Neurocomputing 457 (2021) 214–224

Contents lists available at ScienceDirect

Neurocomputing

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

Fine-grained Question-Answer sentiment classification with hierarchical graph attention network

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

[a] State Key Lab of IoT 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, China
[c] BNU-UIC Joint AI Research Institute, Beijing Normal University, China

ARTICLE INFO

Article history:
Received 9 October 2020
Revised 3 May 2021
Accepted 12 June 2021
Available online 17 June 2021
Communicated by Zidong Wang

Keywords:
Sentiment classification
Graph attention network
Question Answer

ABSTRACT

User-oriented Question-Answer (QA) text pair plays an increasingly important role in online e-commerce platforms, and expresses sentiment information with complicated semantic relations, causing great challenges for accurate sentiment analysis. To address this problem, we propose a novel hierarchical graph attention network (HGAT) to explore abundant relations. Firstly, we utilize the dependency parser to model relations of sentiment words with consideration of syntactic structures within sub-sentences. Then, to better extract hidden features of these sentiment words, we feed the dependency graph into an improved word-level graph attention network (GAT) that incorporates the learned attention weight with the prior graph edge weight. Besides, the sigmoid self-attention mechanism is applied to aggregate salient word representations. Finally, we establish a graph of all sub-sentences with a strong connection and capture inter-relations and intra-relations through the sentence-level GAT. Extensive experiments show that HGAT can achieve significant improvements in QA-style sentiment classification compared with several baselines.

1. Introduction

Nowadays, there are a large scale of reviews provided by users on websites, which are valuable for e-commerce platforms (e.g., *Taobao* and *Amazon*) to analyze them for product recommendation. An effective way is to conduct sentiment analysis on users' review texts directly, aiming to infer sentiment polarity of each sentence [1]. It has been well studied from document-level, sentence-level and aspect-level.

Recently, a new customer QA-style review form has been increasingly popular on e-commerce platforms, and it is conducted to exchange ideas towards diverse characteristics of products in an interactive way. Different from traditional non-interactive reviews, QA-style reviews are more convincing and informative [2]. Meanwhile, due to irregular expressions in QA-style reviews, multi-grained relations among sub-sentences are more complicated. As shown in Fig. 1, there are 8 potential relations between all sub-questions and sub-answers, where dotted arrows (① to ④) and solid lines (⑤ to ⑧) denote inter-relations and intra-relations respectively. Besides, all correlated sub-sentences marked in the same color indicate the identical sentiment polarity to one aspect of the product. It can be seen that relation ⑦ presents positive sentiment while ⑧ shows conflict. This example illustrates that a sub-sentence in a QA pair containing abundant sentiment information may lead to different prediction results. Hence, multiple relations should be considered when determining the overall sentiment of the review. To tackle the above problems, two major challenges should be addressed: 1) capture inter-relations between the question and the answer; and 2) capture intra-relations among sentences.

For the first challenge, if previous algorithms [3,4] that handled one QA text pair as a successive sequence are applied directly, inter-relations between question and answer may be lost. For instance, in Fig. 1(a), sub-answers a_4 and a_5 correspond to sub-questions q_4 and q_2 respectively. Nevertheless, if we treat them as a continuous sequence, the polarity of the entire QA pair is more likely to be positive rather than conflict. To capture inter-relations between the QA, a hierarchical matching network model was proposed in [2]. The authors segmented a QA text pair into several sub-questions and sub-answers, then the Answer-to-Question Attention and the Question-to-Answer Attention mechanism were adopted to explore sentiment information. However, they only considered inter-relations between QA pair while intra-relations were neglected.

* 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.neucom.2021.06.040
0925-2312/

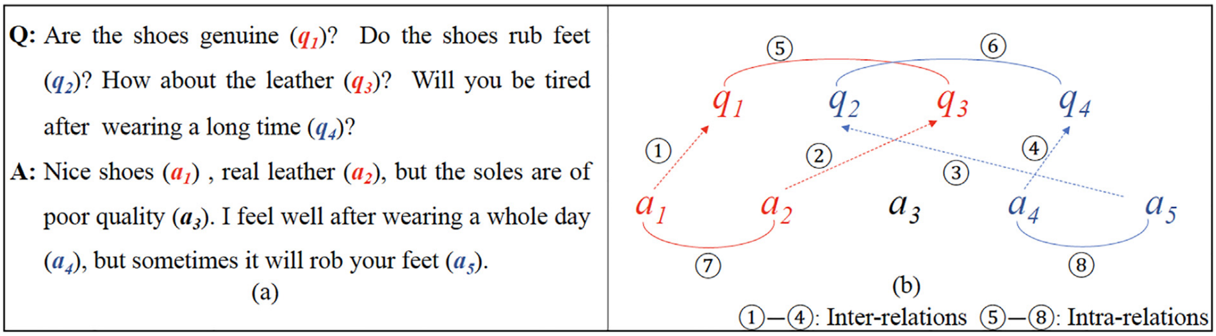


Fig. 1. (a) The original QA text pair with several sub-sentences. (b) All relations among sub-sentences.

For the second challenge, intra-relations among sentences that contain abundant information should be further considered. For instance, in Fig. 1(a), sub-questions q_1 and q_3 both consult the quality of shoes, and q_2 and q_4 query the comfortableness of shoes. Accordingly, a_1 and a_2 are related for they both answer the first question, and the correlated a_4 and a_5 both answer the comfortableness of shoes. Therefore, considering these intra-relations is crucial for accurate sentiment analysis. In addition, salient semantic words are distributed in different positions, and the word-level dependencies need to be captured for better prediction.

To address this issue, we propose a hierarchical graph attention network (HGAT) to capture abundant relations in QA-style reviews. In our work, we first acquire the dependency graph of each sub-sentence through a dependency parser, where all syntactically related words are connected. To extract hidden features between related words, a word-level GAT is designed to combine the learned attention weight with the prior graph edge weight. Then, to aggregate representations of salient words, the sigmoid self-attention mechanism is applied to manage information flow between the graph. Finally, we treat all sub-sentences in one QA pair as a strongly connected graph and acquire inter-relations and intra-relations via the sentence-level GAT. We conduct experiments with two QA-style benchmarks. Experimental results show that HGAT achieves significant improvements compared with several baselines. Our contributions are summarized as follows:

- We propose a novel hierarchical graph attention architecture to capture multi-level relationships in QA-style sentiment analysis.
- We explore the former edge knowledge in GAT that combines the learned weight with the prior graph edge weight, and incorporate the sigmoid self-attention mechanism to select salient information.
- HGAT achieves significant improvements for QA-style sentiment classification compared with several baselines on *Taobao* and SemEval2016 datasets.

The rest of this paper is organized as follows: Section 2 reviews related works. Section 3 describes the proposed HGAT in detail. 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 [5–7] has served an essential role in many NLP tasks, and can be divided into three levels: document-level [8–10], sentence-level [11–13] and aspect-level [14–16]. In the document-level sentiment classification, Dou et al. [8] proposed a deep memory network to predict the sentiment polarity of a whole document. Thongtan et al. [9] utilized cosine similarity as the similarity measure to train document embeddings for sentiment classification. Du et al. [17] combined the textual representation and the corresponding commonsense knowledge representation for stance classification. In the sentence-level sentiment classification, recent context-aware sub-symbolic approaches were proposed to obtain sentiment-specific word embeddings [18], such as attention modeling [19–21] and capsule networks [12,22]. In [11], the authors investigated domain representations of multitask learning for multi-domain sentiment analysis towards each sentence. To overcome the bias problem of sequential LSTM, Wang et al. [12] proposed a capsule tree-LSTM model to assign different weights according to their contributions.

Different from the document-level and the sentence-level sentiment classification that predict the whole polarity of the given text, the aspect-level sentiment analysis aims to identify the polarities of different aspects based on their contextual words and performs finer-grained analysis [23,24]. In the aspect-level sentiment analysis, Tang et al. [25] developed two LSTM models and took target information into account automatically by adding a target connection component. In [26], the authors proposed to automatically mine supervision information of the attention mechanism. Zhang et al. [24] applied graph convolutional networks (GCNs) to obtain long-away aspect-relevant information.

Since QA-style reviews are more complicated and irregular, none of the above method is well suited for this kind of review. Shen et al. [2] is the first work to propose QA-style level sentiment classification with the use of a hierarchical matching network model. More recently, Wang et al. [27] investigated aspect-level sentiment classification with pre-defined aspects on QA-style datasets. However, they failed to capture intra-relations among sub-sentences.

2.2. Graph Neural Networks

Graph neural networks have attracted growing attention with the ability of handing real-world graph-structured data [28–30]. Kipf et al. [31] proposed GCNs to encode the syntactic structure of sentences. Marcheggiani et al. [32] utilized GCNs for semantic role labeling tasks, and presented that GCNs can be used to incorporate syntactic information into neural models. Yao et al. [33] regarded documents and words as nodes, and applied GCNs for text classification. Schlichtkrull et al. [34] proposed relational GCNs to deal with the highly multi-relational data characteristic of realistic knowledge bases.

In [35], the authors proposed graph attention network (GAT) and introduced an attention-based architecture to perform node classification of graph-structured data. Compare to GCN, GAT only requires the structure of neighbor nodes rather than the entire graph structure. Huang et al. [36] utilized GAT for aspect-level sentiment classification, which extracted syntax among words with dependency relationships. Wang et al. [37] proposed to extend the original GAT with additional relational heads for sentiment prediction. Inspired by the above works, we propose an improved hierarchical GAT considering the prior edge weight to capture inter-relations and intra-relations on QA-style datasets.

3. Methodology

The overall architecture of HGAT is presented in Fig. 2, which mainly consists of the word encoding layer, the hierarchical GAT layer, and the output layer.

3.1. Problem definition and notation

Given an input QA text pair: question $Q = [q_1, q_2, \ldots, q_i, \ldots, q_M]$ and answer $A = [a_1, a_2, \ldots, a_j, \ldots, a_N]$, where q_i and a_j denote sub-question and sub-answer respectively. In detail, each question contains M sub-questions and each answer contains N sub-answers, where q_i contains m words: $q_i = [q^w_{i1}, q^w_{i2}, \ldots, q^w_{im}]$ and a_j contains n words: $a_j = \left[a^w_{j1}, a^w_{j2}, \ldots, a^w_{jn}\right]$. The task is to classify sentiment polarity (*positive, negative, neutral* or *conflict*) of a QA text pair.

3.2. Word encoding

The inputs of the sentiment extractor are word tokens, and these tokens are transformed to the distributed representations for the convenience of calculation by neural networks.

3.2.1. Word embedding

We employ the pre-trained models (Skip-Gram [38] or Bert [39]) to obtain the word embedding of each word $w_i \in \mathbb{R}^{d\times 1}$, where d represents the embedding dimension. The dimensions of each sub-question and sub-answer are $q_i \in \mathbb{R}^{m\times d}$ and $a_i \in \mathbb{R}^{n\times d}$ respectively. Furthermore, the input question $Q \in \mathbb{R}^{M\times m\times d}$ and the input answer $A \in \mathbb{R}^{N\times n\times d}$ in a QA pair are obtained by the word embedding layer.

3.2.2. Bi-LSTM encoding

To capture the future context as well as the past, we feed each sub-sentence into the Bidirectional Long-Short Term Memory network (Bi-LSTM) to extract contextual information from words, in which the forward hidden state $\overrightarrow{h} \in \mathbb{R}^{d'\times 1}$ and the backward hidden state $\overleftarrow{h} \in \mathbb{R}^{d'\times 1}$, and d' is the dimension of hidden state. Then, the encoded representation of each word is formed by concatenating two hidden states:

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

where || denotes the vertical concatenating operation.

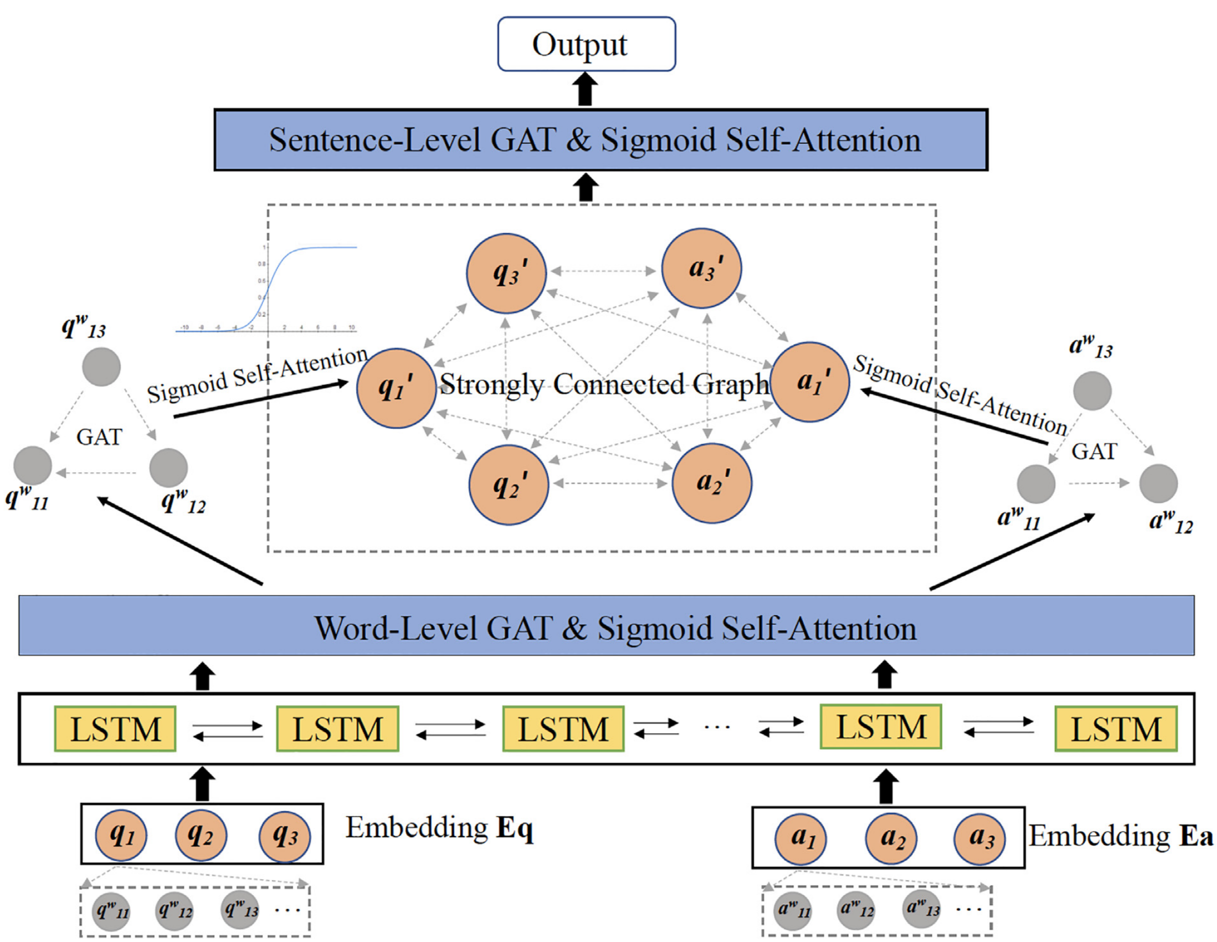


Fig. 2. The architecture of HGAT. In the word-level GAT, the dependency parser is utilized to generate dependency graph. In the sentence-level GAT, a strongly connected graph is formed with all sub-sentences.

3.3. Word-level graph attention network

In this subsection, we will specifically introduce the word-level graph attention network. Firstly, we utilize the dependency parser to acquire syntactic relations between words and transform each sub-sentence into the corresponding dependency graph. Then, we conduct an improved GAT and the sigmoid self-attention mechanism to obtain the representation of each sub-sentence.

3.3.1. Intra-QA Relation Parser

The Stanford Parser [40] toolkit is adopted to acquire the syntactic structure of sub-sentences in each QA. With the universal dependencies generated by the Stanford Parser Tree, intra-QA word-level relations can be obtained. For instance, as shown in Fig. 3(a), the sentence "*The food is decent though not worth the price*" has 8 dependency relations, where the syntactically related words are linked with the directed edges in a dependency graph. Meanwhile, Fig. 3(b) shows the corresponding adjacent matrix of the parsed sentence, where '1' denotes that two words are syntactically connected.

3.3.2. Graph attention network

GAT performs well on graph-structured data, and can aggregate the representations of neighborhood nodes [35]. In this work, we conduct an improved GAT to extract intra-QA information. Given n encoded nodes $h = [h_1, h_2, \ldots, h_i, \ldots, h_n]$, if the node h_j is the neighbor of the node h_i in the dependency graph, we concatenate h_i and h_j with a shared *weight matrix* W_s:

$$w_{i,j} = tanh[W_s h_i || W_s h_j], \tag{2}$$

where $W_s \in \mathbb{R}^{d^{hid} \times 2d'}$ and $w_{i,j} \in \mathbb{R}^{2d^{hid} \times 1}$. Then, the normalized attention weight $a_{i,j}$ can be calculated as:

$$a_{i,j} = \frac{exp(w_{i,j})}{\sum_{p \in Nei} exp(w_{i,p})}, \tag{3}$$

where *Nei* denotes the collection of neighbor nodes for the i-th node.

Fig. 4 shows the motivation of combining the prior edge information. For the sentence "*The food is decent though not worth the price*", the word "*decent*" has a relative larger weight since it determines the polarity of "*food*" in the original GAT. If we take the prior edge information into consideration, the weight of "*decent*" is more significant, showing that the combination of the learned attention weight $a_{i,j}$ with the former graph adjacency matrix $graph_{i,j}$ can better utilize the prior knowledge. Thus we upgrade $a_{i,j}$ with a weighted sum of the learned attention weight and the prior edge weight of the dependency graph:

$$\bar{a}_{i,j} = \lambda * graph_{i,j} + (1 - \lambda) * a_{i,j}, \tag{4}$$

where λ is a pre-defined parameter that decides the importance of learned weight and original adjacency weights. If λ is set to 0, weights of edges are completely learned by the model. If λ is set to 1, the attention mechanism is not required.

The output representation is calculated by:

$$h'_i = \sum_{j \in Nei} \bar{a}_{i,j} * W_s h_j, \tag{5}$$

where $h'_i \in \mathbb{R}^{d^{hid} \times 1}$. In our work, we also apply multi-head attention into graph attention to learn information from different semantic spaces:

$$h'_i = \Big\Vert_{k=1}^{K} \left(\sum_{j \in Nei} \bar{a}^k_{i,j} * W^k_s h_j \right), \tag{6}$$

where K is the number of heads. After the calculation of the improved word-level GAT, the dimension of each word is $\mathbb{R}^{d^{hid} \times 1}$. Meanwhile, we can obtain the sub-question $q_i \in \mathbb{R}^{m \times d^{hid}}$ and sub-answer $a_i \in \mathbb{R}^{n \times d^{hid}}$.

3.3.3. Sigmoid self-attention

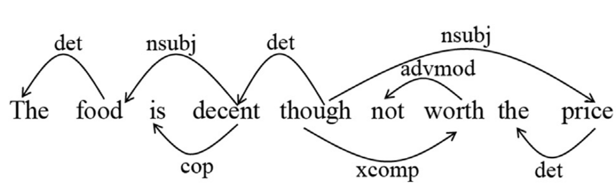
For each sub-question q_i and sub-answer a_i, to simplify the mathematical expression, we denote them as sub-sentence $s_i \in \mathbb{R}^{n \times d^{hid}}$, where n is the number of words. We propose a sigmoid self-attention mechanism to select salient information, and the importance weight α can be learned as follows:

$$w = sigmoid(W_u s_i^T \odot W_m s_i^T), \tag{7}$$

$$\alpha = softmax(w^T W_r), \tag{8}$$

where $W_u \in \mathbb{R}^{d^{hid} \times d^{hid}}, W_m \in \mathbb{R}^{d^{hid} \times d^{hid}}, W_r \in \mathbb{R}^{d^{hid} \times 1}$ are weight matrices, $w \in \mathbb{R}^{d^{hid} \times n}, \alpha \in \mathbb{R}^{n \times 1}$ and $\odot$ is the element-wise multiplication operation. The *sigmoid* function has the domain of all real numbers, with return value monotonically increasing from 0 to 1. It helps assign large weights near 1 to salient sentiment words while assigning small weights to irrelevant words near 0. Then, a *softmax* function is used to normalize the attention weight α, and the sub-sentence representation s'_i is calculated by:

$$s'_i = s_i^T \alpha, \tag{9}$$



(a)

	The	food	is	decent	though	not	worth	the	price
The	1	1	0	0	0	0	0	0	0
food	1	1	0	1	0	0	0	0	0
is	0	0	1	1	0	0	0	0	0
decent	0	1	1	1	1	0	0	0	0
though	0	0	0	1	1	0	1	0	1
not	0	0	0	0	0	1	1	0	0
worth	0	0	0	0	1	1	1	0	0
the	0	0	0	0	0	0	0	1	1
price	0	0	0	0	1	0	0	1	1

(b)

Fig. 3. An illustration of the dependency graph. (a) The universal dependencies are generated by the Stanford Parser Tree. (b) The adjacency matrix of the given dependencies.

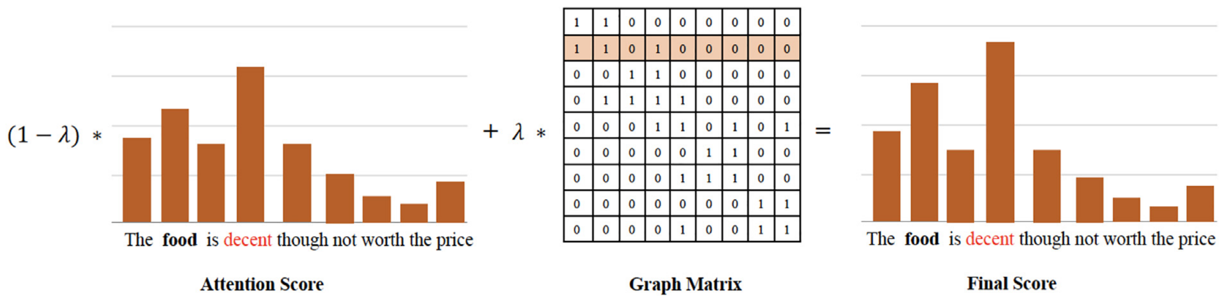


Fig. 4. An illustration of combining the prior edge weight.

where $s'_i \in \mathbb{R}^{d^{hid}\times 1}$ represents the output of the sigmoid self-attention layer.

With the calculation of the word-level GAT and the sigmoid self-attention (Eqs. (2)–(9)), we can obtain the representations of question $Q \in \mathbb{R}^{M\times d^{hid}}$ and answer $A \in \mathbb{R}^{N\times d^{hid}}$ in a QA pair.

3.4. Sentence-level graph attention network

As mentioned before, the outputs of the word-level GAT can be denoted as: question $Q = [q'_1, q'_2, \ldots, q'_i, \ldots, q'_M]$ and answer $A = [a'_1, a'_2, \ldots, a'_i, \ldots, a'_N]$, where q'_i and a'_i are the renewed sub-sentence representations. In the sentence-level GAT, each sub-sentence s'_i is considered as a node, and we treat all nodes as a strongly connected graph to capture inter-relations and intra-relations among sub-sentences. Since any two nodes are linked in a strongly connected graph, each value in the corresponding adjacent matrix is set to 1. The entire QA representation can be calculated as follows:

$$S = \sum_{i\in M+N} GAT(s'_i), \tag{10}$$

where $S \in \mathbb{R}^{(M+N)\times d^{hid}}$ is the collection of all new sub-sentence representations, and GAT is the same operation in the word-level GAT (Algorithm 1 lines 5–14). Then, we adopt sigmoid self-attention mechanism to obtain the QA-pair representation R for final sentiment classification:

$$w' = sigmoid\left(W_u S'^T \odot W_m S'^T\right), \tag{11}$$

$$\alpha' = softmax\left(w'^T W'_r\right), \tag{12}$$

$$R = S^T \alpha', \tag{13}$$

where $W'^T_u \in \mathbb{R}^{d^{hid}\times d^{hid}}, W'^T_m \in \mathbb{R}^{d^{hid}\times d^{hid}}, W'_r \in \mathbb{R}^{d^{hid}\times 1}$ are weight matrices, $w' \in \mathbb{R}^{d^{hid}\times(M+N)}, \alpha' \in \mathbb{R}^{(M+N)\times 1}$, and $R \in \mathbb{R}^{d^{hid}\times 1}$.

3.5. QA-Pair classification

For the final classification model, we feed R into a fully connected network with *softmax* activation function:

$$P = softmax(W_R R + b_R), \tag{14}$$

where $W_R \in \mathbb{R}^{d^{hid}\times c}$ is weight matrix, $b_R \in \mathbb{R}^{c\times 1}$ is the bias and c is the number of sentiment polarities. The detailed steps of the proposed HGAT algorithm are shown in Algorithm 1.

Algorithm 1. HGAT Algorithm

Input: QA text pair question Q, answer A, edge weight λ, head number K.
Output: Predicted sentiment polarity $\hat{y}$.
1: Split each Q and A into sub-sentence q_i and a_i;
2: Obtain the word encoding h_i by Bi-LSTM module;
3: Obtain the dependency graph by the Stanford Parser toolkit;
4: Calculate the new word encoding by *GAT* function;
5: **function** GAT(h_i)
6: **for** k in range(K) **do**
7: $w^k_{i,j} = tanh\left[W^k_s h_i || W^k_s h_j\right]$,
8: $a^k_{i,j} = softmax\left(w^k_{i,j}\right)$,
9: $\overline{a}^k_{i,j} = \lambda * graph_{i,j} + (1-\lambda) * a_{i,j}$,//combine the prior edge weight
10: $h'^k_i = \sum_{j\in Nei} \overline{a}^k_{i,j} * W^k_s h_j$,
11: **end for**
12: $h'_i \leftarrow$ **Concatenate** (h'^k_i),
13: **return** h'_i
14: **end function**
15: Obtain the sub-sentence representation s'_i by the sigmoid self-attention(Eqs. (7)–(9));
16: Calculate the new sub-sentence representation by the *GAT* function with input s'_i;
17: Obtain the final feature representation R by the sigmoid self-attention(Eqs. (11)–(13));
18: $\hat{y} = argmax(softmax(W_R R + b_R))$,
19: **return** $\hat{y}$.

The cross-entropy loss function is defined as:

$$J(\theta) = -\sum_{t=1}^{T} y_t * log\hat{y}_t + \beta\|\theta\|_2^2, \tag{15}$$

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 β is the hyper-parameter to restrict the L2 regularization.

4. Datasets and experimental settings

All experiments are conducted on a Linux server (Ubuntu 18.04.1) with a Interl(R) Xeon(R) Gold 5120 CPU, 8 Nvidia 2080TI

GPUs and 128G RAM. The detailed datasets and experimental settings are described as follows:

Datasets: We conduct experiments on *Taobao* and SemEval2016 datasets. The *Taobao* datasets are released by [2], which are collected from domains of *Beauty*, *Shoe*, and *Electronic*. In each domain, there are 10000 QA text pairs with individual labels (*positive*, *negative*, *neutral* or *conflict*). The SemEval2016 dataset is retrieved from community question answer that aims to classify answers as *good*, *bad* and *potentially useful*. We extract 8407 samples from the original dataset to follow a fair distribution. The detailed distributions of two datasets are shown in Tables 1 and 2. We use both word2vec [38] and Bert [39] to initialize word embeddings. FudanNLP[1] is used to segment Chinese sentences and words.

Hyper-parameters: Adam [41] is adopted to minimize $J(\theta)$ given in Eq. (15). Specifically, learning rate is 0.001, the lengths of sub-question and sub-answer are set to 15 and 20 respectively, and the detailed parameters are given in Table 3.

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

$$\begin{aligned} &Accuracy = \tfrac{T}{N}, \\ &F1 = \tfrac{2PR}{P+R}, \end{aligned} \tag{16}$$

where T is the number of correctly predict samples, N is the total number of samples, P is the positive predictive value, and R is the recall value.

4.1. Baselines

To evaluate the performance of the proposed HGAT, the following baselines are chosen for comparison:

Bi-LSTM [42]: a bidirectional LSTM model that treated a QA text pair as a successive sequence.

ATT-LSTM [43]: the approach conducted the attention mechanism for the aspect-level sentiment analysis. In our implementation, we directly adopted the hidden states of LSTM to yield the attention.

TextCNN [44]: a CNN model that distinguished between important and comparatively inconsequential design decisions for sentence classification. In our implementation, we set the kernel sizes to 3, 4 and 5.

TextGCN [33]: a GCN model that treated each word as a node for text classification. In our implementation, we treated a QA text pair as a successive sequence.

CDT [45]: a state-of-the-art approach that averaged aspect representations for the aspect-level sentiment analysis. In our implementation, we adopted a two-layer graph convolutional network on QA text pairs and averaged question representations for final outputs.

DPCNN [46]: a deep pyramid CNN model that represented long-range associations in texts and obtained more global information.

BiMPM [47]: a state-of-the-art approach that matched sentences from multiple perspectives. In our implementation, we treated the question and the answer as two successive sequences, and conducted the matching mechanism between two sequences.

HMN [2]: the approach first segmented QA pair into sub-sentences, and then employed both QA bidirectional matching mechanism to capture inter-relations.

Fine-grained Bert [48]: the approach classified sentiments by the fine-grained Bert model with a softmax classifier.

TD-GAT [36]: the approach utilized GAT with the consideration of word dependencies for the aspect-level sentiment classification. In our implementation, we ignored aspect relations and treated a QA text pair as a whole sequence.

HGAT-word2vec: our model with word2vec embedding.

HGAT-Bert: our model with pre-trained Bert embedding.

4.2. Results

Table 4 presents the overall comparison results in all domains, showing that:

1. Compared to the previous models (Bi-LSTM, ATT-LSTM, TextCNN and et al.) that treat a QA text pair as a successive sequence, HMN and HGAT achieve better performance on *Taobao* datasets. The reason is that HMN and HGAT segment sentences and can capture inter-relations among sub-sentences.
2. Though Fine-grained Bert treats the QA text pair as a sequence, it outperforms BiMPM and HMN due to its ability for modeling the long-term dependency of salient information with the Bert structure.
3. HGAT achieves about 2.4% to 5.5% improvement over than TD-GAT in terms of the overall performance, showing that HGAT can better capture the relations among sub-sentences by applying the sentence-level GAT.
4. HGAT outperforms all baselines on three domains and shows the effectiveness of the proposed method.

Table 5 reports the detailed results of each class on SemEval2016 dataset, where the pre-trained Skip-Gram is adopted to initialize word embeddings. We compare HGAT with TextCNN, TextGCN, DPCNN and HMN. It can be seen that HGAT and HMN achieve top 2 results since two methods attempt to match relations between sub-sentences. However, TextGCN performs worse than TextCNN due to the reason that the output of GCN using general averaging may cause semantic ambiguity. Meanwhile, in TextCNN, we concatenate three different kernels (3, 4, 5) to capture different information with maxpool operation, alleviating the dilution of important features.

4.3. Ablation study

Furthermore, in order to validate the effects of different parts in HGAT, we experiment HGAT with five settings: 1) replacing Bi-LSTM with one layer fully connected network; 2) removing the word-level GAT; 3) removing the sentence-level GAT; 4) replacing the sigmoid self-attention with the original self-attention mechanism; and 5) replacing the improved GAT with the original GAT. According to Table 6, several results are highlighted:

1. Bi-LSTM module has the most significant influence with about 8.4% to 12.9% accuracy reduction since it can extract bidirectional contextual information.
2. Compared with the word-level GAT, the sentence-level GAT has more impact on the overall results, demonstrating that the relations among sub-sentences are more important than the ones among words.
3. Sigmoid self-attention mechanism improves about 0.6% to 1.8% on M-F1 and 0.1% to 1.8% on ACC with its ability of capturing salient sentiment information.
4. The improved GAT that considers both the learned weight and the prior graph edge weight performs better than the original GAT.
5. Each module in HGAT helps improve the accuracy of sentiment analysis, especially for sentence-level GAT module that captures inter-relations and intra-relations among sub-sentences.

[1] https://github.com/FudanNLP/fnlp/

Table 1
The detailed data distributions of *Taobao*.

		Positive	Negative	Neutral	Conflict	Total
Beauty	train	2950	792	4017	241	8000
	test	726	189	1008	77	2000
Shoe	train	3234	648	3787	331	8000
	test	791	171	957	81	2000
Electronic	train	3073	805	3701	421	8000
	test	734	212	947	107	2000

Table 2
The detailed data distributions of SemEval2016.

		Good	Bad	Potential Useful	Total
SemEval2016	train	2486	2499	2015	7000
	test	514	501	392	1407

Table 3
Parameter Settings.

Description	Symbol	Value
Class number	c	4
Batch size	b	32
Epoch number	e	20
Sub-question length	q_{len}	15
Sub-answer length	a_{len}	20
LSTM hidden size	d'	128
Word-level GAT hidden size	d^{hid}	256
Sentence-level GAT hidden size	$d^{hid'}$	256
Head num	h_n	4
Sigmoid self-attention size	d^{att}	256
Train size	t_r	8000
Test size	t_e	2000
Edge weight	λ	0.2
Learning rate	lr	0.001
Dropout probability	p	0.7
l_2 penalty	β	0.0001

4.4. Training loss curves

We discuss the training loss curves with different λ in this subsection. As can be seen from Fig. 5, the training loss fluctuates a lot when $\lambda = 0$, and the training loss converges very slowly with the minimum loss value 20. When λ is set to 0.2, the loss becomes smooth and can be well trained considering the prior edge weight. However, when λ is set to 0.5, the curve becomes a bit unstable, which is caused by the penalty of the larger scale of the prior edge weight.

4.5. Effects of the model depth

To acquire the impact of the depth of GAT layers, we conduct several experiments on the *Shoe* domain with different GAT layers. In Fig. 6, the model achieves the best M-F1 and ACC when the number of depth is 2. Besides, the results show that the model is trainable when the depth of GCNs is less than 3 layers, which implies that the semantic relations among sentences can be captured within 3-hops away. However, the model presents a sharp downtrend when the number of GAT layers is bigger than 3. One possible reason is that the model becomes more complex and needs more data for training.

4.6. Effects of the prior edge weight

We further explore the impact of different prior edge weight on the *Shoe* domain, where λ increases from 0 to 1. It can be seen from Fig. 7 that the model performs worst when $\lambda = 1$ because the edge weights are completely learned by the dependency relations and no attention weight is needed. Meanwhile, as discussed in Section 4.3, the improved GAT is replaced by the original GAT since no prior edge knowledge is considered when $\lambda = 0$. Additionally, the model performs best when considering about 20% of the prior edge knowledge, showing the effectiveness of the improved GAT.

4.7. Error analysis

To analyze the limitation of HAGT, the percentage of misclassification of four polarities is presented in Fig. 8. Since the semantic of neutral sample is uncertain, it has the largest percentage of error classification. However, the conflict class ranks the lowest among all domains due to its small test samples.

Specifically, Fig. 9 describes the detailed error cases on the *shoe* domain. For the positive case "*Q: Are the shoes good? Are they worth buying? A: Warmth is ok, but worse than the brand.*", HGAT pays more attention to the relations between two sub-answers, thus misclassifies the sample as conflict. Meanwhile, for the second case "*Q: Will it deform in several months? Will it be delaminated or pleated? A: Just ok.*", HGAT conducts the wrong prediction due to the difficulty in comprehension of the answer "*Just ok*". For the neutral case, though the answer just declares the fact, HGAT misclassifies it with larger weights of sub-answers. For the last case, since the answer is not very clear to the question, HGAT classifies the sample as negative with the sub-answer "*poor air permeability*".

Table 4
Comparison of several baselines on *Taobao* QA datasets. The best results of each domain are in bold and the results with † are retrieved from [2].

Model	Beauty		Shoe		Electronic		Average	
	M-F1	ACC	M-F1	ACC	M-F1	ACC	M-F1	ACC
Bi-LSTM	0.535	0.722	0.527	0.752	0.574	0.723	0.545	0.732
ATT-LSTM	0.517	0.731	0.565	0.749	0.561	0.718	0.548	0.733
BiMPM	0.561	0.751	0.591	0.765	0.578	0.740	0.577	0.752
TextCNN	0.625	0.758	0.648	0.793	0.638	0.768	0.637	0.773
TextGCN	0.587	0.744	0.607	0.751	0.635	0.753	0.610	0.749
CDT	0.616	0.743	0.631	0.765	0.654	0.768	0.634	0.759
DPCNN	0.625	0.749	0.681	0.790	0.683	0.785	0.663	0.775
HMN	0.598†	0.776†	0.683†	**0.827**†	0.640†	0.779†	0.640	0.794
HGAT-word2vec	**0.657**	**0.791**	**0.703**	0.820	**0.719**	**0.803**	**0.693**	**0.805**
Fine-grained Bert	0.651	0.780	0.673	0.802	0.612	0.775	0.645	0.785
TD-GAT Bert	0.661	0.796	0.711	0.821	0.710	0.806	0.694	0.808
HGAT-Bert	**0.728**	**0.829**	**0.774**	**0.846**	**0.745**	**0.822**	**0.749**	**0.832**

Table 5
Results of each class on SemEval2016 dataset.

Model	Good	Bad	Potential Useful	Overall	
	ACC	ACC	ACC	M-F1	ACC
TextCNN	0.650	0.682	0.545	0.634	0.638
TextGCN	0.650	0.695	0.541	0.621	0.629
DPCNN	0.709	0.700	0.583	0.666	0.673
HMN	0.713	0.717	0.599	0.673	0.682
HGAT	**0.727**	**0.726**	**0.653**	**0.685**	**0.703**

Table 6
Comparison of all modules in HGAT.

	Beauty		Shoe		Electronic		Average	
	M-F1	ACC	M-F1	ACC	M-F1	ACC	M-F1	ACC
HGAT-Bert	**0.728**	**0.829**	**0.774**	**0.846**	**0.745**	**0.822**	**0.749**	**0.832**
-w/o Bi-LSTM	0.508	0.700	0.565	0.762	0.597	0.723	0.557	0.728
-w/o word-GAT	0.720	0.817	0.746	0.841	0.745	0.817	0.737	0.825
-w/o sen-GAT	0.707	0.811	0.733	0.838	0.727	0.811	0.722	0.820
-w/o sigmoid	0.714	0.811	0.756	0.840	0.739	0.821	0.736	0.824
-w/o prior weight	0.707	0.813	0.770	0.844	0.743	0.814	0.740	0.824

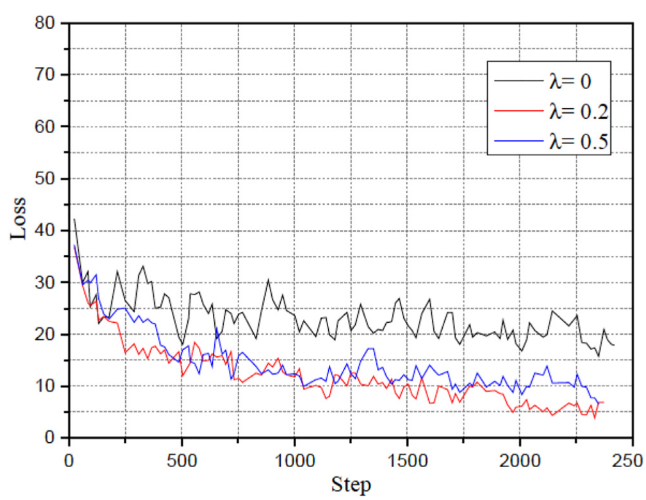


Fig. 5. Training loss with different λ on the *shoe* domain.

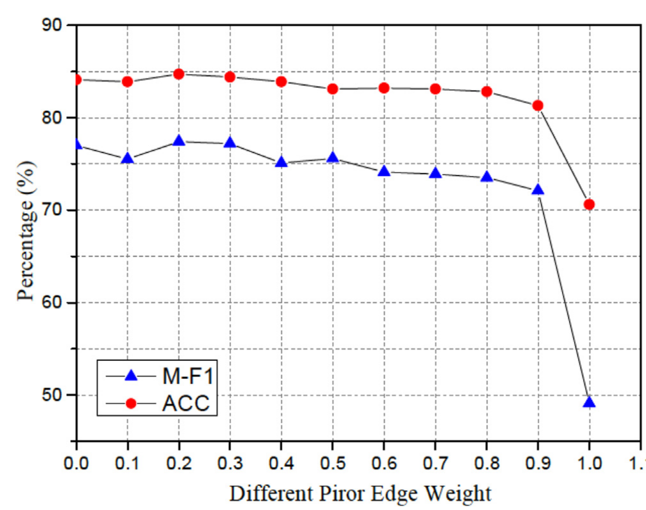


Fig. 7. Effects of Different prior edge weights.

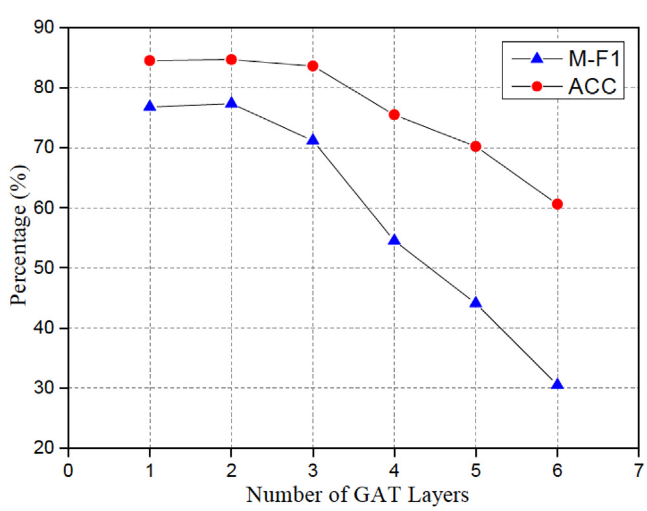


Fig. 6. Effects of Different Number of GAT layers.

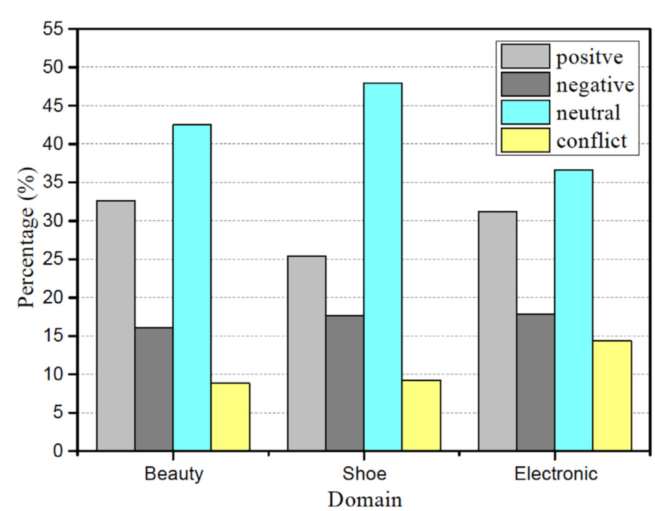


Fig. 8. Error classification percentages on three domains.

Domain	Label	Percentage	Example	Prediction
Shoe	positive	25.14	Q: 鞋子好不? 值得买不? Are the shoes good? Are they worth buying? A: 保暖性还不错, 比品牌的要差 Warmth is ok, but worse than the brand.	conflict
	negative	17.63	Q: 几个月后会变形吗?会不会脱胶，起皱? Will it deform in several months? Will it be delaminated or pleated? A: 一般 Just ok.	neutral
	neutral	47.98	Q: 布的好还是皮的好? Is cloth or leather better? A: 皮的好打理，布的透气好。 Leather is better at caring, cloth is better at air permeability.	positive
	conflict	9.25	Q: 这鞋的质量行么，尤其是鞋底可以么? Is the quality of this shoe okay? especially for the sole? A: 还行，不透气。 OK, poor air permeability.	negative

Fig. 9. Error analysis on the *Shoe* domain.

4.8. Visualization of attention

To gain a more comprehensive understanding of the proposed HGAT, we present the visualization of attention weights. In Fig. 10, we visualize attention weights column by column with blue color, and intra-relations are marked with red rectangles. It can be seen that the salient words can be well marked when considering syntactic information. Meanwhile, both GAT and the improved GAT can assign similar weight distributions. For intra-relations, the second column q_2 assigns larger weight (0.401) to sub-sentence q_1 and the third column a_1 assigns larger weight (0.227) to sub-sentence a_2 in Fig. 10(b). For inter-relations, the third column a_1 and the forth column a_2 both assign larger weights to sub-sentences q_1 and q_2 in two sub-figures.

5. Conclusion

In this paper, we explore the former edge knowledge in GAT and propose a novel hierarchical graph attention architecture to capture inter-relations and intra-relations for QA-style sentiment classification. HGAT embeds the dependency graph into word-level GAT and uses the sigmoid self-attention mechanism to manage information flow between the multi-level GAT, making it capable

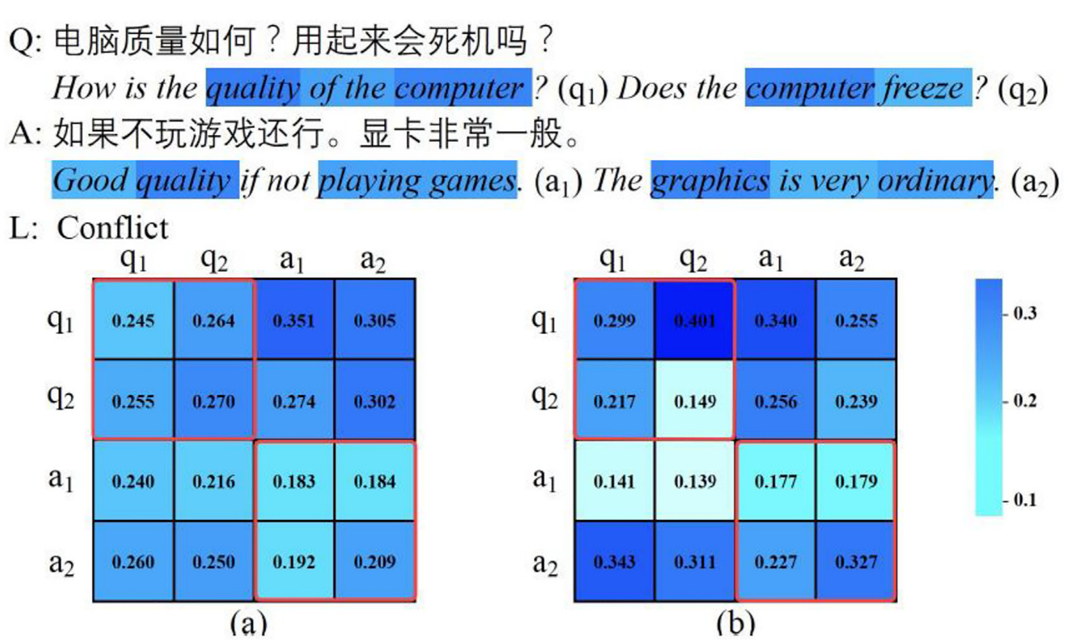


Fig. 10. The visualization of attention weights. (a) attention weights of the original GAT. (b) attention weights of the improved GAT.

to explore abundant syntactic relations among salient sentiment words and multi-grained relations between question and answer. Extensive experiments show that HGAT achieves significant improvements for QA-style sentiment classification compared with several baselines.

Future work will take more structure information about sub-sentences into consideration. Since this work treats all sub-sentences as a strongly connected graph, the universal dependency relations could be further studied. We hope to further explore how prior edge information can be better integrated in GAT, and extend HGAT to analyze intentions and emotions in multi-round dialogue systems.

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.

Acknowledgements

We greatly appreciate anonymous reviewers and the associate editor for their valuable and high quality 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, 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] T. Ma, H. Rong, Y. Hao, J. Cao, Y. Tian, M.A. Al-Rodhaan, A novel sentiment polarity detection framework for chinese, IEEE Transactions on Affective Computing doi:10.1109/TAFFC.2019.2932061..
[2] C. Shen, C. Sun, J. Wang, Y. Kang, S. Li, X. Liu, L. Si, M. Zhang, G. Zhou, Sentiment classification towards question-answering with hierarchical matching network, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, 2018, pp. 3654–3663.
[3] J. Wang, L.-C. Yu, K.R. Lai, X. Zhang, Dimensional sentiment analysis using a regional cnn-lstm model, in: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), 2016, pp. 225–230.
[4] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, E. Hovy, Hierarchical attention networks for document classification, in: Proceedings of the 2016 conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2016, pp. 1480–1489.
[5] S. Poria, D. Hazarika, N. Majumder, R. Mihalcea, Beneath the tip of the iceberg: Current challenges and new directions in sentiment analysis research, IEEE Transactions on Affective Computing doi:10.1109/TAFFC.2020.3038167..
[6] J. Li, B. Chiu, S. Shang, L. Shao, Neural text segmentation and its application to sentiment analysis, IEEE Transactions on Knowledge and Data Engineering doi:10.1109/TKDE.2020.2983360..
[7] D. Yin, T. Meng, K.-W. Chang, SentiBERT: A transferable transformer-based architecture for compositional sentiment semantics, in: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, 2020, pp. 3695–3706.
[8] 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.
[9] 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..
[10] J. Wen, G. Zhang, H. Zhang, W. Yin, J. Ma, Speculative text mining for document-level sentiment classification, Neurocomputing 412 (2020) 52–62.
[11] 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, Volume 1 (Long Papers), 2018, pp. 541–550.
[12] 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 (EMNLP-IJCNLP), 2019, pp. 3423–3428.
[13] T. Zhang, X. Gong, C.P. Chen, Bmt-net: Broad multitask transformer network for sentiment analysis, IEEE Transactions on Cybernetics doi:10.1109/TCYB.2021.3050508..
[14] A. Nazir, Y. Rao, L. Wu, L. Sun, Issues and challenges of aspect-based sentiment analysis: A comprehensive survey, IEEE Transactions on Affective Computing doi:10.1109/TAFFC.2020.2970399..
[15] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, 2018, pp. 5876–5883..
[16] H. Peng, L. Xu, L. Bing, F. Huang, W. Lu, L. Si, Knowing what, how and why: A near complete solution for aspect-based sentiment analysis, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, 2020, pp. 8600–8607..
[17] J. Du, L. Gui, R. Xu, Y. Xia, X. Wang, Commonsense knowledge enhanced memory network for stance classification, IEEE Intelligent Systems 35 (4) (2020) 102–109.
[18] E. Cambria, Y. Li, F.Z. Xing, S. Poria, K. Kwok, Senticnet 6: Ensemble application of symbolic and subsymbolic ai for sentiment analysis, in: Proceedings of the 29th ACM International Conference on Information & Knowledge Management, 2020, pp. 105–114.
[19] Y. Liu, L. Ji, R. Huang, T. Ming, C. Gao, J. Zhang, An attention-gated convolutional neural network for sentence classification, Intelligent Data Analysis 23 (5) (2019) 1091–1107.
[20] J. Wei, J. Liao, Z. Yang, S. Wang, Q. Zhao, Bilstm with multi-polarity orthogonal attention for implicit sentiment analysis, Neurocomputing 383 (2020) 165–173.
[21] F. Huang, X. Li, C. Yuan, S. Zhang, J. Zhang, S. Qiao, Attention-emotion-enhanced convolutional lstm for sentiment analysis, IEEE Transactions on Neural Networks and Learning Systems doi:10.1109/TNNLS.2021.3056664..
[22] C. Du, H. Sun, J. Wang, Q. Qi, J. Liao, T. Xu, M. Liu, Capsule network with interactive attention for aspect-level sentiment classification, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 5492–5501.
[23] H. Peng, Y. Ma, Y. Li, E. Cambria, Learning multi-grained aspect target sequence for chinese sentiment analysis, Knowledge-Based Systems 148 (2018) 167–176.
[24] 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.
[25] 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, 2016, pp. 3298–3307.
[26] 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.
[27] J. Wang, C. Sun, S. Li, X. Liu, L. Si, M. Zhang, G. Zhou, Aspect sentiment classification towards question-answering with reinforced bidirectional attention network, in: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3548–3557.
[28] F. Monti, M. Bronstein, X. Bresson, Geometric matrix completion with recurrent multi-graph neural networks, in: Advances in Neural Information Processing Systems, 2017, pp. 3697–3707.
[29] J. Hewitt, C.D. Manning, A structural probe for finding syntax in word representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4129–4138.
[30] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, M. Wang, Lightgcn: Simplifying and powering graph convolution network for recommendation, in: Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, 2020, pp. 639–648.
[31] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: Proceedings of 2017 International Conference on Learning Representations, 2017, pp. 1–14.
[32] D. Marcheggiani, I. Titov, Encoding sentences with graph convolutional networks for semantic role labeling, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 1506–1515.
[33] L. Yao, C. Mao, Y. Luo, Graph convolutional networks for text classification, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, 2019, pp. 7370–7377..
[34] M. Schlichtkrull, T.N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, M. Welling, Modeling relational data with graph convolutional networks, in: European Semantic Web Conference, Springer, 2018, pp. 593–607.
[35] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph attention networks, in: Proceedings of 2018 International Conference on Learning Representations, 2018, pp. 1–12.

[36] B. Huang, K.M. Carley, Syntax-aware aspect level sentiment classification with graph attention networks, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing, 2019, pp. 5469–5477.
[37] 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.
[38] T. Mikolov, K. Chen, G. Corrado, J. Dean, Efficient estimation of word representations in vector space, in: Proceedings of 2018 International Conference on Learning Representations, 2013, pp. 1–12.
[39] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding (2019) 4171–4186..
[40] D. Chen, C.D. Manning, A fast and accurate dependency parser using neural networks, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 740–750.
[41] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: Proceedings of 2017 International Conference on Learning Representations, 2014, pp. 1–15.
[42] E. Kiperwasser, Y. Goldberg, Simple and accurate dependency parsing using bidirectional lstm feature representations, Transactions of the Association for Computational Linguistics 4 (2016) 313–327.
[43] 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.
[44] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014, pp. 1746–1751.
[45] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Aspect-level sentiment analysis via convolution over dependency tree, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 5683–5692.
[46] R. Johnson, T. Zhang, Deep pyramid convolutional neural networks for text categorization, in: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 562–570.
[47] Z. Wang, W. Hamza, R. Florian, Bilateral multi-perspective matching for natural language sentences, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 4144–4150.
[48] M. Munikar, S. Shakya, A. Shrestha, Fine-grained sentiment classification using bert, in: 2019 Artificial Intelligence for Transforming Business and Society (AITB), vol. 1, 2019, pp. 1–5.

Jiandian Zeng received the BSc degree in computer science and technology from Jianghan University, Wuhan, China, in 2014, and the MSc degree in computer technology from Huaqiao University, Xiamen, China, in 2018. He is currently pursuing the Ph.D degree in Department of Computer and Information Science, Faculty of Science and Technology, University of Macau, Macau, China. His research interests include natural language processing and sentiment analysis, mobile computing.

Tinayi Liu received the BSc degree in the school of Information and Communication Engineering, University of Electronic Science and Technology of China in 2017. He is currently working toward the Ph.D. degree in computer science at Shanghai Jiao Tong University, China. His research interests include relation extraction, natural language processing and knowledge engineering.

Weijia Jia is currently a director of BNU-UIC Institute of Artificial Intelligence and future Networks,Beijing Normal University at Zhuhai, while taking no-pay leave from the position of Zhiyuan Chair Prof from Shanghai Jiao Tong University, China (he received 2013 China 1000 Talent Award). He received BSc/MSc from Center South University, China in 82/84 and Master of Applied Sci./PhD from Polytechnic Faculty of Mons, Belgium in 92/93, respectively, all in computer science. For 93-95, he joined German National Research Center for Information Science (GMD) in Bonn (St. Augustine) as research fellow. From 95-13, he worked in City University of Hong Kong as a full professor in Computer Science Dept. His research interests include smart city; next generation IoT, knowledge graph constructions; multicast and any-cast QoS routing protocols, wireless sensor networks and distributed systems. In these fields, he has over 400 publications in the prestige international journals/conferences and research books and book chapters. He has served as area editor for various prestige international journals, chair and PC member/keynote speaker for many top international conferences. He is the Senior Member of IEEE and the Member of ACM.

Jiantao Zhou (Senior Member, IEEE) received the B.Eng. degree from the Department of Electronic Engineering, Dalian University of Technology, in 2002, the M.Phil. degree from the Department of Radio Engineering, Southeast University, in 2005, and the Ph.D. degree from the Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, in 2009. He held various research positions with the University of Illinois at Urbana-Champaign, The Hong Kong University of Science and Technology, and the McMaster University. He is currently an Associate Professor with the Department of Computer and Information Science, Faculty of Science and Technology, University of Macau. His research interests include multimedia security and forensics, multimedia signal processing, artificial intelligence and big data. He holds four granted U.S. patents and two granted Chinese patents. He has coauthored two articles that received the Best Paper Award at the IEEE Pacific-Rim Conference on Multimedia in 2007 and the Best Student Paper Award at the IEEE International Conference on Multimedia and Expo in 2016. He is an Associate Editor of the IEEE TRANSACTIONS ON IMAGE PROCESSING.