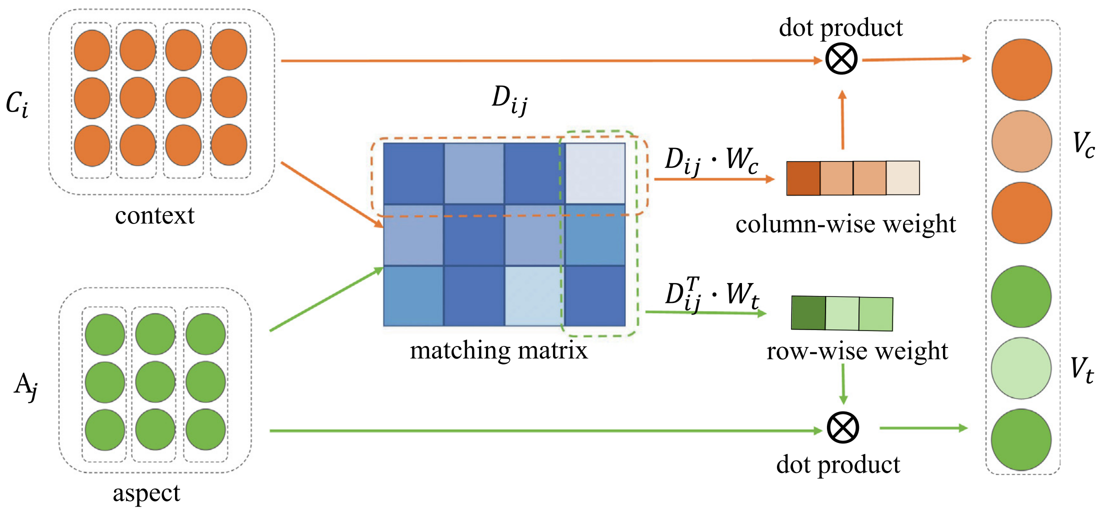


**Fig. 4.** An illustration of the Bi-attention mechanism.

$$
\begin{aligned}
f_c &= tanh(D_{ij} \cdot W_c) \\
a_c &= softmax\left(w_c \cdot f_c^T\right),
\end{aligned} \tag{10}
$$

where $W_c \in \mathbb{R}^{m \times 2d'}$ and $w_c \in \mathbb{R}^{2d'}$ are weight matrices. Finally, the aspect to context attention vector $V_c$ is obtained:

$$
V_c = a_c \cdot C_i. \tag{11}
$$

After computing $V_t$ and $V_c$, we concatenate two matching vectors as the final aspect-specific representation:

$$
V_i = [V_t || V_c]. \tag{12}
$$

### 3.6. Model training

We simultaneously train two tasks: the *aspect-level classification* task and the *relation-level classification* task, which share same parameters before the output layer.

#### 3.6.1. Aspect-level classification

For the final aspect-level classification model, we feed $V$ into a full connected network with softmax activation function:

$$
P_c = softmax(W_v \cdot V + b_v), \tag{13}
$$

where $W_v \in \mathbb{R}^{2d' \times c}$ and $b_v \in \mathbb{R}^c$ are the learned weight and bias. Here, $c$ is the number of aspect sentiment polarities.

The cross-entropy loss function is defined as:

$$
J(\theta) = -\sum_{t=1}^{T} y_t log \hat{y}_t + \beta \|\theta\|_2^2, \tag{14}
$$

where $T$ is the training size, $y_t$ is the true label of the $t$-th sample, $\hat{y}_t$ is the predicted label, and $\beta$ is the hyper-parameter to restrict the $L_2$ regularization.

#### 3.6.2. Relation-level classification

After obtaining two aspect-specific representations $V_i$ and $V_j$, the aspect relation can be modeled with a subtraction operator between them:

$$
R_{ij} = |V_i - V_j|. \tag{15}
$$

Then, we feed $R$ to a softmax classifier:

$$
P_r = softmax(W_r \cdot R + b_r), \tag{16}
$$

where $W_r \in \mathbb{R}^{2d' \times r}$ and $b_r \in \mathbb{R}^r$ are the learned weight and bias, and $r$ is the number of relation labels. The cross-entropy loss function is defined as:

$$
U = -\sum_{t=1}^{T} \frac{1}{n_t} \sum_{k=1}^{n_t} y_t^k log \hat{y}_t^k, \tag{17}
$$

where $n_t$ is the relation number of $t$-th sample. Noting a sentence may contain multiple aspect relations, we average relation-level loss to align with aspect-level loss in the same sentence.

#### 3.6.3. Aspect disagreement regularization

To acquire a larger aspect-specific vector space distance from overlapped representations, we adopt the disagreement regularization on aspects. Given a set of aspect representations $V = [V_1, V_2, \ldots, V_n]$ in one sentence, the average regularization term is calculated as:

$$
D = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{V_i \cdot V_j}{\|V_i\| \cdot \|V_j\|}, \tag{18}
$$

where $D$ is the average cosine distance between each aspect-specific representation and the smaller $D$ denotes that aspect representations are better distinguished.

#### 3.6.4. Training objective

The overall training objective of the proposed model mainly consists of three parts and the overall loss function is defined as follow:

$$
L = J(\theta) + \lambda U + D, \tag{19}
$$

where $J(\theta)$ is the aspect-level classification loss, $U$ is the relation-level classification loss, $D$ is the aspect disagreement regularization loss, and $\lambda$ is a hyper-parameter that decides the relative importance weight of $U$.

## 4. Experiments

We conduct extensive experiments with several settings of the proposed method in this section, showing the significant results and effectiveness of the relation-level classification task.

### 4.1. Datasets and experimental settings

**Datasets:** Several benchmark aspect-level datasets are chosen for experiment, including SemEval 2014 [39], SemEval 2015 [40], SemEval 2016 [41] and MAMS [42]. SemEval is an ongoing series of evaluations of computational semantic analysis systems. Following Zhang et al. [13], samples with conflict labels or without explicit aspects in sentences are removed. MAMS is a well-annotated dataset that each sentence contains at least two aspects. The detailed distributions of five datasets are presented in Table 3.

**Word Embeddings:** The pre-trained GloVe [36] and Bert [43] are used for word embedding initialization, in which out-of-vocabulary words in Golve embeddings are initialized by sampling from the uniform distribution $U\left(-\frac{1}{\sqrt{300}}, \frac{1}{\sqrt{300}}\right)$.

**Parameters:** Following standard methods, we tune our model using fivefold validation and grid-searching on the training set. The learning rate $lr$ is selected from $\{0.1, 0.001, 0.0005, 0.0001\}$, the batch size $b \in \{32, 64, 128\}$, and the hidden size $d \in \{64, 128, 256, 768\}$. Adam [44] is adopted to minimize the total loss $L$ given in Eq. (19). The regularization weight of parameters is $10^{-5}$, the dropout rate is 0.3, the epoch number is 20, and the final parameters are presented in Table 4.

**Evaluation Metric:** *Accuracy* and $Macro-F1$ are used to measure the performance of the models, which are defined as follows:

$$\begin{aligned} Accuracy &= \frac{T_{true}}{N}, \\ F1 &= \frac{2PR}{P+R}, \end{aligned} \tag{20}$$

where $T_{true}$ is the number of correctly predicted samples, $N$ is the total number of samples, $P$ is the positive predictive value, and $R$ is the recall value.

### 4.2. Baselines

We set three schemes for our method: 1) **RMN-A**: the model that considers all relations; 2) **RMN-P**: the model that only considers *similar* and *opposite* relations; and 3 **RMN-Bert**: our model with Bert embeddings. Besides, we choose the following methods for comparisons:

1. **Attention based methods**
   **TD-LSTM** [24] utilized contexts in both directions as feature representations for sentiment classification, which adopted two LSTM networks to model both the left context with target and the right context with target.
   **ATT-BILSTM** [26] learned aspect embeddings to compute attention weights, and conducted the attention mechanism between aspect and the context sentence.
   **AOA** [45] jointly modeled aspects and sentences and conducted an attention over attention module to generate the final representation.
   **ABCDM** [16] utilized two independent bidirectional LSTM and GRU layers to capture temporal information flow. In our implementation, we adopt the dependency graph to generate aspect embeddings.
   **TNet-LF** [46] adopted context preserving modules to handle aspect-level sentiment classification, and designed the target

**Table 3**
Dataset distributions on five datasets.

| Dataset | | Positive | Neutral | Negative | Single | Multiple | Total |
|---|---|---|---|---|---|---|---|
| Lap14 | Train | 994 | 464 | 870 | 911 | 1417 | 2328 |
| | Test | 341 | 169 | 128 | 259 | 379 | 638 |
| Rest14 | Train | 2164 | 637 | 807 | 1007 | 2601 | 3608 |
| | Test | 728 | 196 | 196 | 285 | 835 | 1120 |
| Rest15 | Train | 912 | 36 | 256 | 585 | 618 | 1204 |
| | Test | 326 | 34 | 182 | 301 | 241 | 542 |
| Rest16 | Train | 1240 | 69 | 439 | 879 | 869 | 1748 |
| | Test | 469 | 30 | 117 | 296 | 320 | 616 |
| MAMS | Train | 3380 | 5042 | 2764 | 0 | 11186 | 11186 |
| | Test | 400 | 263 | 393 | 0 | 1336 | 1336 |

**Table 4**
Detailed parameter settings in the experiment.

| Description | Symbol | Value |
|---|---|---|
| Class number | $c$ | 4 |
| Batch size | $b$ | 32 |
| Epoch number | $e$ | 20 |
| Dropout rate | $p$ | 0.3 |
| Hidden size | $d'$ | 300 |
| Learning rate | $lr$ | 0.001 |
| $L_2$ penalty | $\beta$ | 0.0001 |
| Pre-defined distance | $s$ | 15 |
| Maximum sentence length | $n$ | 80 |
| Relation-level task weight | $\lambda$ | (0, 1) |

specific transformation component to better integrate target information into the word representation.
**IARM** [14] incorporated the neighboring aspects related information with memory networks, where the attention mechanism was applied to all neighboring aspects to obtain aspect-specific embeddings.

2. **Multi-task learning based methods:**
**PRET + MULT** [32] incorporated knowledge from document-level corpus for training aspect-level sentiment classifiers, and validated that the aspect-level sentiment classification can be improved with the knowledge gained from document-level sentiment classification.
**TransCap** [33] developed a transfer learning framework to transfer knowledge from the document-level task to the aspect-level task, where an aspect routing approach was designed to generate sentence-level semantic features from both document and aspect.
3. **Graph based methods:**
**SDGCN-G** [15] modeled sentiment dependencies among aspects with multi-GCNs, and the final aspect-specific representation was acquired by the message passing mechanism between aspects in GCNs.
**ASGCN-DG** [13] exploited syntactical dependency structures within a sentence through multi-GCNs to generate aspect-oriented features, and imposed a masking layer on its top to extract the aspect-specific representation.
**R-GAT** [29] combined relation heads and attention heads in the graph attention network, where a aspect-oriented dependency tree structure was constructed by reshaping and pruning an ordinary dependency parse tree.

### *4.3. Results*

Table 5 provides the overall comparison results on all datasets, several results are highlighted: RMN-P outperforms other baselines with the best ACC and M-F1 performance on Rest14, Rest15 and MAMS datasets, and achieves about 0.09% to 0.8% on accuracy and about 0.04% to 1.19% on F1 score. Besides, graph based methods generally perform better than attention based methods, a possible explanation for this is that semantic dependencies are considered to generate aspect-specific representations. Compared to RGAT and ASGCN-DG, RMN-P performs worse on Lap14 dataset. The reason may be that samples on Lap14 dataset are not so sensitive to syntactic information since Lap14 contains more digits. Additionally, RMN-P performs about 0.46% to 2.49% better than RMN-A on all datasets since a *neutral* aspect may generate noise because of its uncertain polarity. At last, RMN-P, RMN-A and PRET + MULT perform well on Rest15 dataset though training samples are relatively small, which may benefit from other related tasks with multi-task learning in some cases. We further conduct several experiments to validate the phenomenon in Section 4.7.

Based on the superior performance of RMN, we can conclude the advantages of our model: 1) RMN captures semantic relations among sentence with GCNs encoder, and can generate more accurate aspect embeddings; 2) RMN constructs aspect relations in the same sentence and utilizes the extracted relations to facilitate sentiment analysis; and 3) Despite the lack of training samples, RMN can handle relatively small datasets with the assistance of the relation extraction task.

### *4.4. Ablation study*

To explore effects of different modules in the proposed model, we experiment RMN with five settings: 1) replacing Bi-LSTM with one layer full connected network; 2) removing GCN layers; 3) removing position encoding; 4) removing aspect disagreement regularization; and 5) removing relation-level classification. The results are presented in Table 6, showing that:

1. Bi-LSTM module has the most significant influence with about 7% reduction on the overall performance since it can extract bidirectional contextual information.
2. Without GCN layers, the performance decreases about 2.30% to 4.20% on all datasets, which validates the importance of considering syntactic structure.

**Table 5**
Performance of all baselines. The best results of each dataset are in bold. The results with ♯ are retrieved from [13]. The results with † are retrieved from [32], in which results on Rest15 and Rest16 are re-generated because of different sample distributions. Meanwhile, the results with ‡ are reproduced from the given code and parameters from [15,29].

| Category | Model | Lap14 | | Rest14 | | Rest15 | | Rest16 | | MAMS | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | | ACC | M-F1 | ACC | M-F1 | ACC | M-F1 | ACC | M-F1 | ACC | M-F1 |
| Att | TD-LSTM | 64.08 | 62.67 | 73.66 | 60.23 | 72.27 | 53.75 | 84.54 | 61.72 | 68.02 | 64.83 |
| | ATT-BiLSTM | 70.39 | 64.83 | 78.21 | 68.34 | 77.15 | 57.66 | 86.35 | 64.01 | 73.25 | 71.69 |
| | AOA♯ | 72.62 | 67.52 | 79.97 | 70.42 | 78.17 | 57.02 | 87.50 | 66.21 | – | – |
| | ABCDM | 72.84 | 68.33 | 79.47 | 70.32 | 78.29 | 57.88 | 87.98 | 68.54 | 74.33 | 72.15 |
| | IARM | 73.80 | – | 80.00 | – | – | – | – | – | – | – |
| | TNet-LF♯ | 74.61 | 70.14 | 80.42 | 71.03 | 78.47 | 59.47 | **89.07** | 70.43 | – | – |
| Multi | PRET + MULT† | 71.15 | 67.46 | 79.11 | 69.73 | 79.04 | 63.43 | 87.95 | 69.78 | 71.61 | 68.23 |
| | TransCap | 73.87 | 70.10 | 79.55 | 71.41 | – | – | – | – | – | – |
| Graph | SDGCN-G‡ | 74.20 | 69.69 | 80.98 | 71.99 | 77.68 | 60.25 | 88.47 | 67.91 | 77.10 | 75.99 |
| | ASGCN-DG♯ | 75.55 | 71.05 | 80.77 | 72.02 | 79.89 | 61.89 | 88.99 | 67.48 | 76.50 | 75.10 |
| | RGAT‡ | **75.68** | **71.19** | 81.07 | 72.12 | 78.59 | 60.95 | 87.17 | 66.36 | 78.97 | 78.01 |
| Ours | RMN-A | 72.95 | 68.25 | 80.43 | 71.98 | 80.15 | 63.14 | 88.20 | 69.05 | 78.24 | 77.95 |
| | RMN-P | 74.50 | 69.79 | **81.16** | **73.17** | **80.69** | **64.41** | 88.75 | **71.54** | **79.26** | **78.41** |
| | RMN-Bert | 77.95 | 70.83 | 84.56 | 79.05 | 82.94 | 66.95 | 89.38 | 71.88 | 79.97 | 78.79 |

**Table 6**
Comparisons of all modules in RMN.

| Model | Lap14 | | Rest14 | | Rest15 | | Rest16 | | MAMS | |
|---|---|---|---|---|---|---|---|---|---|---|
| | ACC | M-F1 | ACC | M-F1 | ACC | M-F1 | ACC | M-F1 | ACC | M-F1 |
| RMN-P | **74.50** | **69.79** | **81.16** | **73.17** | **80.69** | **64.41** | **88.75** | **71.54** | **79.26** | **78.41** |
| -w/o Bi-LSTM | 67.45 | 61.96 | 75.90 | 65.33 | 74.65 | 52.17 | 84.46 | 61.55 | 72.26 | 70.15 |
| -w/o GCN | 71.66 | 66.17 | 79.41 | 70.02 | 77.19 | 60.72 | 86.45 | 65.57 | 75.53 | 74.21 |
| -w/o Position | 71.71 | 66.32 | 79.55 | 71.06 | 79.10 | 63.50 | 87.66 | 67.54 | 77.26 | 75.13 |
| -w/o Regularization | 73.10 | 68.32 | 81.07 | 72.90 | 80.27 | 63.64 | 88.15 | 69.88 | 78.55 | 77.62 |
| -w/o Relation | 73.02 | 68.50 | 80.22 | 71.89 | 79.68 | 62.61 | 87.32 | 64.62 | 76.95 | 76.47 |

3. Position encoding works when generating context representation since context words near the aspect are more important than contexts away from the aspect.
4. Aspect regularization disagreement proves to be useful for its ability to distinguish their representations. Meanwhile, as an auxiliary task, relation-level classification really works since it can extract more aspect information from the original datasets.

### 4.5. Effects of relation-level task

To validate the effectiveness of relation-level task, we incorporate relation-level task with several basic models with ATT-LSTM and GCNs. For the implementation of GCNs model, the hidden sentence representations of Bi-LSTM are fed into two layer GCNs with average pooling operation. As can be seen from Table 7, models with relation-level tasks increase about 0.69%-1.71% on ACC and 1.05%-3.05% on M-F1. Besides, there are significant increments of both ACC and M-F1 on MAMS compared to other four datasets. The reason is that MAMS contains at least two different aspects in each sentence, thus is more suitable for our model.

### 4.6. Effects of model depth and aspect relation weight

We conduct several experiments on Rest14 datasets with different GCN layers to acquire the impact of the depth of GCN layers (ranging from 1 to 6). Fig. 5 (a) reveals that our model achieves the best ACC and M-F1 when the number of GCN layers is 2. Besides, one layer GCN model achieves the secondary performance, which implies that aspect-related sentiment words are generally 2-hops away from target words. Additionally, the results present that the model is trainable when the depth of GCNs within 3 layers. However, when the depth of GCNs is bigger than 4, our model presents a downward trend with the increment of GCN layers. One possible reason is that the model becomes more complex and needs more samples for training. see Table 8.

We also explore the impact of different relation weights ($\lambda$) on Rest14 dataset in this section, where $\lambda$ increases from 0 to 1. The results are presented in Fig. 5 (b), which shows that the model performs best when $\lambda = 0.4$. However, there is a fluctuation when $\lambda$ is in (0.1, 0.3). Two possible reasons may cause the phenomenon: 1) the relation-level loss ($U$) is averaged

**Table 7**
Results of combing relation-level task, and '+' denotes the model with relation-level task.

| Model | Dataset | Origin ⇒ Origin+ ACC | Origin ⇒ Origin+ M-F1 |
|---|---|---|---|
| ATT-BiLSTM | Lap14 | 70.39 ⇒ 71.12 (1.03%↑) | 64.83 ⇒ 65.72(1.37%↑) |
| | Res14 | 78.21 ⇒ 79.46(1.60%↑) | 68.34 ⇒ 70.30(2.87%↑) |
| | Res15 | 77.15 ⇒ 77.73(0.75%↑) | 57.66 ⇒ 59.57(3.31%↑) |
| | Res16 | 86.35 ⇒ 87.83(1.71%↑) | 64.01 ⇒ 65.40(2.17%↑) |
| | MAMS | 73.25 ⇒ 74.73(2.02%↑) | 71.69 ⇒ 73.47 (2.48%↑) |
| GCNs | Lap14 | 70.56 ⇒ 71.05(0.69%↑) | 65.04 ⇒ 65.96(1.41%↑) |
| | Res14 | 78.92 ⇒ 80.26(1.70%↑) | 70.53 ⇒ 71.61(1.53%↑) |
| | Res15 | 77.19 ⇒ 77.81(0.80%↑) | 57.95 ⇒ 60.01(3.55%↑) |
| | Res16 | 85.69 ⇒ 86.51(0.96%↑) | 64.43 ⇒ 65.99(2.42%↑) |
| | MAMS | 74.47 ⇒ 75.61(1.53%↑) | 72.72 ⇒ 74.45(2.38%↑) |

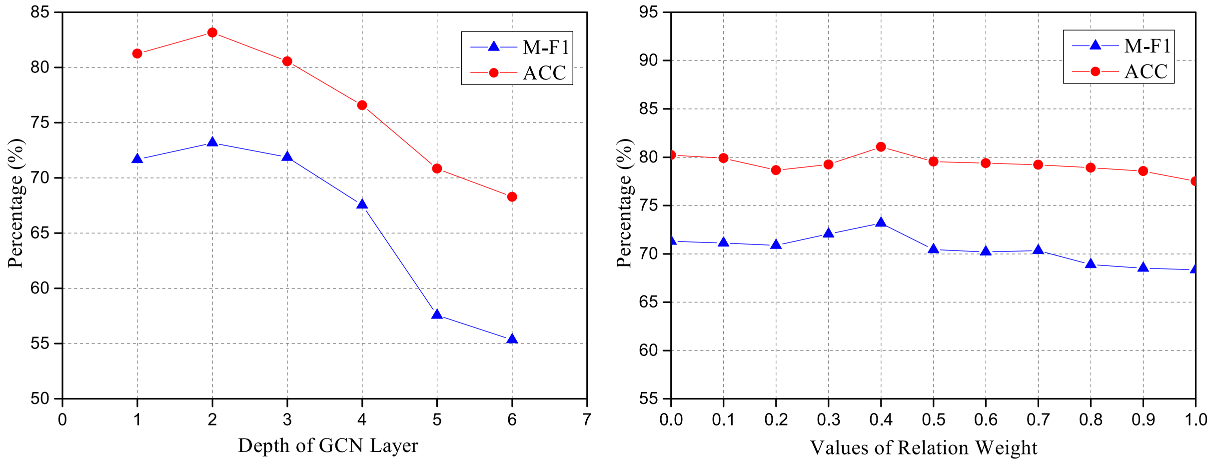


**Fig. 5.** Effects of different settings in RMN-P (a) Results of different depth of GCN layers. (b) Results of different $\lambda$.

when a sample contains multi-relations. 2) the total loss ($L$) is influenced by $U$: aspect-level loss and relation-level loss can not maintain consistency when $\lambda$ is too small. Meanwhile, the performance decreases when $\lambda$ is bigger than 0.6 because the total loss is penalized with bigger weight of relation-level loss.

### *4.7. Effects of data size and multi-aspects*

As mentioned in Section 4.3, we further conduct additional experiments to verify whether RMN can well handle relatively small dataset. We combine Rest14, Rest15 and Rest16 together and generate about 10% to 30% scales of the integrated dataset as subsets. We compare RMN with PRET + MULT because they both apply multi-task learning framework for aspect-level sentiment classification. As Fig. 6 presents, RMN is obviously superior to PRET + MULT on each dataset. The gap between two model becomes smaller with the increment of data size. Though PRET + MULT trains extra document-level corpus for additional knowledge from a similar domain, the lack of training samples makes the prediction difficult in practice. On the contrast, RMN performs better with the ability to extract more information from the original dataset, which expands training samples in some cases.

Meanwhile, we further present the performance on different scales of MAMS dataset. We choose SDGCN-G, ASGCN-DG and RGAT for comparisons because they all adopt GCN based methods. As can be seen in Fig. 7, four methods all show an increasing tendency with more training samples, and the gap between RMN and RGAT becomes smaller. Since SDGCN-G and RGAT both consider the dependency relations between aspects, they perform better than ASGCN-DG which only considers word dependencies among sentences.

### *4.8. Visualization*

To better explore the condition under which relational-level tasks work, we analyze the results that are correctly classified by RMN-P while are misclassified without relation-level task. We adopt the visualization approach presented by [47],

**Table 8**
Distribution of Rest dataset.

| Dataset | | Positive | Neutral | Negative |
|---|---|---|---|---|
| Rest-10% | Train | 362 | 61 | 117 |
| | Test | 159 | 25 | 56 |
| Rest-15% | Train | 537 | 96 | 177 |
| | Test | 234 | 38 | 88 |
| Rest-20% | Train | 698 | 135 | 247 |
| | Test | 316 | 52 | 112 |
| Rest-25% | Train | 884 | 170 | 296 |
| | Test | 383 | 59 | 158 |
| Rest-30% | Train | 1060 | 200 | 360 |
| | Test | 463 | 74 | 183 |

where the attention weights of “The *falafel* was rather over cooked and dried but the *chicken* was fine.” are visualized in Fig. 8.

For the aspect “*falafel*”, two models can classify their polarities correctly. Taking aspect relations into consideration, RMN-P assigns higher weights to the words “*over cooked*” and “*dried*” in sentence 3. Meanwhile, the attention weight of “*chicken*” decreases from 0.093 to 0.037. One possible reason is that the true relation of “*falafel*” and “*chicken*” is *opposite*. For the aspect “*chicken*”, as shown in sentence 2 and 4, the model without relation-level classification wrongly classifies its sentiment. RMN-P gains more attention on the context of “*chicken*” but less on “*falafel*”, which shows that the semantically related word can be better captured with the help of aspect relations.

We also adopt T-SNE [48] toolkit to visualize about 1000 aspect-specific representations on MAMS dataset. In Fig. 9, the representations are generally divided into three categories. With the help of “*similar*” relations, vectors with same semantic are closer and clusters are more dense. Besides, the boundary between two categories is more obvious with “*opposite*” relations in Fig. 9 (b) than the model without relation-level task, as illustrated in Fig. 9 (a).

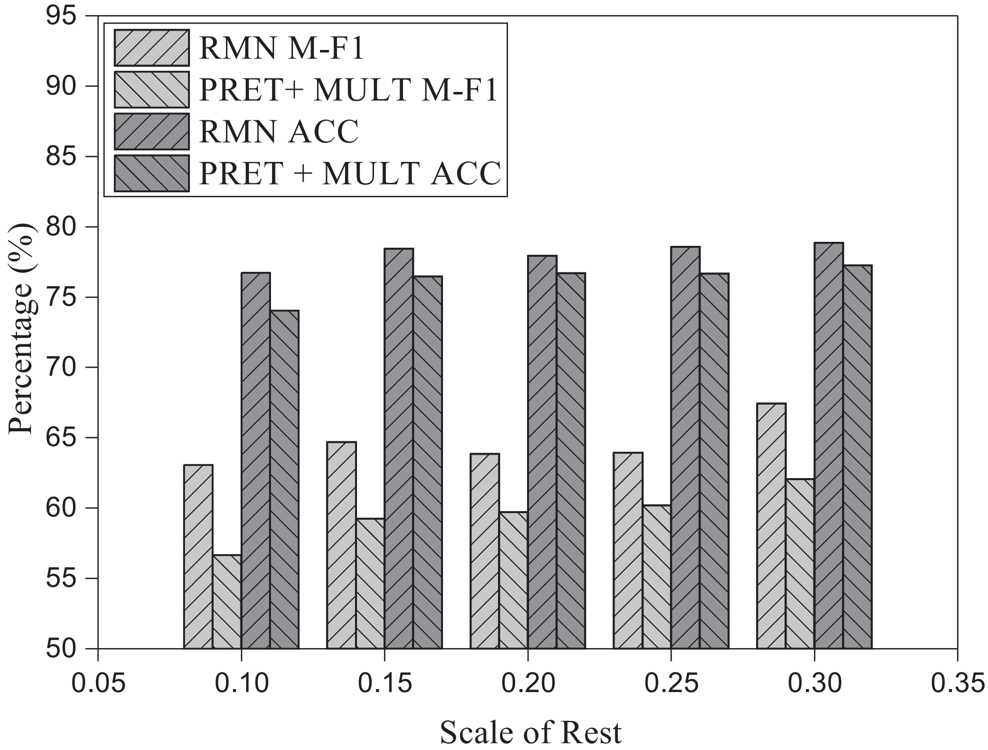


**Fig. 6.** Performance on different scale of Rest dataset.

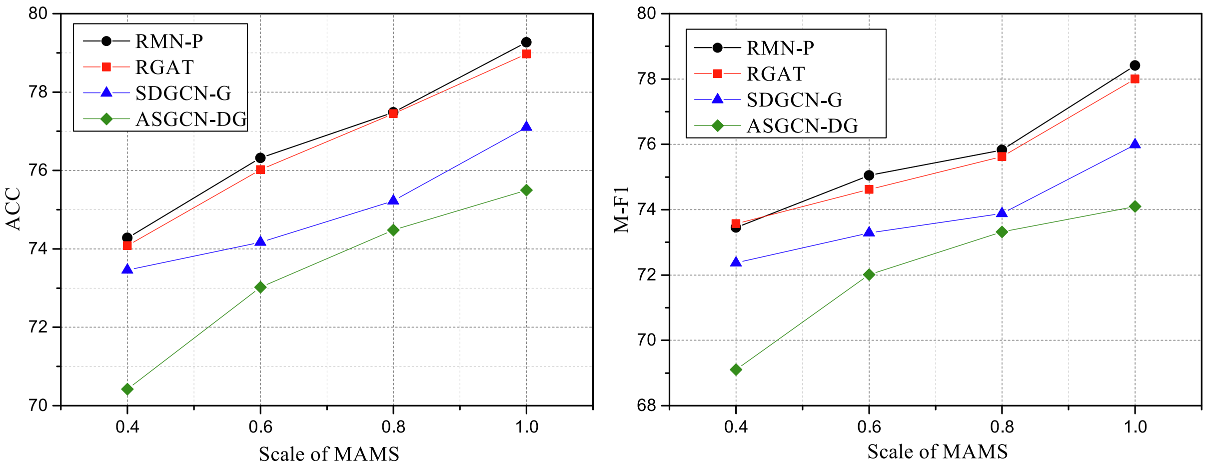


**Fig. 7.** Performance on multi-aspects. (a) Results of ACC on MAMS dataset. (b) Results of M-F1 on MAMS dataset.

| Model | | Sentence | Aspect | Label | Prediction |
|---|---|---|---|---|---|
| w/o Rel. | 1. | The falafel was rather over cooked and dried but the chicken was fine . | falafel | negative | negative |
| | 2. | The falafel was rather over cooked and dried but the chicken was fine . | chicken | positive | neutral |
| RMN-P | 3. | The falafel was rather over cooked and dried but the chicken was fine . | falafel | negative | negative |
| | 4. | The falafel was rather over cooked and dried but the chicken was fine . | chicken | positive | positive |

**Fig. 8.** Visualization of attention weight.

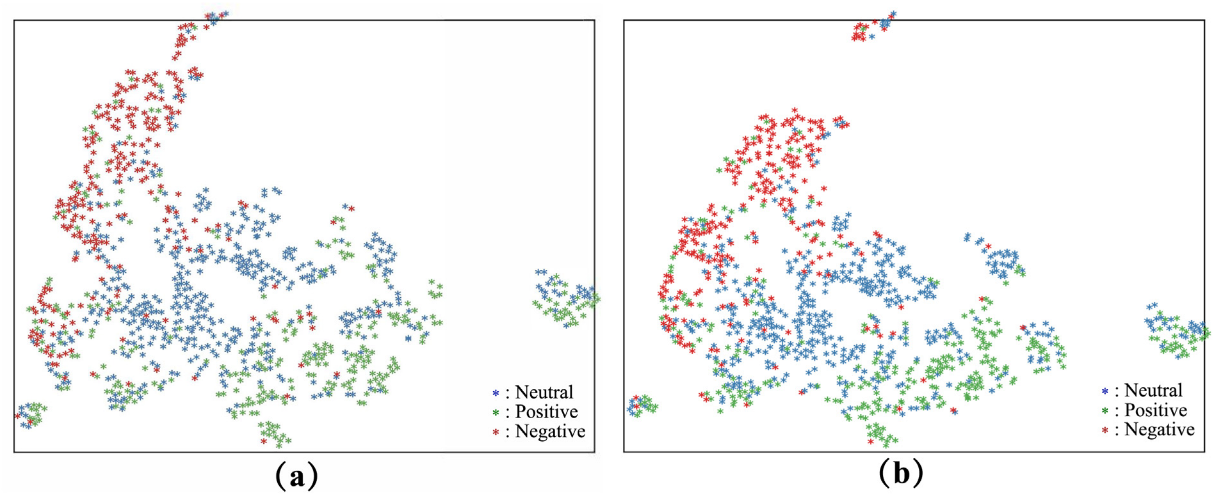


**Fig. 9.** T-SNE visualization on MAMS. (a) RMN-P without relation-level task. (b) RMN-P.

## 5. Conclusion

In this paper, we mainly explore aspect relations for aspect-level sentiment analysis. Our work is the first attempt to construct explicit aspect relations as an auxiliary task. We generate aspect representations with the dependency graph based

GCNs, and utilize the bidirectional attention mechanism to capture semantic relevance between the aspect and the context. No extra corpus is needed in our framework, and RMN performs better on relatively small datasets with the assistance of information extraction by the relational-level task. We also validate the effectiveness of the proposed approach and analyze the advantages brought by the relation-level classification task with extensive experiments.

This study may be further improved in the following ways: 1) since the relation-level task weight $\lambda$ is pre-define in this paper, we could apply the dynamic weight to balance the aspect-level loss and the relation-level loss. It is believe that the dynamic weight could lead to better performance. 2) For the position encoding module, this paper adopts the absolute distance as the basic metric to design position weights. Intuitively, the semantic distance of the corresponding dependency graph will be more accurate for context representations, and we will consider it as our new distance metric.

## CRediT authorship contribution statement

**Jiandian Zeng:** Methodology, Software, Writing-original-draft. **Tianyi Liu:** Methodology, Software, Validation. **Weijia Jia:** Validation, Writing – review & editing. **Jiantao Zhou:** Validation, Writing – review & editing.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

We greatly appreciate anonymous reviewers and the associate editor for their valuable and insightful comments that greatly helped to improve the quality of this article. This work was supported by Macau Science and Technology Development Fund under SKL-IOTSC-2021-2023, 0072/2020/AMJ, 077/2018/A2, and 0060/2019/A1, by Research Committee at University of Macau under MYRG2018-00029-FST and MYRG2019-00023-FST, and by Natural Science Foundation of China under 61971476.

## References

[1] Q. Geng, S. Deng, D. Jia, J. Jin, Cross-domain ontology construction and alignment from online customer product reviews, Inf. Sci. 531 (2020) 47–67.
[2] B. Liu, L. Zhang, A survey of opinion mining and sentiment analysis, in: Mining text data, Springer, 2012, pp. 415–463..
[3] S.K. Khare, V. Bajaj, Time-frequency representation and convolutional neural network-based emotion recognition, IEEE Trans. Neural Networks Learn. Syst. (2020).
[4] Z.-Y. Dou, Capturing user and product information for document level sentiment analysis with deep memory network, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 521–526.
[5] T. Thongtan, T. Phienthrakul, Sentiment classification using document embeddings trained with cosine similarity, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, 2019, pp. 407–414..
[6] M.E. Basiri, M. Abdar, M.A. Cifci, S. Nemati, U.R. Acharya, A novel method for sentiment classification of drug reviews using fusion of deep and machine learning techniques, Knowl.-Based Syst. 198 (2020) 105949.
[7] J. Wang, L.-C. Yu, K.R. Lai, X. Zhang, Investigating dynamic routing in tree-structured lstm for sentiment analysis, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 3423–3428.
[8] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based lstm for aspect-level sentiment classification, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, 2016, pp. 606–615.
[9] A. Esuli, F. Sebastiani, Sentiwordnet: A publicly available lexical resource for opinion mining., in: Proceedings of the 2006 Conference on the European Language Resources Association, vol. 6, 2006, pp. 417–422..
[10] Y. Liu, J.-W. Bi, Z.-P. Fan, A method for multi-class sentiment classification based on an improved one-vs-one (ovo) strategy and the support vector machine (svm) algorithm, Inf. Sci. 394 (2017) 38–52.
[11] F. Fan, Y. Feng, D. Zhao, Multi-grained attention network for aspect-level sentiment classification, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3433–3442.
[12] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the 26th International Joint Conference on Artificial Intelligence, 2017, pp. 4068–4074.
[13] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019, pp. 4568–4578.
[14] N. Majumder, S. Poria, A. Gelbukh, M.S. Akhtar, E. Cambria, A. Ekbal, Iarm: Inter-aspect relation modeling with memory networks in aspect-based sentiment analysis, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3402–3411.
[15] P. Zhao, L. Hou, O. Wu, Modeling sentiment dependencies with graph convolutional networks for aspect-level sentiment classification, Knowl.-Based Syst. 105443 (2020).
[16] M.E. Basiri, S. Nemati, M. Abdar, E. Cambria, U.R. Acharya, Abcdm: An attention-based bidirectional cnn-rnn deep model for sentiment analysis, Future Gener. Comput. Syst. 115 (2021) 279–294.
[17] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.
[18] J. Tang, Z. Lu, J. Su, Y. Ge, L. Song, L. Sun, J. Luo, Progressive self-supervised attention learning for aspect-level sentiment analysis, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 557–566.
[19] M.H. Phan, P.O. Ogunbona, Modelling context and syntactical features for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3211–3220.
[20] Q. Liu, Y. Zhang, J. Liu, Learning domain representation for multi-domain sentiment classification, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 541–550..

[21] Y. Zhang, Z. Zhang, D. Miao, J. Wang, Three-way enhanced convolutional neural networks for sentence-level sentiment classification, Inf. Sci. 477 (2019) 55–64.
[22] B. Zhang, M. Yang, X. Li, Y. Ye, X. Xu, K. Dai, Enhancing cross-target stance detection with transferable semantic-emotion knowledge, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3188–3197.
[23] M. Zhang, V. Palade, Y. Wang, Z. Ji, Attention-based word embeddings using artificial bee colony algorithm for aspect-level sentiment classification, Inf. Sci. 545 (2021) 713–738.
[24] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.
[25] M. Yang, W. Tu, J. Wang, F. Xu, X. Chen, Attention based lstm for target dependent sentiment classification, in: Proceedings of the 31th Conference on Artificial Intelligence, 2017, pp. 5013–5014.
[26] J. Liu, Y. Zhang, Attention modeling for targeted sentiment, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics, 2017, pp. 572–577.
[27] R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, Effective attention modeling for aspect-level sentiment classification, in: Proceedings of the 27th International Conference on Computational Linguistics, 2018, pp. 1121–1131.
[28] D. Hazarika, S. Poria, P. Vij, G. Krishnamurthy, E. Cambria, R. Zimmermann, Modeling inter-aspect dependencies for aspect-based sentiment analysis, in: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2018, pp. 266–270.
[29] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3229–3238.
[30] M. Zhang, T. Qian, Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, 2020, pp. 3540–3549.
[31] L. Yuan, J. Wang, L.-C. Yu, X. Zhang, Graph attention network with memory fusion for aspect-level sentiment analysis, in: Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing, 2020, pp. 27–36.
[32] R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, Exploiting document knowledge for aspect-level sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 579–585.
[33] Z. Chen, T. Qian, Transfer capsule network for aspect level sentiment classification, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 547–556.
[34] Y. Liu, S. Li, X. Zhang, Z. Sui, Implicit discourse relation classification via multi-task neural networks, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 74–79..
[35] R. He, W.S. Lee, H.T. Ng, D. Dahlmeier, An interactive multi-task learning network for end-to-end aspect-based sentiment analysis, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 504–515.
[36] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, 2014, pp. 1532–1543.
[37] A. Graves, A.-R. Mohamed, G. Hinton, Speech recognition with deep recurrent neural networks, in: 2013 IEEE international Conference on Acoustics, Speech and Signal Processing, 2013, pp. 6645–6649.
[38] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452–461.
[39] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), 2014, pp. 27–35.
[40] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: Aspect based sentiment analysis, in: Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015), 2015, pp. 486–495.
[41] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. Al-Smadi, M. Al-Ayyoub, Y. Zhao, B. Qin, O. De Clercq, et al, Semeval-2016 task 5: Aspect based sentiment analysis, in: 10th International Workshop on Semantic Evaluation (SemEval 2016), 2016, pp. 19–30.
[42] Q. Jiang, L. Chen, R. Xu, X. Ao, M. Yang, A challenge dataset and effective models for aspect-based sentiment analysis, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, 2019, pp. 6281–6286.
[43] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding (2019) 4171–4186..
[44] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of the 3rd International Conference on Learning Representations, 2015, pp. 1–15.
[45] B. Huang, Y. Ou, K.M. Carley, Aspect level sentiment classification with attention-over-attention neural networks, in: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation Springer, 2018, pp. 197–206.
[46] X. Li, L. Bing, W. Lam, B. Shi, Transformation networks for target-oriented sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 946–956.
[47] J. Yang, Y. Zhang, Ncrf++: An open-source neural sequence labeling toolkit, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 74–79.
[48] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, J. Mach. Learn. Res. (2008) 2579–2605..