Modeling Sentiment Dependencies with Graph Convolutional Networks for Aspect-level Sentiment Classification

Pinlong Zhao[a], Linlin Hou[b], Ou Wu[a]

[a]Center for Applied Mathematics, Tianjin University
[b]Center for Combinatorics, Nankai University

Abstract

Aspect-level sentiment classification aims to distinguish the sentiment polarities over one or more aspect terms in a sentence. Existing approaches mostly model different aspects in one sentence independently, which ignore the sentiment dependencies between different aspects. However, we find such dependency information between different aspects can bring additional valuable information. In this paper, we propose a novel aspect-level sentiment classification model based on graph convolutional networks (GCN) which can effectively capture the sentiment dependencies between multi-aspects in one sentence. Our model firstly introduces bidirectional attention mechanism with position encoding to model aspect-specific representations between each aspect and its context words, then employs GCN over the attention mechanism to capture the sentiment dependencies between different aspects in one sentence. We evaluate the proposed approach on the SemEval 2014 datasets. Experiments show that our model outperforms the state-of-the-art methods. We also conduct experiments to evaluate the effectiveness of GCN module, which indicates that the dependencies between different aspects is highly helpful in aspect-level sentiment classification.

Keywords: Sentiment classification, Aspect-level, Sentiment dependencies, Graph convolutional networks

1. Introduction

Aspect-level sentiment classification [1, 2] is a fundamental natural language processing task that gets lots of attention in recent years. It is a fine-grained task in sentiment analysis, which aims to infer the sentiment polarities of aspects in their context. For example, in the sentence *"The price is reasonable although the service is poor"*, the sentiment polarities for the two aspect terms, *"price"* and *"service"*, are positive and negative respectively. An aspect term (or simply aspect) is usually an entity or an entity aspect.

Aspect-level sentiment classification is much more complicated than sentence-level sentiment classification, because identifying the parts of sentence describing the corresponding aspects is difficult. Traditional approaches [3, 4] mainly focus on statistical methods to design a set of handcrafted features to train a classifier (e.g., Support Vector Machine). However, such kind of feature-based work is labor-intensive. In recent years, neural network models [5, 6] are of growing interest for their capacity to automatically generate useful low dimensional representations from aspects and their contexts, and achieve great accuracy on the aspect-level sentiment classification without careful engineering of features. Especially, by the ability to effectively identify which words in the sentence are more important on a given aspect, attention mechanisms [7, 8] implemented by neural networks are widely used in aspect-level sentiment classification [9–14]. Chen et al. [10] model a multiple attention
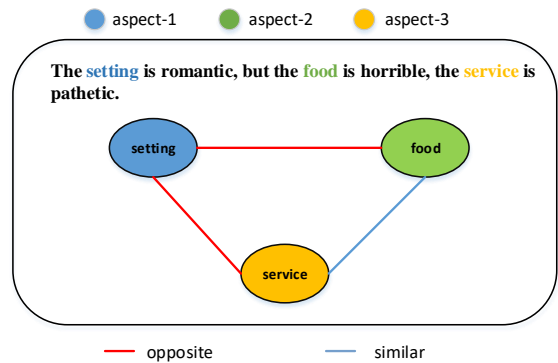

Fig. 1. An example to illustrate the usefulness of the sentiment dependencies between multiple aspects. The dependencies can be inferred by some knowledge in the sentence, e.g., conjunction. The evidence of the usefulness of the sentiment dependencies is that we can easily guess the true sentiment of *"food"* even if we mask the word *"horrible"*.

mechanism with a gated recurrent unit network to capture the relevance between each context word and the aspect. Ma et al. [11] design a model which learns the representations of the aspect and context interactively with two attention mechanisms. Song et al. [13] propose an attentional encoder network, which employ multi-head attention for the modeling between context and aspect. These attention-based models have proven to be successful and effective in learning aspect-specific representations.

Despite these advances, the studies above still remain problems. They all build models with each aspect individually ignoring the sentiment dependencies information between mul-

---

tiple aspects, which will lose some additional information. For example, as we can see from the example given in Fig. 1, the sentiment polarity of the first aspect *"setting"* is positive. From the conjunction *"but"*, we are easy to know that the second aspect *"food"* has opposite sentiment polarity with *"setting"*. By this sentiment dependency relation, we can guess the polarity of aspect *"food"* is negative. Similarly, from the second comma, we conjecture that the sentiment polarity of the last aspect *"service"* is likely the same as *"food"*. Therefore, the sentiment dependencies are helpful to infer the sentiment polarities of aspects in one sentence.

In this paper, we propose a novel method to model **S**entiment **D**ependencies with **G**raph **C**onvolutional **N**etworks (SDGCN) for aspect-level sentiment classification. GCN is a simple and effective convolutional neural network operating on graphs, which can catch inter-dependent information from rich relational data [15]. For every node in graph, GCN encodes relevant information about its neighborhoods as a new feature representation vector. In our case, an aspect is treated as a node, and an edge represents the sentiment dependency relation of two nodes. Our model learns the sentiment dependencies of aspects via this graph structure. As far as we know, our work is the first to consider the sentiment dependencies between aspects in one sentence for aspect-level sentiment classification task. Furthermore, in order to capture the aspect-specific representations, our model applies bidirectional attention mechanism with position encoding before GCN. We evaluate the proposed approach on the SemEval 2014 datasets. Experiments show that our model outperforms the state-of-the-art methods.[1] The main contributions of this paper are presented as follows:

- To the best of our knowledge, this is the first study to consider the sentiment dependencies between aspects in one sentence for aspect-level sentiment classification.

- We design bidirectional attention mechanism with position encoding to capture the aspect-specific representations.

- We propose a novel multi-aspects sentiment classification framework, which employs GCN to effectively capture the sentiment dependencies between different aspects in one sentence.

- We evaluate our method on the SemEval 2014 datasets. And experiments show that our model achieves superior performance over the state-of-the-art approaches.

2. Related work

In this section, we will review related works on aspect-level sentiment classification and graph convolutional network briefly.

2.1. Aspect-level sentiment classification

Sentiment analysis, also known as opinion mining [16, 17], is an important research topic in Natural Language Processing (NLP). Aspect-level sentiment classification is a fine-grained task in sentiment analysis. In aspect-level sentiment classification , early works mainly focus on extracting a set of features like bag-of-words features and sentiment lexicons features to train a sentiment classifier [18]. These methods including rule-based methods [19] and statistic-based methods [20] rely on feature-engineering which are labor intensive. In recent years, deep neural network methods are getting more and more attention as they can generate the dense vectors of sentences without handcrafted features [21, 22]. And the vectors are low-dimensional word representations with rich semantic information remained. Moreover, using the attention mechanism can enhance the sentence representation for concentrating on the key part of a sentence given an aspect [23–25]. Wang et al. [9] propose ATAE-LSTM that combines LSTM and attention mechanism. The model makes embeddings of aspects to participate in computing attention weights. RAM is proposed by Chen et al. [10] which adopts multiple-attention mechanism on the memory built with bidirectional LSTM. Ma et al. [11] design a model with the bidirectional attention mechanism, which interactively learns the attention weights on context and aspect words respectively. Song et al. [13] propose an attentional encoder network, which eschews recurrence and apply multi-head attention for the modeling between context and aspect. However, these attention works model each aspect separately in one sentence, which may loss some sentiment dependency information on multiple aspects case.

2.2. Graph convolutional network

Graph convolutional network [26] is effective at dealing with graph data which contains rich relation information. Many works dedicate to extending GCN for image tasks [27–30]. Chen et al. [31] build the model via GCN for multi-label image recognition, which propagates information between multiple labels and consequently learns inter-dependent classifiers for each of image labels. GCN has also received growing attention in NLP recently such as semantic role labeling [32], machine translation [33] and relation classification [34]. Some works explore graph neural networks for text classification [35, 36]. They view a document, a sentence or a word as a graph node and rely on the relation of nodes to construct the graph. The studies above show that GCN can effectively capture relation between nodes. Inspired by these, we adopt GCN to get the sentiment dependencies between multi-aspects.

3. Methodology

Aspect-level sentiment classification can be formulated as follows. Given an input context consists of N words $W^c = \{w_1^c, w_2^c, \ldots, w_N^c\}$, and K aspect teams $W^a = \{W^{a_1}, W^{a_2}, \ldots, W^{a_K}\}$. Each aspect $W^{a_i} = \{w_1^{a_i}, w_2^{a_i}, \ldots, w_{M_i}^{a_i}\}$ is a subsequence of sentence W^c, which contains $M_i \in [1, N)$
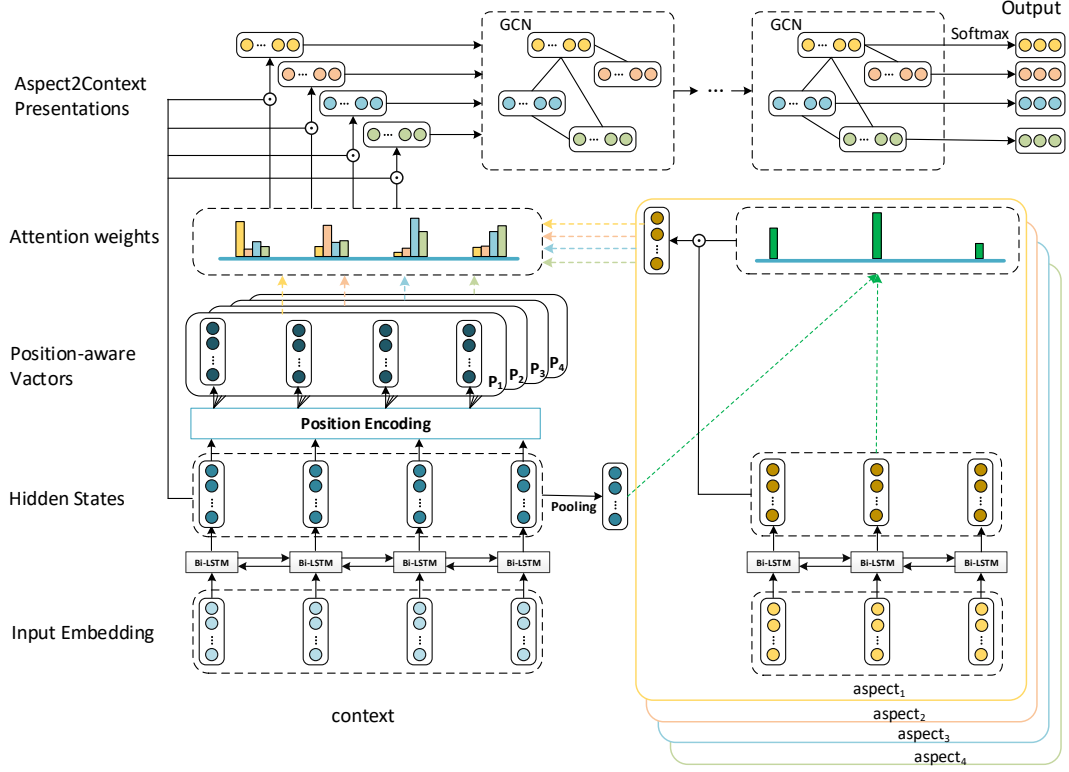

Fig. 2. An example to illustrate the usefulness of the sentiment dependencies between multiple aspects. The dependencies can be inferred by some knowledge in the sentence, e.g., conjunction. The evidence of the usefulness of the sentiment dependencies is that we can easily guess the true sentiment of food even if we mask the word horrible.

words. It is required to construct a sentiment classifier that predicts the sentiment polarities of the multiple aspect teams.

We present the overall architecture of the proposed SDGCN in Fig. 2. It consists of the input embedding layer, the Bi-LSTM, the position encoding, the bidirectional attention mechanism, the GCN and the output layer. Next, we introduce all components sequentially from input to output.

3.1. Input embedding layer

Input embedding layer maps each word to a high dimensional vector space. We employ the pretrained embedding matrix GloVe [37] and pretrained model BERT [38] to obtain the fixed word embedding of each word. Then each word will be represented by an embedding vector $e_t \in \mathbb{R}^{d_{emb} \times 1}$, where d_{emb} is the dimension of word vectors. After embedding layer, the context embedding is denoted as a matrix $E^c \in \mathbb{R}^{d_{emb} \times N}$, and the i-th aspect embedding is denoted as a matrix $E^{a_i} \in \mathbb{R}^{d_{emb} \times M_i}$.

3.2. Bidirectional Long Short-Term Memory (Bi-LSTM)

We employ Bi-LSTM on top of the embedding layer to capture the contextual information for each word. After feeding word embedding to Bi-LSTM, the forward hidden state $\overrightarrow{h_t} \in \mathbb{R}^{d_{hid} \times 1}$ and the backward hidden state $\overleftarrow{h_t} \in \mathbb{R}^{d_{hid} \times 1}$ are obtained, where d_{hid} is the number of hidden units. We concatenate both the forward and the backward hidden state to form the final representation:

$$h_t = [\overrightarrow{h_t}, \overleftarrow{h_t}] \in \mathbb{R}^{2d_{hid} \times 1} \tag{1}$$

In our model, we employ two Bi-LSTM separately to get the sentence contextual hidden output $H^c = [h_1^c, h_2^c, \ldots, h_N^c] \in \mathbb{R}^{2d_{hid} \times N}$ and each aspect contextual hidden output $H_i^a = [h_1^{a_i}, h_2^{a_i}, \ldots, h_{M_i}^{a_i}] \in \mathbb{R}^{2d_{hid} \times M_i}$. Note that, the Bi-LSTM for each different aspect shares the parameters.

3.3. Position encoding

Based on the intuition that the polarity of a given aspect is easier to be influenced by the context words with closer distance to the aspect, we introduce position encoding to simulate this normal rules in natural language. Formally, given an aspect W^{a_i} that is one of the K aspects, where $i \in [1, K]$ is the index of aspects, the relative distance $d_t^{a_i}$ between the t-th word and the i-th aspect is defined as follows:

$$d_t^{a_i} = \begin{cases} 1, & dis = 0 \\ 1 - \frac{dis}{N}, & 1 \leq dis \leq s \\ 0, & dis > s \end{cases} \tag{2}$$

where dis is the distance between a context word and the aspect (here we treat an aspect as a single unit, and $d = 0$ means that the context word is also the aspect word), s is a pre-specified constant, and N is the length of the context. Finally, we can obtain the position-aware representation with position information:

$$\begin{aligned} p_t^{a_i} &= d_t^{a_i} h_t^c \\ P^{a_i} &= P_i = [p_1^{a_i}, p_2^{a_i}, \ldots, p_N^{a_i}] \end{aligned} \tag{3}$$

3

3.4. Bidirectional attention mechanism

In order to capture the interactive information between the context and the aspect, we employ a bidirectional attention mechanism in our model. This mechanism consists of two modules: *context to aspect attention* module and *aspect to context attention* module. Firstly, the former module is used to get new representations of aspects based on the context. Secondly, based on the new representations, the later module is employed to obtain the aspect-specific context representations which will be fed into the downstream GCN.

3.4.1. Context to aspect attention

Context to aspect attention learns to assign attention weights to the aspect words according to a query vector, where the query vector is $\overline{h^c} \in \mathbb{R}^{2d_{hid} \times 1}$ which is obtained by average pooling operation over the context hidden output H^c. For each hidden word vector $h_t^{a_i} \in \mathbb{R}^{2d_{hid} \times 1}$ in one aspect, the attention weight $\beta_t^{a_i}$ is computed as follows:

$$f_{ca}(\overline{h^c}, h_t^{a_i}) = \overline{h^c}^T \cdot W_{ca} \cdot h_t^{a_i} \tag{4}$$

$$\beta_t^{a_i} = \frac{exp(f_{ca}(\overline{h^c}, h_t^{a_i}))}{\sum_{t=1}^{M_i} exp(f_{ca}(\overline{h^c}, h_t^{a_i}))} \tag{5}$$

where $W_{ca} \in \mathbb{R}^{2d_{hid} \times 2d_{hid}}$ is the attention weight matrix.

After computing the word attention weights, we can get the weighted combination of the aspect hidden representation as a new aspect representation:

$$m^{a_i} = \sum_{t=1}^{M_i} \beta_t^{a_i} \cdot h_t^{a_i} \tag{6}$$

3.4.2. Aspect to context attention

Aspect to context attention learns to capture the aspect-specific context representation, which is similar to *context to aspect attention*. Specifically, the attention scores is calculated by the new aspect representation m^{a_i} and the position-aware representation $p_t^{a_i}$. The process can be formulated as follows:

$$f_{ac}(m^{a_i}, p_t^{a_i}) = m^{a_i T} \cdot W_{ac} \cdot p_t^{a_i} \tag{7}$$

$$\gamma_t^{a_i} = \frac{exp(f_{ac}(m^{a_i}, p_t^{a_i}))}{\sum_{t=1}^{N} exp(f_{ac}(m^{a_i}, p_t^{a_i}))} \tag{8}$$

$$x^{a_i} = x_i = \sum_{t=1}^{N} \gamma_t^{a_i} \cdot h_t^c \tag{9}$$

where $W_{ac} \in \mathbb{R}^{2d_{hid} \times 2d_{hid}}$ is the attention weight matrix. By now, we get the aspect-specific representations $X = [x_1, x_2, \ldots, x_K]$ between each aspect and its context words, where K is the number of aspects in the context.
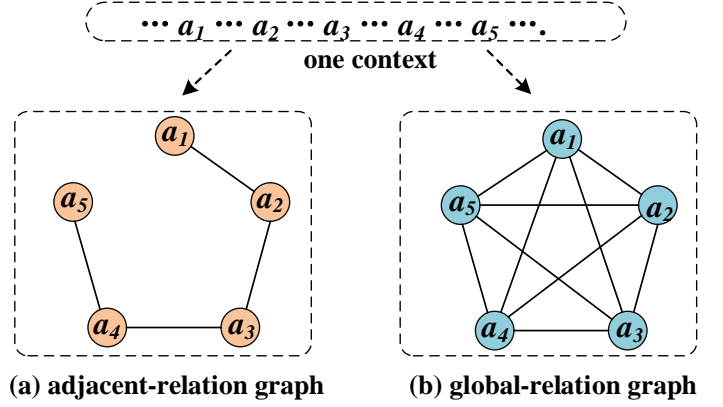

Fig. 3. Illustration of our proposed sentiment graphs. a_1, a_2, a_3, a_4 and a_5 denote five aspects in one context.

3.5. Graph convolutional network

GCN is widely used to deal with data which contains rich relationships and interdependency between objects, because GCN can effectively capture the dependence of graphs via message passing between the nodes of graphs. We also employ a graph to capture the sentiment dependencies between aspects. The final output of each GCN node is designed to be the classifier of the corresponding aspect in our task. Moreover, there are no explicit edges in our task. Thus, we need to define the edges from scratch.

3.5.1. Sentiment graph

We construct a graph, named sentiment graph, to capture the sentiment dependencies between multi-aspects in one sentence, where each node is regarded as an aspect and each edge is treated as the sentiment dependency relation. As shown in Fig. 3, we define two kinds of undirected sentiment graphs:

- adjacent-relation graph: An aspect is only connected to its nearby aspects.

- global-relation graph: An aspect is connected to all other aspects.

If two nodes are connected by an edge, it means that the two nodes are neighboring to each other. Formally, given a node v, we use $N(v)$ to denote all neighbors of v. $u \in N(v)$ means that u and v are connected with an edge.

3.5.2. Sentiment graph based GCN

GCN encodes relevant information about its neighborhood as a new representation vector, where each node in the graph indicates a representation of aspect. In addition, as Kipf et al. [15] do, we assume all nodes contain self-loops. Then, the new node representation is computed as follows:

$$x_v^1 = relu(\sum_{u \in N(v)} W_{cross} x_u + b_{cross}) + ReLU(W_{self} x_v + b_{self}) \tag{10}$$

where $W_{cross}, W_{self} \in \mathbb{R}^{d_m \times d_n}$, $b_{cross}, b_{self} \in \mathbb{R}^{d_m \times 1}$, x_u is the u-th aspect-specific representation (see Eq.(9)), and *ReLU* is the
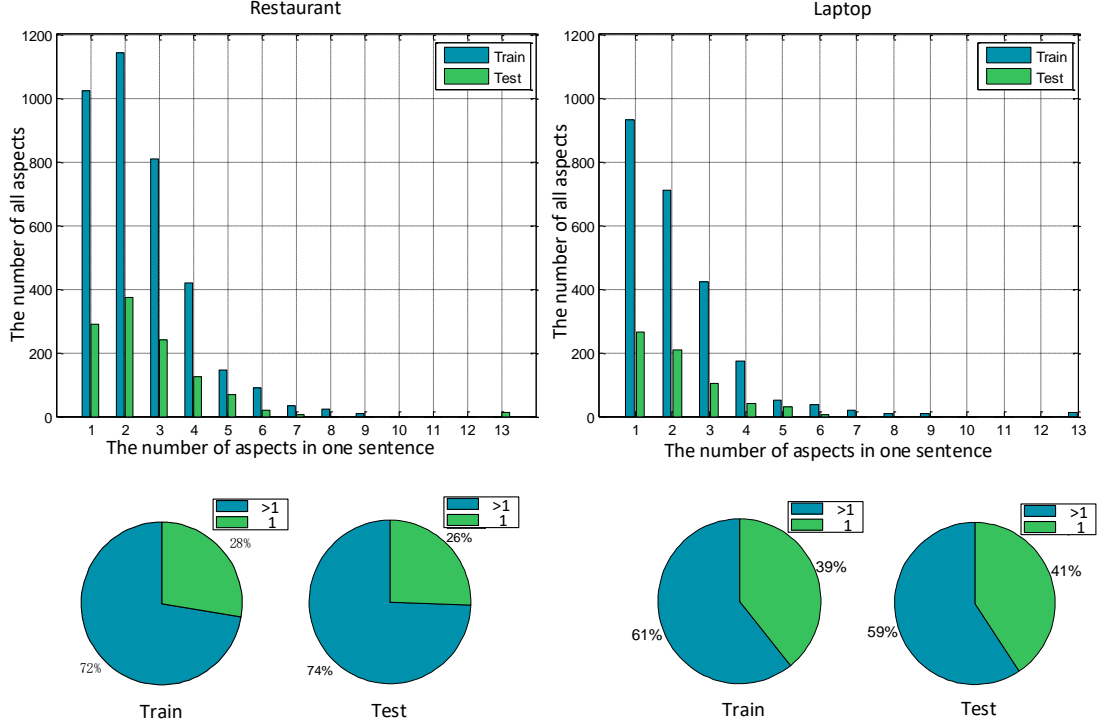
Fig. 4. Statistics of the number of aspects in one sentence on SemEval 2014 data set.

rectifier linear unit activation function. In this work, we use $d_m = d_n = 2h_{hid}$.

By stacking multiple GCN layers, the final hidden representation of each node can receive messages from a further neighborhood. Each GCN layer takes the node representations from previous layer as inputs and outputs new node representations:

$$x_v^{l+1} = relu(\sum_{u \in N(v)} W_{cross}^l x_u^l + b_{cross}^l) + relu(W_{self}^l x_v^l + b_{self}^l)$$

(11)

where l denotes the layer number and $1 \le l \le L - 1$.

3.6. Output layer

The final output of each GCN node x_i^L is treated as a classifier of the i-th aspect. At last, we use a fully-connected layer to map x_i^L into the aspect space of C classes:

$$z_i = W_z x_i^L + b_z$$

(12)

where $W_z \in \mathbb{R}^{C \times 2d_{hid}}$ is the weight matrix, and $b_z \in \mathbb{R}^{2d_{hid} \times C}$ is the bias. The predicted probability of the i-th aspect with sentiment polarity $j \in [1, C]$ is computed by:

$$y_{ij}' = \frac{exp(z_{ij})}{\sum_{k=1}^C exp(z_{ik})}$$

(13)

3.7. Model training

Our model is trained by minimizing the cross entropy with L2-regularization term. For a given sentence, the loss function

Table 1: The details of the experimental data sets.

Data		Positive	Negative	Neutral
Restaurant	Train	2164	807	637
	Test	728	196	196
Laptop	Train	994	870	464
	Test	341	128	169

is defined as:

$$loss = \sum_{i=1}^K \sum_{j=1}^C y_{ij} log(y_{ij}') + \lambda \parallel \theta \parallel^2$$

(14)

where y_{ij} is a one-hot labels of the i-th aspect for the j-th class, λ is the coefficient for L2-regularization, θ is the parameters that need to be regularized. Furthermore, we adopt the dropout strategy during training step to avoid over-fitting.

4. Experiments

4.1. Data sets and experimental settings

To demonstrate the effectiveness of our proposed method, as most previous works [9, 11, 13, 25], we conduct experiments on two datasets from SemEval 2014 Task4[2] [39], which contains the reviews in laptop and restaurant. The details of the SemEval 2014 datasets are shown in Table 1. Each dataset consists of

---

[2]The detailed introduction of this task can be found at `http://alt.qcri.org/semeval2014/task4`.

Table 2: Comparisons with baseline models on the Restaurant dataset and Laptop dataset. The results of baseline models are retrieved from published papers. The best results in GloVe-based models and BERT-based models are all in bold separately. -A means that the model is based on adjacent-relation graph, and -G means the model is based on global-relation graph.

Word Embedding	Models	Restaurant		Laptop	
		Acc	Macro-F1	Acc	Macro-F1
GloVe	TD-LSTM	75.63	-	68.13	-
	ATAE-LSTM	77.20	-	68.70	-
	MenNet	78.16	65.83	70.33	64.09
	IAN	78.60	-	72.10	-
	RAN	80.23	70.80	74.49	**71.35**
	PBAN	81.16	-	74.12	-
	TSN	80.1	-	73.1	-
	AEN	80.98	72.14	73.51	69.04
	SDGCN-A w/o p	81.61	72.22	73.20	68.54
	SDGCN-G w/o p	81.61	72.93	73.67	68.70
	SDGCN-A	82.14	73.47	75.39	70.04
	SDGCN-G	**82.95**	**75.79**	**75.55**	**71.35**
BERT	AEN-BERT	83.12	73.76	79.93	76.31
	SDGCN-BERT	**83.57**	**76.47**	**81.35**	**78.34**

train and test set. Each review (one sentence) contains one or more aspects and their corresponding sentiment polarities, i.e., positive, neutral and negative. To be specific, the number in table means the number of aspects in each sentiment category. To demonstrate the necessity of considering the sentiment dependencies between the aspects, we further calculate the number of aspects in each sentence, which is presented in Fig. 4. From the histogram in Fig. 4, we can see that each sentence contains one to thirteen aspects. The number of aspects in most reviews is 1 to 4. The pie chart shows the proportion of only one aspect and more than one aspect in one sentence. It can be seen that more than half of the aspects do not appear alone in a review. According to these statistics, we can conclude that it is common to have multi-aspects within one sentence. Our model mainly aims to model the sentiment dependencies between different aspects in one sentence.

In our implementation, we respectively use the GloVe[3] [37] word vector and the pre-trained language model word representation BERT[4] [38] to initialize the word embeddings. The dimension of each word vector is 300 for GloVe and 768 for BERT. The number of LSTM hidden units is set to 300, and the output dimension of GCN layer is set to 600. The weight matrix of last fully connect layer is randomly initialized by a normal distribution $N(0,1)$. Besides the last fully connect layer, all the weight matrices are randomly initialized by a uniform distribution $U(-0.01, 0.01)$. In addition, we add $L2$-regularization to the last fully connect layer with a weight of 0.01. During training, we set dropout to 0.5, the batch size is set to 32 and the optimizer is Adam Optimizer with a learning rate of 0.001. We implement our proposed model using Tensorflow[5]. To evaluate performance of the model, we employ Accuracy and Macro-F1

metrics. The Macro-F1 metric is more appropriate when the data set is not balanced.

4.2. Comparative methods

To comprehensively evaluate the performance of proposed SDGAN, we compare our model with the following models.

- **TD-LSTM** [6] constructs aspect-specific representation by the left context with aspect and the right context with aspect, then employs two LSTMs to model them respectively. The last hidden states of the two LSTMs are finally concatenated for predicting the sentiment polarity of the aspect.

- **ATAE-LSTM** [9] first attaches the aspect embedding to each word embedding to capture aspect-dependent information, and then employs attention mechanism to get the sentence representation for final classification.

- **MemNet** [40] uses a deep memory network on the context word embeddings for sentence representation to capture the relevance between each context word and the aspect. Finally, the output of the last attention layer is used to infer the polarity of the aspect.

- **IAN** [11] generates the representations for aspect terms and contexts with two attention-based LSTM network separately. Then the context representation and the aspect representation are concatenated for predicting the sentiment polarity of the aspect.

- **RAM** [10] employs a gated recurrent unit network to model a multiple attention mechanism, and captures the relevance between each context word and the aspect. Then the output of the gated recurrent unit network is obtained for final classification.

6

- **PBAN** [41] appends the position embedding into each word embedding. It then introduces a position-aware bidirectional attention network (PBAN) based on Bi-GRU to enhance the mutual relation between the aspect term and its corresponding sentence.

- **TSN** [25] is a two-stage framework for aspect-level sentiment analysis. The first stage, it uses a position attention to capture the aspect-dependent representation. The second stage, it introduces penalization term to enhance the difference of the attention weights towards different aspects in one sentence.

- **AEN** [13] mainly consists of an embedding layer, an attentional encoder layer, an aspect-specific attention layer, and an output layer. In order to eschew the recurrence, it employs attention-based encoders for the modeling between the aspect and its corresponding context.

- **AEN-BERT** [13] is AEN with BERT embedding.

4.3. Overall results

Table 2 shows the experimental results of competing models. In order to remove the influence with different word representations and directly compare the performance of different models, we compare GloVe-based models and BERT-based models separately. Our proposed model achieves the best performance on both GloVe-based models and BERT-based models, which demonstrates the effectiveness of our proposed model. In particularly, SDGCN-BERT obtains new state-of-the-art results.

Among all the GloVe-based methods, the TD-LSTM approach performs worst because it takes the aspect information into consideration in a very coarse way. ATAE-LSTM, MenNet and IAN are basic attention-based models. After taking the importance of the aspect into account with attention mechanism, they achieve a stable improvement comparing to the TD-LSTM. RAM achieves a better performance than other basic attention-based models, because it combines multiple attentions with a recurrent neural network to capture aspect-specific representations. PBAN achieves a similar performance as RAM by employing a position embedding. To be specific, PBAN is better than RAM on Restaurant dataset, but worse than RAN on Laptop dataset. Compared with RAM and PBAN, the overall performance of TSN is not perform well on both Restaurant dataset and Laptop dataset, which might because the framework of TSN is too simple to model the representations of context and aspect effectively. AEN is slightly better than TSN, but still worse than RAM and PBAN. It indicates that the discard of the recurrent neural networks can reduce the size of model while lead to the loss of performance.

Comparing the results of SDGCN-A w/o position and SDGCN-G w/o position, SDGCN-A and SDGCN-G, respectively, we observe that the GCN built with global-relation is slightly higher than built with adjacent-relation in both accuracy and Macro-F1 measure. This may indicate that the adjacent relationship is not sufficient to capture the interactive information among multiple aspects due to the neglect of the long-distance relation of aspects. Moreover, the two models
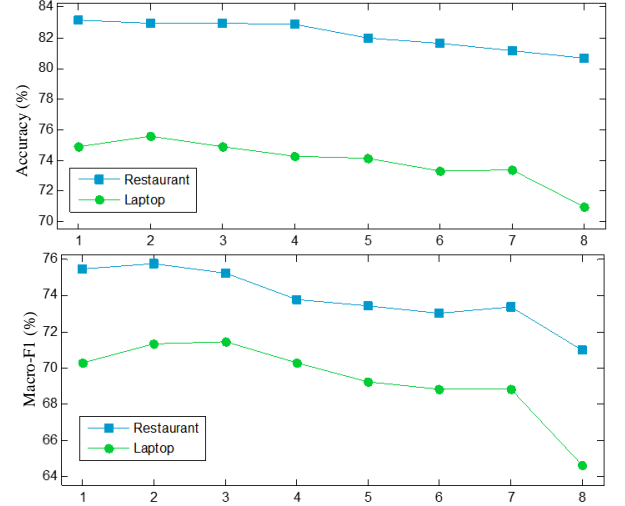

Fig. 5. Comparisons with different depths of GCN in our model.

(SDGCN-A and SDGCN-G) with position information gain a significant improvement compared to the two models without position information. It shows that the position encoding module is crucial for good performance.

Benefits from the power of pre-trained BERT, BERT-based models have shown huge superiority over GloVe-based models. Furthermore, compared with AEN-BERT, on the Restaurant dataset, SDGCN-BERT achieves absolute increases of 1.09% and 1.86% in accuracy and Macro-F1 measure respectively, and gains absolute increases of 1.42% and 2.03% in accuracy and Macro-F1 measure respectively on the Laptop dataset. The increments prove the effectiveness of our proposed SDGCN.

Table 3: The effect of GCN.

Models	Restaurant		Laptop	
	Acc	Macro-F1	Acc	Macro-F1
Att	81.43	72.40	72.12	68.67
Att+GCN	82.77	74.33	74.61	70.33
BiAtt	81.61	73.49	73.51	69.73
BiAtt+GCN (SDGCN)	82.95	75.79	75.55	71.35

4.4. The effect of GCN module

In this section, we design a series of models to further verify the effectiveness of GCN module. These models are:

- **BiAtt+GCN** is just another name of our proposed SDGCN model.

- **BiAtt** is based on BiAtt -GCN, where we remove the GCN module. Therefore, it predicts the sentiments of different aspects in one sentence independently.

- **Att+GCN** is a simplified version of BiAtt+GCN. The only difference between Att+GCN and BiAtt+GCN is that Att+GCN does not have *context to aspect attention*.

Fig. 6. Illustration of attention weights obtained by model with GCN and without GCN respectively. (a) and (b) are two examples from the Laptop dataset. (a) Aspects: keyboard, screen; (b) Aspects: resolution, fonts.

- **Att** is the model of Att+GCN removing the GCN module.

Table 3 shows the performances of all these models. It is clear to see that, comparing with GCN-reduced models, the two models with GCN achieve higher performance, respectively. The results verify that the modeling of the sentiment dependencies between different aspects with GCN plays a great role in predicting the sentiment polarities of aspects.

4.5. Impact of GCN layer number

The number of GCN layers is one very important setting parameter that affects the performance of our model. In order to investigate the impact of the GCN layer number, we conduct experiment with the different number of GCN layers from 1 to 8. The performance results are shown in Fig. 5. As can be seen from the results, in general, when the number of GCN layers is 2, the model works best. When the number of GCN layers is bigger than 2, the performance drops with the increase of the number of GCN layers on both the datasets. The possible reason for the phenomenon of the performance drop may be that with the increase of the model parameters, the model becomes more difficult to train and over-fitting.

4.6. Case study

In order to have an intuitive understanding of the difference between with-GCN model (our proposed model) and without-GCN model, we use two examples with multiple aspects from laptop dataset as a case study. We draw heat maps to visualize the attention weights on the words computed by the two models, as shown in Fig. 6. The deeper the color, the more attention the model pays to it.

As we can see from the first example, i.e., *"i love the keyboard and the screen."*, with two aspects *"keyboard"* and *"screen"*, without-GCN model mainly focuses on the word *"love"* to predict the sentiment polarities of the two aspects. While for with-GCN model, besides the word *"love"*, it also pays attention to the conjunction *"and"*. This phenomenon

indicates that with-GCN model captures the sentiment dependencies of the two aspects through the word *"and"*, and then predicts the sentiments of *"keyboard"* and *"screen"* simultaneously.

The second example is *"air has higher resolution but the fonts are small."* with two aspects *"resolution"* and *"fonts"*. It is obvious that the sentiments of the two aspects *"resolution"* and *"fonts"* are opposite connected by the conjunction *"but"*. Without-GCN model predicts the polarity of aspect *"resolution"* by the word *"higher"* and the polarity of aspect *"fonts"* by the word *"small"* in isolation, which ignores the relation between the two aspects. In the contrary, with-GCN model enforces the model to pay attention on the word *"but"* when predicting the sentiment polarity for aspect *"fonts"*.

From these examples, we can observe that our proposed model (with-GCN model) not only focuses the corresponding words which are useful for predicting the sentiment of each aspect, but also considers the textual information which is helpful for judging the relation between different aspects. By using attention mechanism to focus on the textual words describing the interdependence between different aspects, the downstream GCN module can effectively further represent the sentiment dependencies between different aspects in one sentence. With more useful information, our proposed model can predict aspect-level sentiment category more accurately.

5. Conclusion

In this paper, we design a novel GCN based model (SDGCN) for aspect-level sentiment classification. The key idea of our model is to employ GCN to model the sentiment dependencies between different aspects in one sentence. Specifically, SDGCN first adopts bidirectional attention mechanism with position encoding to obtain aspect-specific representations, then captures the sentiment dependencies via message passing between aspects. Thus, SDGCN benefits from such dependencies which are always ignored in previous studies. Experiments on

SemEval 2014 verify the effectiveness of the proposed mode, and SDGCN-BERT obtains new state-of-the-art results. The case study shows that SDGCN can not only pay attention to those words which are important for predicting the sentiment polarities of aspects, but also pay attention to the words which are helpful for judging the sentiment dependencies between different aspects.

In our future work, we will explore how to build a more precise sentiment graph structure between aspects. The two kinds of undirected sentiment graphs in this work are coarse. We conjecture that making use of textual information to define a graph may create a better graph structure.

References

[1] B. Pang, L. Lee, Opinion mining and sentiment analysis, Synthesis Lectures on Human Language Technologies 2(1-2) (2008) 1–135.

[2] B. Liu, Sentiment analysis and opinion mining, Foundations and Trends in Information Retrieval 5(1) (2012) 1–167.

[3] S. Kiritchenko, X. Zhu, C. Cherry, S. Mohammad, Nrccanada-2014: Detecting aspects and sentiment in customer reviews, in: Proceedings of the 8th International Workshop on Semantic Evaluation, 2014, pp. 437–442.

[4] J. Wagner, P. Arora, S. Cortes, U. Barman, D. Bogdanova, J. Foster, L. Tounsi, Dcu: Aspect-based polarity classification for semeval task 4, in: Proceedings of the 8th international workshop on semantic evaluation (SemEval 2014), 2014, pp. 223–229.

[5] S. Poria, E. Cambria, D. Hazarika, P. Vij, A deeper look into sarcastic tweets using deep convolutional neural networks, in: 26th International Conference on Computational Linguistics, 2016, pp. 1601–1612.

[6] D. Tang, B. Qin, X. Feng, T. Liu, Effective lstms for target-dependent sentiment classification, in: Proceedings of COLING, the 26th International Conference on Computational Linguistics: Technical Papers, 2016, pp. 3298–3307.

[7] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, Recurrent models of visual attention, in: Advances in Neural Information Processing Systems 27, 2014, pp. 2204–2212.

[8] D. Bahdanau, K. Cho, Y. Bengio, Neural machine translation by jointly learning to align and translate, in: 3rd International Conference on Learning Representations, 2015.

[9] Y. Wang, M. Huang, L. Zhao, X. Zhu, Attention-based lstm for aspect-level sentiment classification, in: Proceedings of the conference on empirical methods in natural language processing, 2016, pp. 606–615.

[10] P. Chen, Z. Sun, L. Bing, W. Yang, Recurrent attention network on memory for aspect sentiment analysis, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 452–461.

[11] D. Ma, S. Li, X. Zhang, H. Wang, Interactive attention networks for aspect-level sentiment classification, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 4068–4074.

[12] Y. Ma, H. Peng, E. Cambria, Targeted aspect-based sentiment analysis via embedding commonsense knowledge into an attentive lstm, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 5876–5883.

[13] Y. Song, J. Wang, T. Jiang, Z. Liu, Y. Rao, Attentional encoder network for targeted sentiment classification, CoRR abs/1902.09314.

[14] R. Ma, K. Wang, T. Qiu, A. K. Sangaiah, D. Lin, H. B. Liaqat, Feature-based compositing memory networks for aspect-based sentiment classification in social internet of things, Future Generation Comp. Syst. 92 (2019) 879–888.

[15] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, 2017.

[16] Y. Kim, Convolutional neural networks for sentence classification, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, 2014, pp. 1746–1751.

[17] F. Abid, M. Alam, M. Yasir, C. Li, Sentiment analysis through recurrent variants latterly on convolutional neural network of twitter, Future Generation Comp. Syst. 95 (2019) 292–308.

[18] D. Rao, D. Ravichandran, Semi-supervised polarity lexicon induction, in: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, 2009, pp. 675–682.

[19] X. Ding, B. Liu, P. S. Yu, A holistic lexicon-based approach to opinion mining, in: Proceedings of the 2008 international conference on web search and data mining, 2008, pp. 231–240.

[20] L. Jiang, M. Yu, M. Zhou, X. Liu, T. Zhao, Target-dependent twitter sentiment classification, in: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, 2011, pp. 151–160.

[21] L. Dong, F. Wei, C. Tan, D. Tang, M. Zhou, K. Xu, Adaptive recursive neural network for target-dependent twitter sentiment classification, in: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, 2014, pp. 49–54.

[22] T. H. Nguyen, K. Shirai, Phrasernn: Phrase recursive neural network for aspect-based sentiment analysis, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015, pp. 2509–2514.

[23] C. Li, X. Guo, Q. Mei, Deep memory networks for attitude identification, in: Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2017, pp. 671–680.

[24] W. L. Xin Li, Lidong Bing, B. Shi, Transformation networks for target-oriented sentiment classification, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, 2018, pp. 946–956.

[25] X. Ma, J. Zeng, L. Peng, G. Fortino, Y. Zhang, Modeling multi-aspects within one opinionated sentence simultaneously for aspect-level sentiment analysis, Future Generation Comp. Syst. 93 (2019) 304–311.

[26] J. Bruna, W. Zaremba, A. Szlam, Y. LeCun, Spectral networks and locally connected networks on graphs, in: Proceedings of the 2nd International Conference on Learning Representations, 2014.

[27] M. Henaff, J. Bruna, Y. LeCun, Deep convolutional networks on graph-structured data, CoRR abs/1506.05163 (2015).

[28] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional neural networks on graphs with fast localized spectral filtering, in: Advances in Neural Information Processing Systems 29, 2016, pp. 3837–3845.

[29] X. Qi, R. Liao, J. Jia, S. Fidler, R. Urtasun, 3d graph neural networks for RGBD semantic segmentation, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, 2017, pp. 5209–5218.

[30] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, X. Wang, Factorizable net: An efficient subgraph-based framework for scene graph generation, in: Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part I, 2018, pp. 346–363.

[31] Z. Chen, X. Wei, P. Wang, Y. Guo, Multi-label image recognition with graph convolutional networks, CoRR abs/1904.03582 (2019).

[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] J. Bastings, I. Titov, W. Aziz, D. Marcheggiani, K. Sima'an, Graph convolutional encoders for syntax-aware neural machine translation, in: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing7, 2017, pp. 1957–1967.

[34] Y. Li, R. Jin, Y. Luo, Classifying relations in clinical narratives using segment graph convolutional and recurrent neural networks (seg-gcrns), JAMIA 26 (3) (2019) 262–268.

[35] H. Peng, J. Li, Y. He, Y. Liu, M. Bao, L. Wang, Y. Song, Q. Yang, Large-scale hierarchical text classification with recursively regularized deep graph-cnn, in: Proceedings of the 2018 World Wide Web Conference on World Wide Web, 2018, pp. 1063–1072.

[36] Y. Zhang, Q. Liu, L. Song, Sentence-state LSTM for text representation, in: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers, 2018, pp. 317–327.

[37] J. Pennington, R. Socher, C. D. Manning, Glove: Global vectors for word representation, in: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the

ACL, 2014, pp. 1532–1543.

[38] J. Devlin, M. Chang, K. Lee, K. Toutanova, BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers), 2019, pp. 4171–4186.

[39] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, Semeval-2014 task 4: Aspect based sentiment analysis, in: Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014., 2014, pp. 27–35.

[40] D. Tang, B. Qin, T. Liu, Aspect level sentiment classification with deep memory network, in: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016, 2016, pp. 214–224.

[41] S. Gu, L. Zhang, Y. Hou, Y. Song, A position-aware bidirectional attention network for aspect-level sentiment analysis, in: Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018, 2018, pp. 774–784.