Neurocomputing 518 (2023) 373–383

Contents lists available at ScienceDirect

Neurocomputing

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

Learn from structural scope: Improving aspect-level sentiment analysis with hybrid graph convolutional networks

Lvxiaowei Xu [a], Xiaoxuan Pang [b], Jianwang Wu [a], Ming Cai [a,*], Jiawei Peng [a]

[a] Department of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
[b] Alibaba Group-China Digital Commerce-Tao Cai Cai-Tech Dept-Data Intelligence Growth, Hangzhou 310027, China

ARTICLE INFO

Article history:
Received 7 May 2022
Revised 15 September 2022
Accepted 24 October 2022
Available online 5 November 2022
Communicated by Zidong Wang

Keywords:
Aspect-level sentiment analysis
Syntactic parse tree
Hybrid graph convolutional network
Cascaded model

ABSTRACT

Aspect-level sentiment analysis aims to determine the sentiment polarity towards a specific target in a sentence. The main challenge of this task is to effectively model the relation between targets and sentiments so as to filter out noisy opinion words from irrelevant targets. Most recent efforts capture relations through target-sentiment pairs or opinion spans from a word-level or phrase-level perspective. Based on the observation that targets and sentiments essentially establish relations following the grammatical hierarchy of phrase-clause-sentence structure, it is hopeful to exploit comprehensive syntactic information for better guiding the learning process. Therefore, we introduce the concept of *Scope*, which outlines a structural text region related to a specific target. To jointly learn structural *Scope* and predict the sentiment polarity, we propose a hybrid graph convolutional network (HGCN) to synthesize information from constituency tree and dependency tree, exploring the potential of linking two syntax parsing methods to enrich the representation. Experimental results on five public datasets illustrate that our HGCN model outperforms current state-of-the-art baselines. More specifically, the average accuracy/ F_1 score improvements of our HGCN compared to baseline models on Restaurant 14, 15 and 16 are 2.46%/5.36%, 2.25%/5.70% and 1.73%/5.50%, while the performance improvements are 3.32%/4.30% and 2.50%/3.08% on the Laptop and Twitter datasets, respectively. Furthermore, when cascaded to five models, our method has significantly improved their performances by simplifying the sentence from multiple targets to a single one.

1. Introduction

Aspect-level sentiment analysis (ALSA) is a fine-grained classification task, aiming at identifying opinion polarities towards specific entities called targets. Fig. 1 shows a review commenting on the restaurant with two target terms "*menu*" and "*food*". The sentiment polarities over them are positive and negative respectively. Since multiple targets may appear in one sentence and convey different sentiments, the main challenge of ALSA task is to effectively filter out noisy or misleading opinion words from irrelevant targets and extract distraction-free text from opinion expressions accurately.

Considerable efforts have been devoted to overcoming the difficulties. Their works can be broadly divided into the following three categories. **Attention-based models:** Attention mechanism has been widely adopted to learn target-specific features [1–4] so as to help models concentrate on crucial parts of a sentence for sentiment classification. However, the attention scatters across the whole sentence and is prone to attend on noisy opinion words from other targets. **Syntax-based models:** Syntax-based method aims to model syntactic structures that exist in the sentence. Among them, dependency tree [5–9] has been mostly used to capture the structural dependencies between targets and the sentiment expressions. Nevertheless, multi-hop dependency relations between target and unrelated opinion words are susceptible to confusing and misleading the model. **Span-based models:** Span-based models [10–12] apply a divide-and-conquer strategy to handle the ALSA task. Specifically, the model extracts an opinion span towards each target based on a soft or hard selection approach, and then focuses on the phrase-level features for sentiment classification. Therefore, it is difficult to generate structurally complete spans for complicated sentences due to lack of syntactic information.

To tackle aforementioned problems, we introduce a concept of structural *Scope*, which is a well-structured and continuous text region expressing target-specific opinion. *Scope* delineates the

* Corresponding author.
E-mail addresses: xlxw@zju.edu.cn (L. Xu), wujw@zju.edu.cn (J. Wu), cm@zju.edu.cn (M. Cai), pengjw@zju.edu.cn (J. Peng).

https://doi.org/10.1016/j.neucom.2022.10.071
0925-2312/

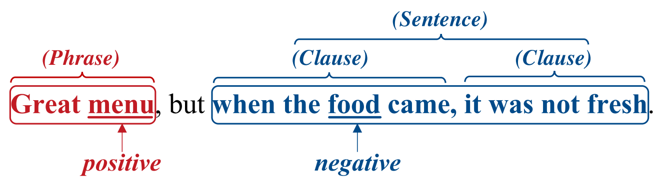


Fig. 1. An instance of a restaurant review with two target terms which have opposite sentiments. The *Scopes* are highlighted with a box and contain their corresponding target.

exact context of each target so as to alleviate the distraction problem caused by opinion words from other targets. Compared with opinion span, *Scope* is characterized by a well-defined structure, which highlights the learning of various syntactic features (e.g., phrase, clause and sentence) related to the target to ensure structural and semantic completeness. Fig. 1 demonstrates an example containing two *Scopes* with their corresponding targets in different structures. It combines a phrase-level relation (a noun phrase connecting the target "*menu*" and opinion word "*great*") and a sentence-level relation (the target "*food*" linking the remote opinion words "*not fresh*" outside of the local clause through a coreference relation). This demonstrates that the targets and sentiments establish relations following the grammatical hierarchy of phrase-clause-sentence structure, which substantially filters out noisy opinion words from irrelevant targets.

Three observations motivate us to aggregate syntactic information of constituency tree and dependency tree to learn structural *Scope*. First, constituency parsing decomposes a sentence into phrases and combines them hierarchically into clauses and sentences, which naturally follows the phrase-clause-sentence structure of *Scope* and can serve as the foundation to encode structural information. Second, dependency parsing is capable of capturing long-range dependencies such as coreference relation, which can help the model obtain the sentence-level connection. As for constituency parsing, it displays the entire sentence structure and relations, while dependency parsing explores relations between words.Third, dependency tree can actually not only parse the dependency relations, but also derive constituent boundaries, which is critical for modeling *Scope*. However, this requires unbounded number of hops over dependency tree to determine the boundaries that entail more layers of networks (e.g., GCN) for obtaining syntactic representations. It would be resource consuming and impractical. In this perspective, constituency tree and dependency tree present two complementary sides of syntactic relations in sentences. Therefore, it would be beneficial to incorporate constituency tree and dependency tree into the model to learn more comprehensive syntactic relations.

On the basis of above analysis, we present a hybrid graph convolutional network (HGCN) to jointly learn the structural *Scope* and determine the sentiment polarity. For encoding the constituency tree, we propose the constituency GCN module and the computation is performed in three stages. First, the representations of the constituents are composed by the representations of the words and the embeddings of constituent labels. Second, graph convolutional networks (GCNs) [13] are exploited to learn constituent representations. Third, compared to the previous scheme of encoding constituent tree [14], they utilize an additional BiLSTM module to decompose back the constituency syntax into words, which significantly increases the computational complexity. Therefore, we propose a constituent-token attention mechanism to decompose back the information to word representations which can assign constituency syntax more reasonably and reduce the computational complexity. Regarding the dependency tree, we utilize a relational graph convolutional network (RGCN) to obtain syntactic representation with labeled edges. Finally, we incorporate such heterogeneous information to determine the *Scope* region through the conditional random field (CRF) [15], and to identify the opinion polarities as well.

The extensive experiments are conducted on the SemEval and Twitter datasets, and the results demonstrate the effectiveness of our HGCN model that achieves superior performance to the baseline methods. Meanwhile, *Scope* can significantly improve the performance of other baseline models when they are cascaded with ours. The analysis shows that these phenomena are aligned with our intuition.

The main contributions of this work include:

- We introduce the concept of *Scope*, which enables our model to focus on text spans that are directly related to the specific target, rather than dispersing over the entire sentence. We utilize a syntactically-informed method to capture the relations between targets and sentiment polarities, which could generate distraction-free opinion expressions for sentiment polarity identification.
- We present a new HGCN model to synthesize constituency relations and dependency relations, exploiting the complementary strengths of two syntax parsing methods to learn syntactic relations for *Scope*.
- We propose a constituency GCN module in HGCN equipped with constituent-token (C-T) attention mechanism to encode structural representations from constituency tree, which has a lower resource consumption compared to the previous method.
- Our model can be used as a pre-module to cascade other models to enhance their performance.

2. Related work

Traditional sentiment analysis is mainly focuses on sentence level or document level while aspect-level sentiment analysis is a more fine-grained sentiment task. Therefore, it has received extensive research attentions in recent years. Early works focus on extracting hand-craft features for statistical classification models, such as SVM-based models [16]. They are usually labor intensive. Recently, with the development of deep learning methods, neural networks [17–20] are widely used in sentiment analysis for their ability to automatically learn feature representations from data. Furthermore, some methods have been applied to this fine-grained task by connecting target terms with their opinion words.

In general, opinion words are not particularly far from the target term. Thus among neural networks methods, most recent works utilize attention-based models to discriminate sentiment polarity via words nearby the target term implicitly. As aspect-aware approaches, Wang et al. [1] apply attention mechanism on LSTM with aspect representations to determine sentiment information relating to target term. Huang et al. [3] are inspired by attention-over-attention structure from machine translation that can bidirectional concern with aspect term and sentence simultaneously. Fan et al. [4] propose a multi-grained attention network with both fine-grained and coarse-grained attentions to capture word-level interactions between target term and context to mitigate information loss in coarse-grained attention mechanisms. Xu et al. [21] propose the MAN model which utilizes dual-attention layers to obtain representations of words and capture interactive information between target and context, facilitating the determination of the sentiment polarity. These methods prove to be more efficient than hand-craft features based methods in aspect-level sentiment analysis. However, the main disadvantage of them is that attention scatters across the whole sentence and may mislead model to identify the sentiment polarity.

Besides attention mechanism, the state-of-the-art models have injected syntactic information into neural networks. Recent researches are concerned with leveraging dependency parse tree to establish the connection between target terms and opinion words. For example, Zhang et al. [5] and Sun et al. [6] propose the models to apply graph convolutional network for encoding word representations from dependency parse tree and then obtain aspect-oriented features with specific techniques for judging sentiment polarity. After that, Wang et al. [7] prune the dependency tree to retain only edges with the target terms to reduce complexity, and then propose a relational graph attention network (R-GAT) to encode the pruned dependency trees. Zhang and Qian [8] propose a BiGCN model based on graph convolutional network to construct the global lexical graph and a concept hierarchy to synthesize syntactic dependency information. Li et al. [9] incorporate syntactic structure and semantic relevance to generate features with orthogonal and differential regularizers to capture semantic correlations between words precisely. In addition, some researchers [22–24] take pre-trained language model (e.g., BERT [25]) as the encoder to possess abundant syntactic information for sentiment polarity determination.

Some other efforts try to extract snippet of opinion words directly from the sentence with external annotated data. Then these models determine the sentiment polarity of target based on opinion words. Wang and Lu [10] propose a segmentation attention model to distill the sentiment information from the sentence, and then corporate conditional random field layer to capture the structural dependencies from opinion words. Hu et al. [11] exploit an alternative hard-selection approach that determines the start and end positions of the opinion words by self-critical reinforcement learning based on pre-trained language model. Xu et al. [12] present a neat and effective multiple CRFs based attention model to extract aspect-specific opinion spans.

Furthermore, in order to obtain constituent representations, we propose the constituency GCN model and constituent-token attention in this work for incorporating constituency parse tree inspired by the graph-based models used in semantic role labeling (SRL) task [14].

3. Proposed method

3.1. Definition of Scope

To eliminate noisy opinion words from unrelated targets, we propose a syntactic structure guided modeling concept called *Scope*. In general, *Scope* contains a specific target and its dependent sentiment expression within a continuous and minimum text in a sentence. Specifically, we first define a constituent[1] candidate set C_s to hold a group of constituents (denoted as c_i), which include a consecutive text span from target to opinion words in their leaf nodes (denoted as $Node(c_i)$). Each c_i can be categorized into three levels according to grammatical definitions, which are phrase level (e.g., verb phrase, noun phrase), clause level (e.g., subordinate clauses) and sentence level. *Scope* is then defined as seeking the span of text corresponding to the constituent with the fewest leaf nodes in C_s:

$$Node(C_s[\mathrm{argmin}([\mathrm{count}(Node(c_i)), c_i \in C_s])]) \tag{1}$$

where $\mathrm{count}(\cdot)$ calculates the number of nodes excluding extraneous components (e.g., adjunct and punctuation).

3.2. Graph convolutional network

We leverage graph convolutional networks over constituency tree and dependency tree to encode structural information. GCNs are neural networks that compute the representation of a node conditioned on its neighboring nodes in a given graph. Each node is updated by aggregating the propagated message from the neighboring nodes through multilayer GCNs. Given a graph $\mathscr{G}$ contains sets of nodes $\mathscr{V}$ and edges $\mathscr{E}$. Following the idea of [13], we allow $\mathscr{G}$ to have self-loops. After that, we can obtain the adjacency matrix $\mathscr{A} \in \mathbb{R}^{n \times n}$ according to the $\mathscr{G}$, where n denotes the size of $\mathscr{V}$. The A_{ij} in the adjacency matrix reflects the connection between the node i and node j. Specifically, $A_{ij} = 1$ if node i is connected to node j and in the other scenario $Aij = 0$. Then GCN is able to propagate information over the paths and update node representations. In general, stacking l layers of GCN allows aggregation of information across the l-th order neighborhood. In such an operation, the representation of each node is updated with normalization factor as follow:

$$h_i^{(l+1)} = \sigma\left(\sum_{j=1}^{n} c_i A_{ij}\left(W^{(l)} h_j^{(l)} + b^{(l)}\right)\right) \tag{2}$$

where $h_j^{(l)}$ is the hidden representation for node j that is generated by l-th layer of the GCN. $W^{(l)}$ and $b^{(l)}$ are trainable parameters of weights and bias in l-th layer, respectively. c_i is the normalization factor calculated as $c_i = 1/d_i$ and d_i denotes the degree of node i in the graph, which is computed as $d_i = \sum_{j=1}^{k} A_{ij}$. And σ is a non-linear activation function ($\mathrm{RELU}(\cdot)$ [26]).

3.3. Hybrid graph convolutional network

Our model aims to extract the *Scope* relevant to the target term and predict the sentiment polarity jointly. Fig. 2 gives an overview of our HGCN model. In the aspect-level sentiment classification task, a sentence with n words $\{w_1, \ldots, w_i, w_{i+1}, \ldots, w_j, w_{j+1}, \ldots, w_n\}$ is given, where the target $\{w_i, \ldots, w_j\}$ is a subsequence of the sentence.

Then, BiLSTM [27] or BERT [25] (We replace the components of Embedding and Encoder in Fig. 2 with BERT as backbone) is applied as sentence encoder to integrate contextual information into hidden representations of words. For the BiLSTM encoder, we first embed each word into low-dimensional vector space by looking up an embedding matrix $E_{emb} \in \mathbb{R}^{d \times |V|}$, where d denotes the dimension of word embedding and $|V|$ is the size of vocabulary. Then we feed the embedding of words into BiLSTM encoder for obtaining contextual hidden representations $H = \{h_1, h_2, \cdots, h_n\}$, where $h_i \in \mathbb{R}^{2d}$ is the hidden representations from BiLSTM. As for the BERT, we construct an aspect oriented context "[CLS] sentence [SEP] target [SEP]" as input to obtain contextual hidden representations of the sentence. Moreover, there is a gap between word-piece representations of BERT and word-level based syntactic features. Therefore, we compose the representations of the sub-word as word representations with average-pooling to tackle this mismatch. Then the contextual representations are fed into the Constituency GCN (CGCN) and Dependency GCN (DGCN), respectively.

Eventually, we concatenate representations from DGCN and CGCN module to form syntactic representations. After that, a CRF layer is utilized to generate BIO-tags for modeling the *Scope* with syntactic representations and we aggregate the syntactic representations over the target terms via pooling to determine sentiment polarity jointly. In the following, we will elaborate the details of each component in our proposed HGCN model.

3.3.1. Constituency GCN (CGCN)

The constituency tree breaks a sentence into phrases, and then they can be composed hierarchically into clause and sentence level from bottom to top. Intuitively, it is reasonable to utilize constituency tree to encode structural information for modeling *Scope*.

[1] Note that we regard non-terminal node above the part-of-speech (POS) level as constituent whereas the terminal node is the word.

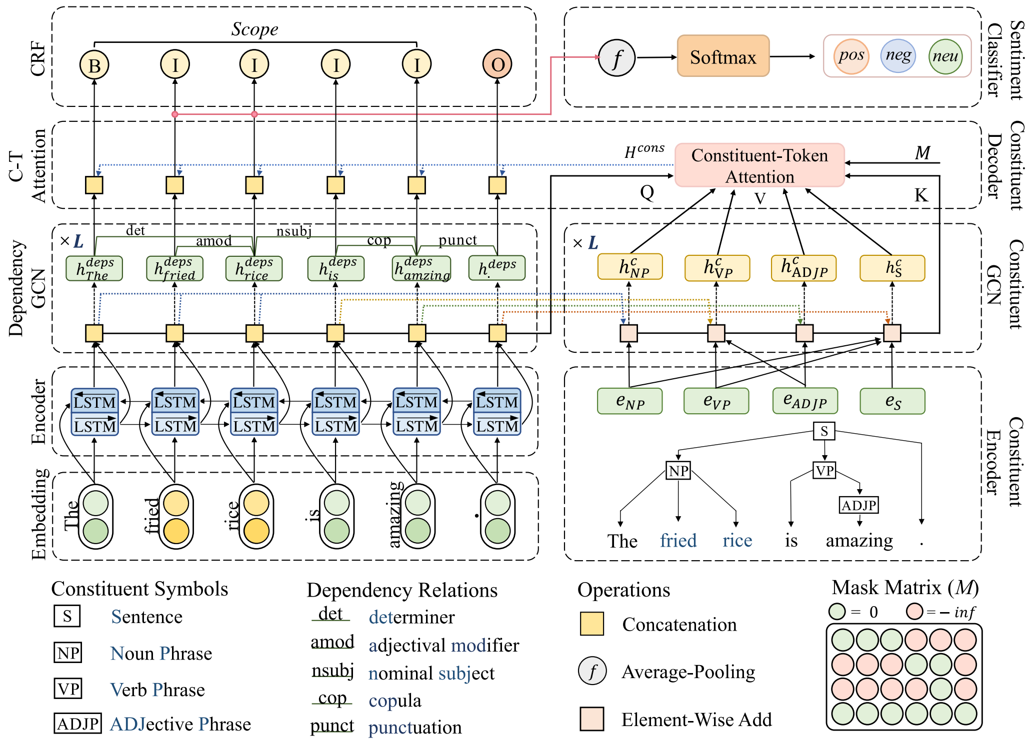


Fig. 2. Overview of hybrid graph convolutional network (HGCN).

In order to incorporate such structural constituency syntax, we propose CGCN module with multi-stage architecture for obtaining constituency-aware representations. Compared with the SpanGCN model [14], we design the constituent-token (C-T) attention to reassign the representation of words as a substitute for BiLSTM, which reduces the computational complexity and attains better performance. As shown in Fig. 2, the architecture of CGCN is composed by three sub-modules: constituent encoder, constituent GCN and constituent decoder.

Constituent Encoder. The constituency tree is composed of constituents ($\mathscr{V}_c$) and words ($\mathscr{V}_w$). In order to obtain representations of constituents, we first embed each constituent label using $E_{cons} \in \mathbb{R}^{d_c \times |\mathscr{N}|}$, where d_c is the dimension of embedding and $|\mathscr{N}|$ denotes the number of constituents. Then we assign the representation of each constituent $v \in \mathscr{V}$ in this stage relying on the embedding of the constituent label (e.g., NP or VP) and their child nodes set $\mathscr{S}_i \subseteq (\mathscr{V}_c \cup \mathscr{V}_w)$. Take Fig. 2 as an example. The constituent "S" has three child nodes. Two of them belong to $\mathscr{V}_c$ and the rest one belongs to $\mathscr{V}_w$. Its representation is obtained through averaging the embeddings of its own label, the label embedding of its two child constituents and the representations of word ".".

Constituent GCN. In this stage, it enables the interactive information passing from each constituent and ensures information of the child nodes, which can be integrated into the representation of their immediate parent node and vice versa. GCN is operated on the graph constructed by constituency tree where the nodes represent all constituents in set $\mathscr{V}_c$ and the edges correspond to the connection between constituents and their child nodes in both directions and each constituent is connected to itself on account of the self-loop. After that, structural syntactic representation $H^c = \{h^c_1, h^c_2, \ldots, h^c_m\}$ can be obtained from Eq. 2 with layer normalization applied, where m refers to the number of constituents.

Constituent Decoder. After the message passing in GCN module, structural syntactic information are injected into the representation of constituents. At this point, we propose the constituent-token (C-T) attention mechanism to assign the information back to each word. In our implementation, the attention weights indicate the contribution of the word to the representation of corresponding constituent which can be computed in constituent encoder stage. Then the allocation coefficient (i.e. attention scores) can be computed as below:

$$A^{cons} = \mathrm{softmax}\left(\frac{QW_q \times (KW_k)^{\mathrm{T}}}{\sqrt{d}} + M\right) \tag{3}$$

where the matrices Q and K are the representation of words and the representation of constituents composed by constituent encoder, respectively. W_q and W_k denote the trainable parameters, while d is the dimensionality of the hidden representations. In addition, M stands for a mask matrix to ensure that information can only be injected back to the words from their parent constituent. Here, the dot product is adopted to calculate relatedness between words and constituents.

Then we can obtain constituency representations of words based on attention scores, which can be formulated as:

$$H^{cons} = A^{cons} H^c \tag{4}$$

After the above three-stage process, we obtain the constituency representations $H^{cons} = \{h^{cons}_1, h^{cons}_2, \ldots, h^{cons}_n\}$.

3.3.2. Dependency GCN (DGCN)

The DGCN is used to capture long-range dependencies such as coreference relationships. Intuitively, different dependency relations should have distinctive impacts on adjacent nodes. However, most of the previous work fails to consider the relationship of the edges. Inspired by [28], we utilize relational graph convolutional network (RGCN) to encode dependency structure with labeled edges.

Firstly, in order to integrate the relationship information of the edges, we embed each kind of edge label using a randomly initialized embedding matrix $E_{rel} \in \mathbb{R}^{d_r \times |\mathscr{M}|}$ where d_r refers to the dimension of word embedding and $|\mathscr{M}|$ is the number of relation types.

Then we can obtain the representation of the relation between node i and node j as r_{ij}. We consider the representation r_{ij} as the gate-value to determine how much of the information will be updated to neighboring nodes. Thus the adjacency matrix A^d_{ij} takes the form:

$$A^d_{ij} = \sigma(r_{ij}W_r + b_r) \tag{5}$$

where W_r and b_r denote trainable parameters of weights and bias. $\sigma(\cdot)$ refers to the sigmoid function. After that, we regard the hidden representation H from BiLSTM or BERT encoder as initial node representations in DGCN. After that, we can compute dependency graph representation $H^{\text{deps}} = \{h_1^{\text{deps}}, h_2^{\text{deps}}, \cdots, h_n^{\text{deps}}\}$based on Eq. 2.

Finally, we apply concatenation operation on CGCN and DGCN to obtain the syntactic representations H^{syn} for *Scope* modeling. Next an average pooling operation is utilized to generate the sentiment representation H^{senti} for the ALSA task. Then $H^{senti}$is fed into a linear layer and mapped to probabilities of different polarities by a softmax function, i.e.,

$$p(y) = \text{softmax}\left(W_p H^{senti} + b_p\right) \tag{6}$$

where W_p and b_p are the learned weights and bias.

3.4. Training and testing

During the training process, we define a joint loss for selection of *Scope* and sentiment classification after obtaining the syntactic representation from CGCN and DGCN:

$$\mathcal{L}(\Theta) = \mathcal{L}_{polarity} + \gamma\mathcal{L}_{scope} + \lambda\|\Theta\|_2 \tag{7}$$

where γ is the coupling co-efficiency that regulates the two losses. Θ represents all trainable parameters in our proposed HGCN model and λ refers to the coefficient of L_2-regularization. Moreover, $\mathcal{L}_{polarity}$ denotes a standard three-way cross-entropy loss defined for the ALSA task. $\mathcal{L}_{scope}$ is defined as the negative conditional log-likelihood loss of the CRF layer for *Scope* modeling. We can formulate them as:

$$\mathcal{L}_{polarity} = -\sum_i \hat{y}_i \log p(y_i) \tag{8}$$

$$\mathcal{L}_{scope} = -\log P(\mathbf{y}|\mathbf{u}; W, b) \tag{9}$$

where $\hat{y}_i$ denotes the ground truth, i is the i-th sentiment polarity and W, b are trainable parameters of weights and bias.

In addition, as for the *Scope* selection task, Viterbi algorithm is utilized to predict the most likely label (BIO-tags) assignment of sequence during the testing process.

4. Experiments

4.1. Experimental setups

Datasets. We conduct experiments of aspect-level sentiment analysis task on five benchmark datasets. As the statistics of the datasets shown in Table 1, there are restaurant-domain (Rest14, Rest15, Rest16) datasets, laptop-domain (Lap14) taken from SemEval [29–31] and Twitter dataset [32] collected from tweets. In order to utilize *Scope*, we develop a semi-automated annotation tool and then ask 4 experienced annotators and 2 verifiers to annotate the datasets with BIO-tags of *Scope*. It is worth mentioning that our tool can automatically complete the preliminary annotation based on syntax and designed rules. Then annotators make fine refinements on this basis.

Evaluation Metrics. In ALSA task, we are required to determine the sentiment polarity of targets in the given sentences. There are three sentiment polarities in our five benchmark datasets: Positive, Negative and Neutral. We can regard the task as a multi-label classification task. Therefore, we adopt accuracy and macro F_1 score as the metrics to calculate the performance of our model and other baselines. More specifically, the accuracy is defined as:

Table 1
Statistics for the five experimental datasets.

Dataset	Total	Positive		Neutral		Negative	
		Train	Test	Train	Test	Train	Test
Rest14	4727	2164	727	637	196	807	196
Lap14	2914	976	337	455	167	851	128
Rest15	1746	912	326	36	34	256	182
Rest16	3365	1657	611	101	44	748	204
Twitter	6728	1507	172	3016	336	1528	169

$$accuracy = \frac{T}{T+F} \tag{10}$$

where T is the number of samples that the models predict correctly, while F is the number of incorrectly predicted samples. The accuracy metric indicates the proportion of correct samples predicted by the models among all samples. Thus higher accuracy can always represent better performance of the models.

However, accuracy is easily dominated by categories that include larger amount of samples. Therefore, we also employ the macro F_1 metric to measure the performance of the models for comprehensive evaluation. The macro F_1 is computed by F_1 of each category as follows:

$$Macro\,F_1 = \frac{\sum_{c=1}^{N} F_1^c}{N} \tag{11}$$

where F_i^c indicates the F_1 metric of category c, and N represents the number of categories. Concretely, F_1 metric of the c-th category can be computed as:

$$F_1^c = \frac{2P_cR_c}{P_c + R_c} \tag{12}$$

where P_c and R_c indicate the precision and recall of class c, respectively. We can formulate them as:

$$P_c = \frac{TP_c}{TP_c + FP_c}, \quad R_c = \frac{TP_c}{TP_c + FN_c} \tag{13}$$

where TP_c indicates the number of samples that are classified correctly in category c, while FP_c indicates the number of samples that are incorrectly classified into category c. FN_c indicates the number of samples with label c that are classified into other categories.

Baselines. Baselines can be categorized into the following three groups:

Attention-based method. ATAE-LSTM (ATAE) [1] uses aspect embedding and attention mechanism in ALSA tasks. **AOA** [3] utilizes two LSTMs and an interactive attention mechanism to generate representations for the aspect and sentence. **MGAN** [4] proposes a multigrained attention network to capture word-level interactions between target term and context.

Syntactic-based method. ASGCN [5] and **CDT** [6] first apply graph convolutional network for encoding aspect-specific word representations. **BiGCN** [8] adopts hierarchical graph structure to encode dependency and word co-occurrence information. **InterGCN** [24] employs a GCN over a dependency tree to learn aspect representations. **RGAT**, **RGAT + BERT** [7] utilize a relational graph attention network to encode the pruned dependency trees. **DGEDT**, **DGEDT + BERT** [33] provide a dependency graph enhanced dual-transformer network to encode heterogeneous information.

DualGCN, **DualGCN + BERT** [9] incorporate syntactic structure and semantic relevance to generate features.

Span-based method. SA-LSTM [10] presents a segmentation attention model to distill the sentiment semantics for capturing the structural dependencies with external opinion data. **MCRF-SA** [12] proposes a hard selection approach to determine the opinions by self-critical reinforcement learning with external annotated opinions data.

Training Details. We utilize Stanford Parser [34] for constituency and dependency parsing. The hidden state size of BiLSTM is turned to 100. As for our HGCN model, we utilize 300-dimensional GloVe vectors [35] to initialize word embeddings. Meanwhile, the dimension of the position embedding, and part-of-speech (POS) embeddings is set to 30. We concatenate the word, position embeddings and POS embeddings. Then these embeddings are fed into the BiLSTM model, whose hidden size is set to 100. The size of dependency relation embeddings are set to 30, while constituency embeddings are set to 100. We use the Adam optimizer with the learning rate of 0.01. HGCN is trained for 100 epochs with batch size 32. The regularization coefficients λ is set to 1e-4 and coupling co-efficiency γ is set to 3e-2. As for HGCN + BERT, we use the fine-tuned BERT with officially released pre-trained BERT (BERT-base) parameters (See Table 1).

4.2. Main performances

The overall results compared with other baseline models on Restaurant, Laptop and Twitter datasets are shown in Table 2, from which several observations can be derived. First, the HGCN model consistently outperforms all compared baselines on different datasets. Second, compared with attention-based models, our model is significantly better since it focuses directly on the text regions that are semantically associated with the corresponding target. Therefore, it can eliminate noises introduced by the attention mechanism. Third, the performance of HGCN is also better than syntax-based models. It demonstrates that utilizing constituency and dependency information simultaneously enables the model to capture more structural syntactic information as a way to enhance the representations and HGCN can also avoid multi-hop between target and unrelated opinion words in dependency parsing. As for span-based models, our HGCN is able to model more complete text region compared to them and ensures semantic integrity, which allows our model to have a more robust semantic basis for identifying the sentiment polarity.

In addition, after we integrate our model with BERT, it obtains further improvement and reaches a new state-of-the-art. In a conclusion, these results can illustrate the effectiveness of our HGCN for capturing structural syntactic information in aspect-level sentiment analysis.

4.3. Effect of multiple aspects

In the datasets, there exists the scenario where a sentence may have multiple targets. Therefore, models may be misled by the opinion words of irrelevant targets and identify the sentiment polarity incorrectly. We conduct experiments on three typical models from three categories mentioned above.

As shown in Fig. 3 (There are no sentences with 5 targets in Rest15 dataset. Meanwhile, there is no multi-target sample in the Twitter dataset. Thus we do not need to analyze the Twitter dataset in this part.), syntax-based method (RGAT) outperforms attention-based method (AOA) in most cases. It indicates that establishing dependencies between words can partly avoid noisy from unrelated opinion words. As for span-based method (MCRF-SA), it can be seen that the performances tend to fluctuate when the number of targets is more than 3. This is due to the informal expressions and complicated structure of opinion words in sentences. Moreover, it is obvious that our HGCN model achieves better performances on these four datasets with different number of targets. This confirms the previously established viewpoint that

Table 2
Performance (%) comparison on baselines over 10 runs with random initialization. The best result with each dataset is in bold. For our models, the upper results represent the best performance and the lower are the average performance among 10 runs. Since MCRF-SA [12] requires additional opinion span data and they did not annotate the Twitter dataset, it is left blank here.

Model	Rest14		Lap14		Rest15		Rest16		Twitter	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
• Attention-based methods										
ATAE	77.01	62.93	70.09	63.75	76.60	51.14	84.36	52.52	69.19	65.52
AOA	79.97	70.42	72.62	67.52	78.17	57.02	87.50	66.21	70.03	67.34
MGAN	79.96	68.56	71.81	64.81	78.15	52.30	85.29	53.63	71.08	68.66
• Syntactic-based methods										
ASGCN	80.80	72.17	74.61	70.28	79.43	61.50	88.02	70.13	72.41	70.64
CDT	81.74	72.37	74.65	70.26	77.94	57.65	87.50	68.72	72.24	70.41
BiGCN	80.83	72.40	74.96	71.28	80.33	64.42	88.47	71.12	73.16	71.76
InterGCN	81.26	73.37	75.06	71.74	79.54	64.62	88.52	70.69	73.14	71.08
DGEDT	80.89	70.81	73.04	69.72	78.32	59.64	86.52	70.45	72.76	71.38
RGAT	81.50	72.20	73.72	70.49	78.78	59.74	88.21	71.20	73.45	71.56
DualGCN	82.85	75.61	76.66	72.78	80.16	63.26	88.19	70.17	74.15	72.56
• Span-based methods										
SA-LSTM	77.64	64.37	70.67	63.72	76.63	56.08	85.96	63.70	69.80	68.42
MCRF-SA	80.94	70.01	74.11	69.54	78.71	59.81	87.76	66.33	-	-
HGCN	**84.09**	**76.19**	**78.64**	**74.92**	**82.66**	**65.99**	**89.84**	**72.93**	**75.57**	**74.33**
(Ours)	**82.91**	**75.80**	**76.82**	**73.12**	**80.81**	**64.63**	**88.92**	**71.74**	**74.45**	**73.02**
• Models with BERT										
BERT	84.30	76.92	77.45	72.62	81.85	64.64	90.24	74.68	75.01	73.61
DGEDT	85.99	79.77	78.88	75.04	82.95	68.24	90.94	75.94	76.09	74.70
RGAT	85.91	79.44	79.10	74.78	83.15	68.32	91.39	76.05	75.61	74.37
DualGCN	86.29	80.00	79.51	75.66	83.78	70.11	91.43	78.55	76.42	75.28
HGCN	**87.41**	**82.14**	**81.49**	**77.32**	**85.61**	**71.65**	**93.02**	**80.34**	**77.46**	**76.49**
(Ours)	**86.45**	**80.60**	**79.59**	**76.24**	**83.91**	**68.68**	**91.72**	**78.71**	**76.52**	**75.37**

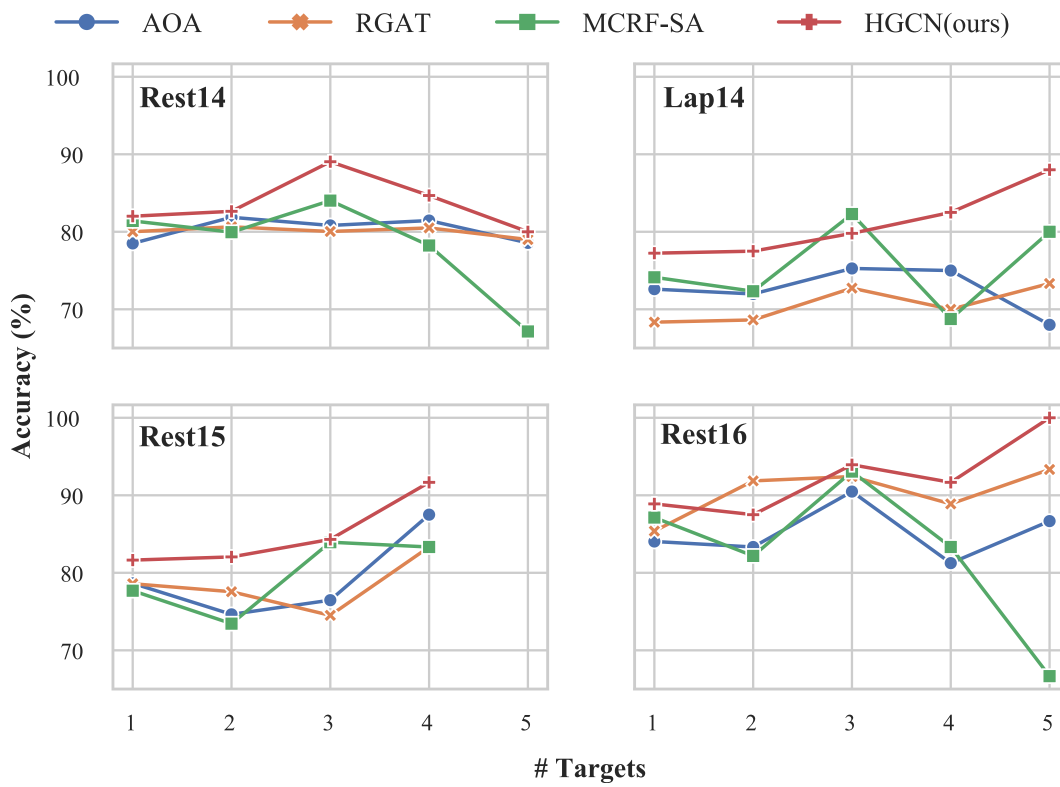


Fig. 3. Accuracy versus the number of targets (# Targets).

the *Scope* can efficiently delineate the text region of each target so as to alleviate the distraction problem caused by opinion words from irrelevant targets. In addition, the results also demonstrate that our HGCN model is able to learn more comprehensive syntactic information for reaching robust results.

4.4. Investigation on the mask matrix

In order to investigate the necessity of the mask matrix on assigning the information back to each word from constituents, we visualize the C-T attention score matrix of the CGCN w/o mask (i.e., the mask matrix M in Eq. 3 is dropped) and intact CGCN.

We take the sample shown in Fig. 2 with its constituency tree. As shown in Fig. 4, constituent is able to assign more information to the words that form it based on the attention score matrix constructed by C-T attention mechanism. However, the C-T attention may cause the problem that constituents are forced to inject information back to the words that are not composing them. Though only little information can be assigned to these words, it may introduce interference into CGCN. The mask matrix is utilized to constrain the assignment policy of C-T attention for tackling the problem. Compared with CGCN w/o mask and intact CGCN in Fig. 4 (M denotes the part that should be masked in C-T attention), the attention score matrix produced by intact CGCN is relatively sparse. For example, only "amazing" can be assigned the information from constituent "ADJP". Therefore, the mask matrix is a necessary component in our proposed CGCN.

4.5. Effect of model cascade

Having the ability of extracting the target term related region, the HGCN model can be regarded as a pre-module to enhance other aspect-level sentiment classification models by alleviating the distraction problem caused by misleading opinion words. Specifically, we harness the *Scope* selection part of the model to output the sentence associated with the target term from the CRF layer and then cascade to other models for the determination of sentiment polarity. More specifically, we keep the tokens inside the *Scope* and replace the ones outside with padding tag, making them invisible to the baseline models so that the *Scope* BIO-tags of output in CRF layer can be exploited to select the tokens that will be fed into the baseline models.

To evaluate performance improvements, we conduct experiments on five typical models from different categories. The results of accuracy enhancement are shown in Table 3. We can find that the attention-based models (ATAE-LSTM and MGAN) have the greatest performance improvement, which is due to the fact that our model drives the attention of each word to be more aggregated on the corresponding opinion words of the target term. Attention scores of these models are not scattered across the entire sentence, thus overcoming their dispersion problem.

There are also significant enhancements to syntax-based models (ASGCN and CDT). We infer that the *Scope* can assist the models to hard select reasonable dependencies to avoid problems caused by multiple-hops, which excludes interference from irrelevant opinion word of other target terms in sentiment polarity determination. In addition, we observe that our model can also be plugged in the BERT-equipped model for improving the performance. This indicates that the strategy of simplifying the sentence structure from multiple targets to a single target is also effective for pre-trained models.

4.6. Ablation

To examine the level of benefits on each component in our model, we conduct extensive ablation studies. The results are shown in Table 4. The first observation is that CGCN outperforms DGCN on Rest14 and Rest15, while it fails to act as well as DGCN

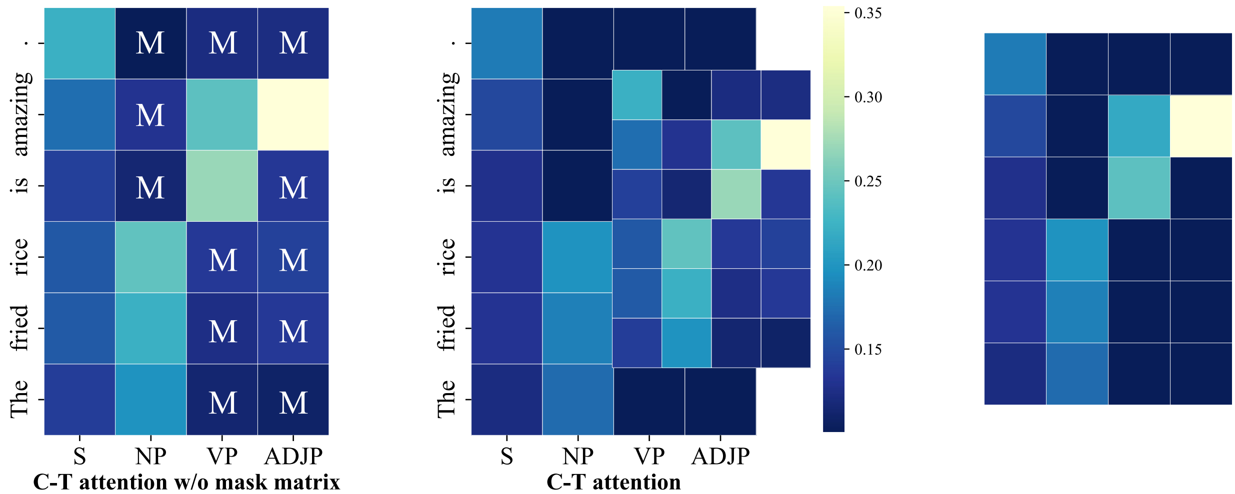


Fig. 4. The C-T attention score matrix of CGCN.

Table 3
Model accuracy improvement(%) among 10 runs when our model is used as the pre-module of other methods. The marker † refers to p-value < 0.01 and ‡ refers to p-value < 0.05 when comparing with original methods in paired t-test.

Model	Rest14	Lap14	Rest15	Rest16	Twitter
ATAE	0.76†	0.96‡	1.39†	1.08†	1.30†
MGAN	0.48‡	1.14‡	0.65‡	0.73†	0.65†
ASGCN	0.29†	1.07†	0.30†	0.29‡	0.47†
CDT	0.23‡	1.16‡	1.37†	2.59†	0.81†
BERT	0.27†	0.16‡	0.15†	0.34†	0.55†
Average	***0.41***	***0.90***	***0.77***	***1.01***	***0.76***

on the Lap14, Rest16 and Twitter. Moreover, it is beneficial to incorporate CGCN and DGCN for achieving better performances. This indicates that aggregating syntactic information from two complementary sides of syntactic relations can improve the robustness of the model. More importantly, HGCN w/o CRF means that we remove the subtask of *Scope* modeling so that HGCN cannot utilize external *Scope* data. However, our HGCN w/o CRF still outperforms the majority of the baselines shown in Table 2. It proves that our motivation of synthesizing constituency tree and dependency tree can learn more comprehensive syntactic information. In addition, the joint method (i.e., our HGCN) is better than the HGCN w/o CRF and achieves the best performance. This confirms that the joint scheme of *Scope* modeling and sentiment polarity determination is effective.

Meanwhile, we also explore the effectiveness of the constituency tree and dependency tree on *Scope* selection task. The *Overlap* metric is adopted here for evaluation, which calculates the coverage of the predicted sequence tags with the ground truth tags. As shown in Table 5, the results of comparison with LSTM + CRF (CGCN and DGCN are both detached from HGCN) show that these two components are both beneficial for the performance of *Scope* selection. However, CGCN performs better on the Rest14, Rest15, Rest16 and Twitter datasets, while it fails to perform as well as DGCN on the Lap14 dataset. This illustrates the ability of CGCN to encode structural information of *Scope*. Furthermore, it demonstrates that incorporating CGCN and DGCN can obtain richer structural information and achieve higher performance on *Scope* selection task.

5. Analysis

5.1. Case study

In this section, we investigate the behaviour of HGCN, CDT and RGAT on case examples. As we can see from the first example in Table 6, the CDT model is mistakenly identified since it is affected by an incorrect connection in the dependency tree where the edges connect "*service*" and "*great*" with 2-hops. Due to the gate mechanism, RGAT can filter out the noisy opinion word "*great*" by utilizing the edge labels, thus responding correctly. Our HGCN can exploit region information provided by the structural *Scope* to ensure the correctness of the determination. As for the second example, our model reasonably generates the *Scope* related to the target term "*location*" and identifies the sentiment polarity correctly. However, due to the edge between target term and "*treasure*" in dependency tree, RGAT and CDT are both misled. The third example shows that RGAT and CDT are all confused by another target "*blueberry*" in the sentence with its opinion word

Table 4
Experimental accuracy results (%) of ablation study. w/o CRF denotes model does not apply *Scope* selection task.

Model	Rest14		Lap14		Rest15		Rest16		Twitter	
	Acc	F1	Acc	F1	Acc	F1	Acc	F1	Acc	F1
• Single Module										
CGCN	83.21	75.34	76.30	72.77	81.38	63.00	88.59	69.79	74.15	72.62
DGCN	82.77	74.99	76.75	73.18	80.48	59.61	89.21	72.49	74.35	72.81
• w/o CRF										
CGCN	83.13	74.70	75.68	71.07	80.64	63.37	87.81	59.16	73.95	72.40
DGCN	82.32	74.57	76.60	73.26	80.25	63.29	88.43	**73.07**	73.67	72.10
HGCN	83.48	**76.54**	76.76	73.36	80.98	65.16	88.75	68.93	75.09	73.11
HGCN	**84.09**	76.19	**78.64**	**74.92**	**82.66**	**65.99**	**89.84**	72.93	**75.57**	**74.33**

Table 5
Experimental *Overlap* results of *Scope* selection task for ablation study.

Model	Rest14	Lap14	Rest15	Rest16	Twitter
LSTM + CRF	83.85	81.23	85.95	79.04	86.83
CGCN	86.00	85.08	89.13	83.32	91.07
DGCN	85.33	85.84	88.25	82.66	90.76
HGCN	**87.50**	**86.29**	**89.72**	**84.65**	**91.60**

Table 6
The words highlighted in red and blue denote the given targets and their *Scope* generated by HGCN. The notations P, N and O represent positive, negative and neutral sentiment, respectively.

Sentence	HGCN	CDT	RGAT
1.Great food but the service was dreadful!	P✓N✓	P✓P✗	P✓N✓
2.Unfortunately, unless you live in the neighborhood, it's not in a convenient location but is more like a hidden treasure.	N✓	P✗	P✗
3.My friend had a burger and i had these wonderful blueberry pancakes.	O✓P✓	P✗P✓	P✗P✓

"*wonderful*". By comparison, our model is able to determine sentiment polarity correctly since it delineates the exact context of each target so as to filter out noisy opinion words from irrelevant targets. These examples all illustrate the effectiveness of our *Scope*.

5.2. Error analysis

To analyze the limitation of our model, we trace back the error cases in the test sets, and identify three categories of reasons: short-sighted error, interruption error and representation error. The short-sighted error comes from the reason that the *Scope* fails to attend implicit transitive descriptions related to the target term. For example, the sentence "For someone who used to hate Indian food, Baluchi's has changed my mind." with its target term "*Indian food*", our *Scope* ignores the latter part of the sentence which contains the semantic transitions. Thus our model considers this to be an expression of negative sentiment polarity resulting in misidentification. As for the interruption error, some grammatical connection of the target term in the sentence (e.g., parenthetical text) can be interruptive, which may interfere with the constituency parsing and the modeling of *Scope*. Consequently, the *Scope* selection will suffer from missing essential information or ill-structured problems, like separate segments. Most of the other errors can be summarized as representation error. Though the model can correctly delineate the *Scope*, it is still unable to determine the sentiment polarity accurately, especially for the expression of negation and modality, which remains challenging for pre-trained language models and model structure of the upper layers.

5.3. Computational complexity analysis

Our HGCN model incorporates information from both constituency tree and dependency tree simultaneously. Intuitively, our model is more complex than other baseline models. Therefore, it is necessary to explore the differences of the models in terms of computational complexity. The number of parameters and floating point operations (FLOPs) are two metrics that are widely utilized to evaluate computational complexity of model. We calculate these metrics via THOP, which is a third-party library for PyTorch. However, some models (e.g. RGAT, InterGCN) use a hidden size of 300 in BiLSTM, which is much larger than other baselines. This can cause them to have more parameters and FLOPs. In order to compare the computational comlexity of all models in a fairer way, we scale common hyper-parameters of models (e.g., vocabulary size, the hidden size of BiLSTM and GCN module). Then we run these comparative models to obtain the results.

Table 7
Computational complexity analysis of models. **#Params** and **#FLOPs** represent the number of parameters and floating point operations for models.

Category	Model	#Params (k)	#FLOPs (M)
Attention	ATAE	762.3	4.29
	AOA	643.8	3.23
	MGAN	645.6	3.23
Syntax	ASGCN	322.2	5.82
	CDT	400.1	3.61
	BiGCN	1312.3	22.28
	InterGCN	322.2	15.19
	DGEDT	2886.2	57.75
	RGAT	1447.2	40.6
	DualGCN	511.0	8.18
Span	SA-LSTM	362.7	6.18
	MCRF-SA	177.9	16.56
Ours	HGCN	656.8	6.11

We illustrate the performances of HGCN and other baselines on these two metrics in Table 7. Three observations can be derived. First, the number of parameters and floating point operations in our HGCN are fewer than many of the models (e.g., BiGCN, DGEDT, RGAT) while the performances of our model outperforms these baselines. These results demonstrate the effectiveness of our HGCN.

Second, DGEDT is significantly more complex than other baselines due to the dual-transformer structure (DGEDT employs the architecture which consists of a multi-layer Transformer and a multi-layer BiGCN).

Besides, the attention-based models have fewer floating point operations compared to the syntax-based models. This is due to the fact that syntax-based models tend to adopt multi-layer graph neural networks, which involve more computational operations.

6. Conclusion and future work

In this paper, we provide a novel perspective that captures relations between targets and sentiments following the grammatical hierarchy of phrase-clause-sentence structure, aiming at filtering out noisy opinion words from irrelevant targets. Therefore, we introduce the definition of *Scope*, which is a structural text region related to the target term. To learn comprehensive syntactic relations for *Scope*, we propose a hybrid graph convolutional network to synthesize information from constituency tree and dependency tree. Therefore, our model can take advantage of the complementary features from constituency tree and dependency tree to better model the *Scope* and determine sentiment polarity. Experiments on extensive datasets demonstrate that our model outperforms the baselines and achieves the state-of-the-art performance. Furthermore, we also demonstrate the capability of our *Scope* that can be employed as a pre-module on other ALSA baseline models to improve their performances.

This study may be further improved in the following aspects. First, commonsense knowledge can be introduced into neural networks for better modeling the context-aware *Scope* in our future work. Second, a stronger strategy for capturing negation *Scope* is necessary [7,10] to address double negation expression. Third, with the development of prompt-based models, we can generate *Scope* via prompt methods, then infer the sentiment polarity of target from the *Scope*.

CRediT authorship contribution statement

Lvxiaowei Xu: Methodology, Software, Formal analysis, Writing – original draft, Writing – review & editing. **Xiaoxuan Pang:** Software, Formal analysis, Writing – review & editing. **Jianwang Wu:** Data curation, Investigation, Visualization, Writing – review & editing. **Ming Cai:** Conceptualization, Funding acquisition, Project administration, Supervision, Writing – original draft. **Jiawei Peng:** Data curation, Validation.

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.

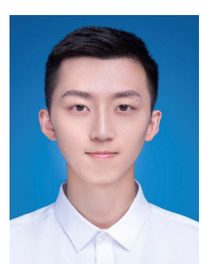
Acknowledgements

This research is supported by the Science and Technology Project of Zhejiang Province (2022C01044) and the National Natural Science Foundation of China (51775496).

References

[1] Y. Wang, M. Huang, X. Zhu, L. Zhao, Attention-based lstm for aspect-level sentiment classification, in: EMNLP, 2016, pp. 606–615.
[2] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: IJCAI, 2017.
[3] 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.
[4] F. Fan, Y. Feng, D. Zhao, Multi-grained attention network for aspect-level sentiment classification, in: EMNLP, 2018, pp. 3433–3442.
[5] C. Zhang, Q. Li, D. Song, Aspect-based sentiment classification with aspect-specific graph convolutional networks, in: EMNLP-IJCNLP, 2019, pp. 4568–4578.
[6] K. Sun, R. Zhang, S. Mensah, Y. Mao, X. Liu, Aspect-level sentiment analysis via convolution over dependency tree, in: EMNLP-IJCNLP, 2019, pp. 5679–5688.
[7] K. Wang, W. Shen, Y. Yang, X. Quan, R. Wang, Relational graph attention network for aspect-based sentiment analysis, in: ACL, 2020, pp. 3229–3238.
[8] M. Zhang, T. Qian, Convolution over hierarchical syntactic and lexical graphs for aspect level sentiment analysis, in: EMNLP, 2020, pp. 3540–3549.
[9] R. Li, h. Chen, F. Feng, Z. Ma, X. Wang, E. Hovy, Dual graph convolutional networks for aspect-based sentiment analysis, in: ACL, 2021, pp. 6319–6329.
[10] B. Wang, W. Lu, Learning latent opinions for aspect-level sentiment classification, in: AAAI, 2018.
[11] M. Hu, S. Zhao, H. Guo, R. Cheng, Z. Su, Learning to detect opinion snippet for aspect-based sentiment analysis, in: CoNLL, 2019, pp. 970–979.
[12] L. Xu, L. Bing, W. Lu, F. Huang, Aspect sentiment classification with aspect-specific opinion spans, in: EMNLP, 2020, pp. 3561–3567.
[13] T.N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, ICLR (2016).
[14] D. Marcheggiani, I. Titov, Graph convolutions over constituent trees for syntax-aware semantic role labeling, in: EMNLP, 2020, pp. 3915–3928.
[15] J.D. Lafferty, A. McCallum, F.C. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, in: ICML, 2001, pp. 282–289.
[16] S. Kiritchenko, X. Zhu, C. Cherry, S.M. Mohammad, Detecting aspects and sentiment in customer reviews, in: SemEval, 2014, pp. 437–442.
[17] Y. Kim, Convolutional neural networks for sentence classification, in: EMNLP, 2014, pp. 1746–1751.
[18] P. Liu, X. Qiu, X. Huang, Recurrent neural network for text classification with multi-task learning, in: IJCAI, 2016.
[19] A. Joulin, É. Grave, P. Bojanowski, T. Mikolov, Bag of tricks for efficient text classification, in: ACL, 2017, pp. 427–431.
[20] W. Xue, T. Li, Aspect based sentiment analysis with gated convolutional networks, ACL (2018).
[21] Q. Xu, L. Zhu, T. Dai, C. Yan, Aspect-based sentiment classification with multi-attention network, Neurocomputing 388 (2020) 135–143.
[22] M.H. Phan, P.O. Ogunbona, Modelling context and syntactical features for aspect-based sentiment analysis, in: ACL, 2020, pp. 3211–3220.
[23] C. Chen, Z. Teng, Y. Zhang, Inducing target-specific latent structures for aspect sentiment classification, in: EMNLP, 2020, pp. 5596–5607.
[24] B. Liang, R. Yin, L. Gui, J. Du, R. Xu, Jointly learning aspect-focused and inter-aspect relations with graph convolutional networks for aspect sentiment analysis, in: COLING, 2020, pp. 150–161.
[25] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, Bert: Pre-training of deep bidirectional transformers for language understanding, in: ACL, 2019, pp. 4171–4186.
[26] X. Glorot, A. Bordes, Y. Bengio, Deep sparse rectifier neural networks, in: PMLR, 2011, pp. 315–323.
[27] 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.
[28] 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.
[29] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, in: SemEval, 2014, pp. 27–35.
[30] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, I. Androutsopoulos, Semeval-2015 task 12: Aspect based sentiment analysis, in: SemEval, 2015, pp. 486–495.
[31] 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: International workshop on semantic evaluation, 2016, pp. 19–30.
[32] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: ACL, 2014, pp. 49–54.
[33] H. Tang, D. Ji, C. Li, Q. Zhou, Dependency graph enhanced dual-transformer structure for aspect-based sentiment classification, in: ACL, 2020, pp. 6578–6588.
[34] C.D. Manning, M. Surdeanu, J. Bauer, J.R. Finkel, S. Bethard, D. McClosky, The stanford corenlp natural language processing toolkit, in: ACL, 2014, pp. 55–60.
[35] J. Pennington, R. Socher, C.D. Manning, Glove: Global vectors for word representation, in: EMNLP, 2014, pp. 1532–1543.

Lvxiaowei Xu received the B.E. degree from Hebei University, China, in 2020, where he is currently pursuing the Ph.D. degree with the College of Computer Science and Technology in Zhejiang University. he was with the AI-Empowered System Software Laboratory, Zhejiang University, where he is involved in sentiment analysis and pre-trained language model.



Xiaoxuan Pang received the B.S. degree in civil engineering from Hohai University, and M.S. degree in computer science from Zhejiang University, China, in 2019 and 2022 respectively. During Master's degree, he was with the AI-Empowered System Software Laboratory, Zhejiang University, where he was involved in natural language processing. He is currently working as a software engineer in Alibaba.

Jianwang Wu received the B.S. degree in Mathematics major from Zhengzhou University of Light Industry, Zhengzhou, China in 2019. He is currently pursuing the M.S. degree in Software Engineering major from Zhejiang University, Zhejiang, China. His research interests in NLP include sentiment analysis, identity of named entity and so on.

Ming Cai received his B.S. degree in electrical engineering, and his M.S. and Ph.D. degrees both in computer science from Zhejiang University, China, in 1996, 1999, and 2002 respectively. He is currently working as an associate professor with the College of Computer Science and Technology, Zhejiang University. Dr. Cai has published more than 80 research articles in international conferences and refereed journals. His research interests include Internet of Things, natural language processing and embedded system. He is currently leading some research project supported by National Natural Science Foundation of China (NSFC).

Jiawei Peng received the B.S. degree in computer science from Zhejiang University, China, in 2021. He is currently pursuing the master's degree with the College of Computer Science and Technology in Zhejiang University. He is with the AI-Empowered System Software Laboratory, Zhejiang University. His research interests include deep learning and natural language processing.