End-to-End Emotion-Cause Pair Extraction with Graph Convolutional Network

Ying Chen[1] **Wenjun Hou**[1] **Shoushan Li**[2] **Caicong Wu**[1] **Xiaoqiang Zhang**[1]
[1]College of Information and Electrical Engineering, China Agricultural University, China
[2]Natural Language Processing Lab, Soochow University, China
{chenying,houwenjun,wucc,xqzhang}@cau.edu.cn
lishoushan@suda.edu.cn

Abstract

Emotion-cause pair extraction (ECPE), which aims at simultaneously extracting emotion-cause pairs that express emotions and their corresponding causes in a document, plays a vital role in understanding natural languages. Considering that a cause usually appears around its corresponding emotion, we construct a pair graph and a Pair Graph Convolutional Network (PairGCN) to model dependency relations among local neighborhood candidate pairs. Moreover, in our proposed graph, there are three types of dependency relations and each type of dependency relations has its own way to propagate contextual information. Experiments on a benchmark Chinese emotion-cause pair extraction corpus demonstrate the effectiveness of the proposed model.

1 Introduction

Emotion-cause pair extraction (ECPE), which was first proposed in Xia & Ding (2019), aims to extract emotion expressions and their corresponding causes in a document simultaneously. Different from emotion cause extraction (ECE) (Lee et al., 2010; Gui et al., 2016) which extracts the causes for given emotion expressions, EPCE is a much more challenging task.

There has been a surging interest in developing neural models either for emotion cause extraction or for emotion extraction, while ECPE, as a special causal relation extraction task, is newly proposed and remains largely unexplored. Previous research on ECPE (Xia and Ding, 2019) focused on designing pipeline systems in which emotion clauses and cause clauses are extracted separately, and the two sets of clauses are paired to generate candidate emotion-cause pairs, and then emotion-cause pairs are selected from these candidate pairs with a filter. Hence, prediction errors unavoidably accumulate through the pipeline framework. Therefore, in this work, we aim to design an end-to-end framework in which any two clauses in a document are paired (i.e., one is a candidate emotion clause, and the other is a candidate cause clause) so as to generate candidate emotion-cause pairs and then emotion-cause pairs are selected from these candidate pairs. E.g., in Fig. 1, there are 25 candidate pairs, and only (c_4, c_2) and (c_4, c_3) are emotion-cause pairs.

Furthermore, modelling contextual information for a candidate pair is also crucial for ECPE. In previous pipeline systems (Xia and Ding, 2019), two features were extracted for each clause in a document for emotion extraction and emotion cause extraction respectively, and then a candidate pair was represented by the combination of these clause-level features. However, dependency among candidate pairs does not take into account. In fact, an emotion usually has a few cause clauses occurring within a specific distance from the emotion expression. E.g, in the Chinese ECPE corpus provided by Xia & Ding (2019), $\sim$90% emotions has one and only one cause clause, and $\sim$96% cause clauses occur within a window size of 2 from their corresponding emotion clauses. The emotion-cause co-occurrence property indicates that in a local neighborhood if one candidate pair has been detected as an emotion-cause pair, other candidate pairs are usually non-emotion-cause pairs. Thus, modelling contextual information should consider pair-level dependency. Here, a local neighborhood refers to a set of candidate pairs whose candidate emotion

Proceedings of the 28th International Conference on Computational Linguistics, pages 198–207
Barcelona, Spain (Online), December 8-13, 2020

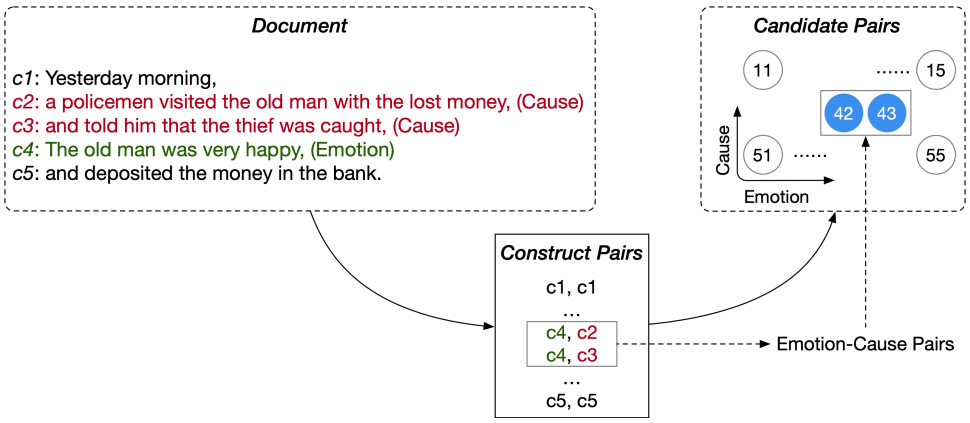

Figure 1: An ECPE example for a document consisting of 5 clauses. There are two emotion-cause pairs in the document: (c4, c2) and (c4, c3).

clauses are the same, and candidate cause clauses are not far away from each other. In this paper, we propose a novel Pair Graph Convolutional Network (PairGCN), an end-to-end model for ECPE. We first construct a pair graph to model three types of dependency relations among the candidate pairs in a local neighborhood, where a node represents a candidate pair and an edge connecting two nodes represents a dependency relation between the corresponding two candidate pairs. Then, a Graph Convolutional Network (GCN) is designed to use the three types of edges to propagate contextual information in the pair graph.

Above all, our main contributions can be summarized as follow:

- We propose a PairGCN model that utilizes an end-to-end framework for ECPE.
- We design a Graph Convolutional Network to model three types of dependency relations among local neighborhood candidate pairs so as to facilitate the extraction of pair-level contextual information.
- Our model is evaluated on a benchmark Chinese emotion-cause pair extraction dataset for three tasks, i.e., emotion-cause pair extraction, emotion extraction, and emotion cause extraction. Experimental results demonstrate the effectiveness of our PairGCN model.

2 Related Works

In this section, we will briefly summarise related research on emotion cause extraction (ECE), emotion-cause pair extraction (ECPE), and graph neural networks (GNNs).

2.1 Emotion Cause Extraction and Emotion-Cause Pair Extraction

The task of emotion cause extraction (ECE) which extracts the causes of given emotion keywords has been intensively studied for years. Most of the previous works focused on contextual information extraction from the context of the given emotion keyword either with manual rules or with machine learning methods. Lee et al. (2010) constructed an emotion cause corpus from Sinica Corpus and then built a rule-based system to extract linguistic features. Based on this corpus, Chen et al. (2010) proposed a multi-label approach with linguistic patterns that can capture linguistic cues in contexts with manual rules. Other rule-based feature extraction methods (Neviarouskaya and Aono, 2013; Li and Xu, 2014; Gao et al., 2015a; Gao et al., 2015b; Yada et al., 2017; Yu et al., 2019) were also proposed to extract contextual features.

Other than rule-based methods, Gui et al. (2016) constructed a Chinese event-driven ECE corpus with SINA city news and proposed a convolution kernel-based multi-kernel Support Vector Machine (SVM) to

extract contextual features from given syntactical trees. Afterward, deep learning has attracted attention from the ECE research community. Gui et al. (2017) converted the ECE task to a Question Answering (QA) task and proposed a Convolutional Multiple-Slot Deep Memory Network to store relevant contextual information. Other neural models (Li et al., 2018; Xu et al., 2019) were also proposed to extract contextual information. Besides, Chen et al. (2018) presented a neural network-based joint approach for emotion extraction and emotion cause extraction to capture mutual benefits across these two emotion analysis tasks.

Different from ECE in which emotion keywords are provided before the extraction of their causes, the task of emotion-cause pair extraction (ECPE) was first proposed in Xia & Ding (2019), in which emotions and their corresponding causes are extracted at the same time. For this new task, they proposed a two-step approach, which firstly extracted emotion clauses and cause clauses individually using an interactive multi-task learning network which consists of two hierarchical BiLSTM networks, and then each emotion clause was paired with each cause clause and these candidate pairs were filtered by a logistic regression model. Overall, in the previous works on ECE and ECPE, modelling contextual information does not consider dependency relations among local neighborhood candidate pairs.

2.2 Graph Neural Networks

The Graph Convolutional Network (GCN) was first proposed in Kipf & Welling (2017) for node classification, which operated directly on a graph. After that, Graph Neural Networks have been widely applied to various NLP tasks, such as relation extraction, aspect-level sentiment analysis, and text classification. Zhang et al. (2018) used GCNs to capture long-range relations among dependency trees and further applied a novel pruning strategy to the input trees. Sun et al. (2019) proposed a GCN for aspect-level sentiment analysis, which propagated both contextual and dependency information from opinion words to aspect words. In addition, Yao et al. (2019) built a text graph based on word co-occurrence and document-word relations and then learned a Text Graph Convolutional Network for text classification. Ghosal et al.(2019) used two layers of GCNs to capture speaker information for emotion recognition in conversations. In this paper, we attempt to use GCNs to model dependency relations in a local neighborhood so as to capture pair-level contextual information for ECPE.

3 Methodology

In this section, we briefly introduce the definition of ECPE. Then, we describe our PairGCN which models two types of contexts for ECPE: sequential clause context and pair-level context. The former refers to the clause sequence in a given document that provides sequential information for each clause, and the latter refers to the candidate pairs in a local neighborhood which gives dependency information for each candidate pair. Accordingly, as illustrated in Fig. 2, there are two encoders in our PairGCN: clause-level context encoder and pair-level context encoder. The clause-level context encoder uses two hierarchical BiLSTM networks to model the sequential clause context and then extracts an emotion feature and a cause feature for a clause respectively. The pair-level context encoder uses a Pair Graph Convolutional Network to model the pair-level context and then extracts a contextual feature for a candidate pair. Finally, classification assigns a label to a candidate pair according to its feature representation.

3.1 Task Definition

Given a document $D = \{c_1, c_2, \ldots, c_L\}$, the clauses are formed into a set of candidate emotion-cause pairs P using the Cartesian product.

$$P = \{\ldots, c^p_{i,j}, \ldots\} \tag{1}$$

$$c^p_{i,j} = (c^e_i, c^c_j) \tag{2}$$

where c^e_i is clause c_i serving as a candidate emotion clause, c^c_j is clause c_j serving as a candidate cause clause, and there are totally $L \times L$ candidate pairs in P. Given a candidate pair $c^p_{i,j} = (c^e_i, c^c_j)$, ECPE

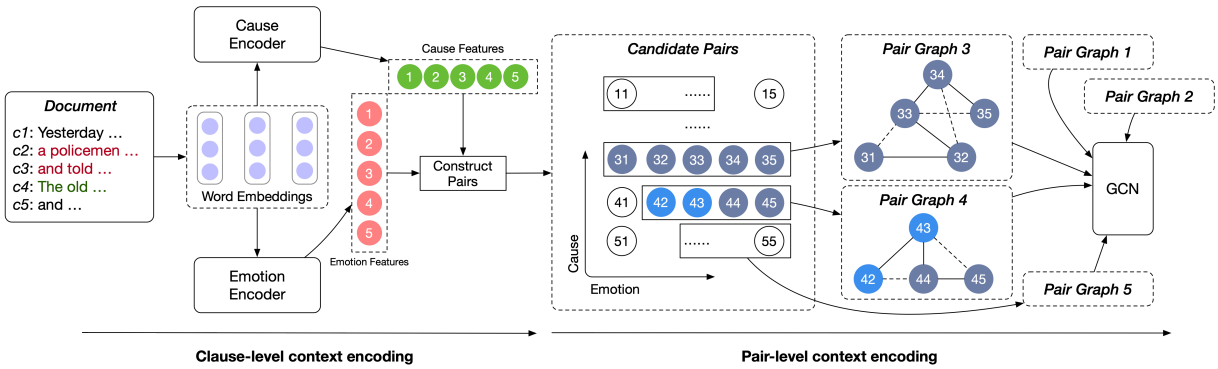

Figure 2: The overview of our Pair Graph Convolutional Network. In a pair graph, solid lines are $D1$ edges, dashed lines are $D2$ edges, and SL edges are eliminated for simplification. In addition, blue nodes represent emotion-cause pairs, and both white nodes and gray nodes are non-emotion-cause pairs.

assigns a binary label, where "1" means that clause c_i^e expresses an emotion and clause c_j^c provides the cause of this emotion, and "0" indicates that such an emotion-cause relation does not exist in the candidate pair. E.g., in Fig.1, (c_4, c_2) is an emotion-cause pair (with label "1") and (c_4, c_5) is a non-emotion-cause pair (with label "0").

3.2 Clause-level Context Encoder

For clause c_t, we employ a clause-level context encoder to extract two features based on its sequential clause context: the clause-level emotion feature $\mathbf{v}_t^e$ when c_t serves as a candidate emotion clause, and the clause-level cause feature $\mathbf{v}_t^c$ when c_t serves as a candidate cause clause. In order to extract the emotion features and the cause features respectively, the clause-level context encoder consists of two hierarchical BiLSTM networks (i.e., the cause encoder and the emotion encoder in Fig. 2), and each hierarchical BiLSTM network consists of a word-level BiLSTM and a clause-level BiLSTM. Finally, for the document with L clauses, an emotion feature sequence representation $\mathbf{u}^e = \{\mathbf{u}_1^e, \mathbf{u}_2^e, \ldots, \mathbf{u}_L^e\}$ and a cause feature sequence representation $\mathbf{u}^c = \{\mathbf{u}_1^c, \mathbf{u}_2^c, \ldots, \mathbf{u}_L^c\}$ are obtained respectively by the word-level BiLSTM, and an emotion feature sequence representation $\mathbf{v}^e = \{\mathbf{v}_1^e, \mathbf{v}_2^e, \ldots, \mathbf{v}_L^e\}$ and a cause feature sequence representation $\mathbf{v}^c = \{\mathbf{v}_1^c, \mathbf{v}_2^c, \ldots, \mathbf{v}_L^c\}$ are obtained respectively by the clause-level BiLSTM. Since the word-level BiLSTM is similar to the one used in Xia & Ding (2019), we omit the details for limited space and only present our clause-level BiLSTM.

To further capture contextual information for a clause from the perspective of the whole document, we feed either $\mathbf{u}^e$ or $\mathbf{u}^c$ to a clause-level BiLSTM. Moreover, although emotions can be identified solely without their causes, identifying whether an event is the cause of an emotion could be much difficult if the relevant emotion information does not be provided. Thus, our clause-level BiLSTM uses different input to extract emotion features (see Eq. 3) and cause features (see Eq. 4) respectively:

$$\mathbf{v}_t^e = \mathbf{BiLSTM}_c^e(\mathbf{u}_t^e) \tag{3}$$

$$\mathbf{v}_t^c = \mathbf{BiLSTM}_c^c([\mathbf{u}_t^e, \mathbf{u}_t^c]) \tag{4}$$

where $[,]$ is the concatenating function, $\mathbf{BiLSTM}_c^e$ is a clause-level BiLSTM to extract an emotion feature $\mathbf{v}_t^e \in \mathbb{R}^{2d_h}$, and $\mathbf{BiLSTM}_c^c$ is another clause-level BiLSTM to extract a cause feature $\mathbf{v}_t^c \in \mathbb{R}^{2d_h}$.

3.3 Pair-level Context Encoder

We propose a pair-level context encoder to extract contextual information that can capture dependency among local neighborhood candidate pairs. We construct a pair graph (e.g., Pair Graph 3 and 4 in Fig. 2) to model the candidate pairs in a local neighborhood. Then, we design a feature transformation process (i.e., GCN in Fig. 2) to transform clause-level contextual features into pair-level contextual features.

Distance	Number	Percent	Distance	Number	Percent
0	511	23.6	≤ 0	511	23.6
1	1342	61.9	≤ 1	1853	85.5
2	224	10.3	≤ 2	2077	95.8

Table 1: Statistics of distances between emotion clauses and their cause clauses in the Chinese ECPE corpus (Xia and Ding, 2019).

3.3.1 Pair Graph Construction

Nodes: Given the set of candidate emotion-cause pairs P, each candidate pair is considered as a node. Moreover, a candidate pair $c^p_{i,j} = (c^e_i, c^c_j)$ is represented as $\mathbf{v}^p_{i,j}$ which concatenates the emotion feature $\mathbf{v}^e_i$ and the cause feature $\mathbf{v}^c_j$ output from the clause-level context encoder:

$$\mathbf{v}^p_{i,j} = [\mathbf{v}^e_i, \mathbf{v}^c_j] \tag{5}$$

Edges: because the candidate pairs in a local neighborhood have the same candidate emotion clause, we build a pair graph for the candidate emotion clause. In the case of a document with L clauses, there are L pair graphs in total. E.g., in Fig. 2, there are 5 pair graphs for a document with 5 clauses. Furthermore, a cause clause is likely to appear 1 or 2 offset of its emotion clause. E.g., as illustrated in Table 1, 95.8% cause clauses are mentioned within a window size of 2 from their corresponding emotion clauses. Therefore, during building a pair graph for a candidate emotion clause c^e_i, the nodes in its corresponding pair graph are:

$$c^p_{i,[i-2:i+2]} = \{c^p_{i,i-2}, c^p_{i,i-1}, c^p_{i,i}, c^p_{i,i+1}, c^p_{i,i+2}\} \tag{6}$$

Considering that a node has different influences to its neighboring nodes, three types of edges, namely SL, $D1$, and $D2$ edges, are used in a pair graph:

(1) SL edge: This is a self-loop edge for the self-transformation of a node.

(2) $D1$ edge: This is an edge connecting two nodes which have a distance of 1 between their candidate cause clauses (e.g., the edges between $c^p_{i,i}$ and $c^p_{i,i\pm1}$).

(3) $D2$ edge: This is an edge connecting two nodes which have a distance of 2 between their candidate cause clauses (e.g., the edges between $c^p_{i,i}$ and $c^p_{i,i\pm2}$).

The incorporation of these edges into a pair graph forms the three types of dependency relations among the candidate pairs in a local neighborhood and allows the contextual information transmit through these edges, which in succession would facilitate the extraction of the pair-level contextual features.

3.3.2 Feature Transformation

Inspired by Ghosal et al.(2019), we use two layers of GCN (i.e., two transformations) to capture the contextual information for a node in a pair graph.

For node $c^p_{i,j}$, the first transformation is applied to obtain its representation $\mathbf{g}^1_{i,j}$ using the features output from the clause-level context encoder. Specifically, the features of the nodes in the pair graph are aggregated with different transformation parameters according to the types of their edges linked to $c^p_{i,j}$:

$$\mathbf{g}^1_{i,j} = \sigma(\frac{1}{z}\sum_{k\in D1}\mathbf{v}^p_{i,k}\mathbf{W}^1_{D1} + \frac{1}{z}\sum_{t\in D2}\mathbf{v}^p_{i,t}\mathbf{W}^1_{D2} + \frac{1}{z}\mathbf{v}^p_{i,j}\mathbf{W}^1_{SL}) \tag{7}$$

where $\mathbf{W}^1_{D1} \in \mathbb{R}^{d_{in}\times d_{out}}$, $\mathbf{W}^1_{D2} \in \mathbb{R}^{d_{in}\times d_{out}}$, and $\mathbf{W}^1_{SL} \in \mathbb{R}^{d_{in}\times d_{out}}$ are weight matrices for the nodes linked to node $c^p_{i,j}$ with $D1$ edges, $D2$ edges, and SL edges respectively. In addition, z is a normalization factor which is the node degree. σ is a non-linear activation function and $ReLU$ (Nair and Hinton, 2010) is used in this paper.

After that, the second transformation is applied to obtain the representation $\mathbf{g}_{i,j}^2$ for node $c_{i,j}^p$ using the features output from the first transformation:

$$\mathbf{g}_{i,j}^2 = \sigma(\sum_{k \in D1} \mathbf{g}_{i,k}^1 \mathbf{W}_{D1}^2 + \sum_{t \in D2} \mathbf{g}_{i,t}^1 \mathbf{W}_{D2}^2 + \mathbf{g}_{i,j}^1 \mathbf{W}_{SL}^2) \tag{8}$$

where $\mathbf{W}_{D1}^2 \in \mathbb{R}^{d_{out} \times d_{out}}$, $\mathbf{W}_{D2}^2 \in \mathbb{R}^{d_{out} \times d_{out}}$, and $\mathbf{W}_{SL}^2 \in \mathbb{R}^{d_{out} \times d_{out}}$ are weight matrices for the normalized nodes linked to node $c_{i,j}^p$.

Compared to the feature transformation process used in Ghosal et al. (2019), we distinguish contextual information propagation through $D1$ edges and $D2$ edges and use different weight matrices to deal with the two propagations separately. Moreover, using the two transformations plus $D2$ edges, contextual information can be propagated between any two nodes with the greatest distance in a pair graph. E.g., in Pair Graph 3 in Fig. 2, the information on $c_{3,1}^p$ and $c_{3,5}^p$ can be transmit to each other through the two transformations which use two $D2$ edges (i.e., $c_{3,1}^p \leftrightarrow c_{3,3}^p \leftrightarrow c_{3,5}^p$).

3.4 Classification

Emotion-Cause Pair Extraction: Since two clauses in an emotion-cause pair are likely to appear within a specific distance (see Table 1), distance information needs to be taken into consideration during classification (Xia and Ding, 2019). Thus, for a candidate pair $c_{i,j}^p$, its representation for classification is the concatenation of $\mathbf{g}_{i,j}^2$ and $\mathbf{d}_{i,j}$, where $\mathbf{d}_{i,j} \in \mathbb{R}^{d_{dis}}$ is a distance embedding. Then, a softmax function is applied as follows:

$$\hat{p}_{i,j} = softmax(\mathbf{W}_p^T [\mathbf{g}_{i,j}^2, \mathbf{d}_{i,j}] + \mathbf{b}_p) \tag{9}$$

where $\mathbf{W}_p \in \mathbb{R}^{(d_{out}+d_{dis}) \times d_p}$ is a weight matrix, and $\mathbf{b}_p \in \mathbb{R}^{d_{pout}}$ is a bias vector. Finally, we obtain the predicted probability distribution $\hat{p}_{i,j}$ and the corresponding predicted label $\hat{EC}_{i,j}$ for the candidate pair $c_{i,j}^p$. During model training, we use Cross-Entropy loss as loss function.

Emotion Extraction and Emotion Cause Extraction: After obtaining the ECPE predictions for all candidate pairs, we can extract emotion clauses and cause clauses from them. Specifically, for emotion extraction, the prediction label $\hat{E}_i$ for clause c_i can be obtained as:

$$\hat{E}_i = \begin{cases} 1, & if \ \sum_{j=1}^{L} (\hat{EC}_{i,j}) > 0 \\ 0, & otherwise \end{cases} \tag{10}$$

Similarly, for emotion cause extraction, the prediction label $\hat{C}_j$ for clause c_j can be obtained.

4 Experiments

4.1 Datasets and Metrics

We evaluate the performance of our model on a Chinese ECPE corpus released by Xia & Ding (2019), which was constructed from a benchmark Chinese ECE corpus (Gui et al., 2016). In the Chinese ECPE corpus, there are 1,945 documents and 490,367 pair candidates in total, including 2,167 emotion-cause pairs and 488,200 non-emotion-cause pairs. In other words, there are less than 1% emotion-cause pairs in this corpus.

Similar to previous work (Xia and Ding, 2019), we evaluate our model on three tasks: emotion-cause pair extraction, emotion extraction, and emotion cause extraction. To obtain statistically credible results, we use their data-split setting, repeat the experiments 10 times, and then report the average results of precision (P), recall (R), and F_1-score (F_1) to evaluate the performances of our model. Moreover, for each experiment, we set aside 10% of training documents as development set.

4.2 Experimental Settings

In our experiments, we follow experimental settings in Xia & Ding (2019), using the same word embeddings pre-trained on the corpora from Chinese Weibo [1] with Word2Vec (Mikolov et al., 2013). Moreover, BERT representations (Devlin et al., 2019) are also utilized, where we use the based Chinese model. While extracting BERT embeddings, the basic input unit is a clause. Besides, the dimension of distance embeddings is 50, and other parameters of our models are listed in Table 2. Finally, the learnable parameters (including all weight matrices and bias vectors) are randomly initialized by a uniform distribution of $U(-0.01, 0.01)$.

	Word2Vec	BERT
# dimension of word embeddings	200	768
# hidden unit of BiLSTM	100	200
# hidden unit of GCN	100	200

Table 2: The experiment settings of our models.

While training, we use the Adam optimizer (Kingma and Ba, 2015) to update all parameters. Each training batch contains 32 documents, and the learning rate is set to 0.005. To reduce over-fitting, dropout (Srivastava et al., 2014) is applied to all features vectors, including word embeddings and hidden representations, and it is set to 0.5.

4.3 Model Comparison

In order to evaluate the performance of our model, we make a comparison with the following three pipeline systems (1), (2), and (3), and one end-to-end system (4).

(1) Indep: This is an interactive multi-task learning pipeline system, which extracts emotion clauses and cause clauses using two hierarchical BiLSTM independently. Then, the two sets of clauses are paired with each other and emotion-cause pairs are extracted using a filter (Xia and Ding, 2019).

(2) Inter-CE: This is an enhanced version of Indep, which is capable of capturing the correlation between emotions and causes. While extracting emotion clauses and cause clauses, emotion cause extraction is used to improve emotion extraction (Xia and Ding, 2019).

(3) Inter-EC: This is another enhanced version of Indep, while it uses emotion extraction to improve emotion cause extraction during extracting emotion clauses and cause clauses (Xia and Ding, 2019).

(4) Hier-BiLSTM: This is an end-to-end model, which extracts emotion features and cause features using two hierarchical BiLSTM independently, and the concatenation of an emotion feature and a cause feature is used to represent a candidate pair. Specifically, the hierarchical BiLSTM is similar to the one used in our clause-level context encoder, except that the input to the clause-level BiLSTM in the cause encoder is only the word-level cause feature $\mathbf{u}_t^c$ (see Eq. 4).

4.4 Results

We first compare our PairGCN model with the three pipeline systems. As shown in Table 3, PairGCN outperforms all pipeline systems on ECPE. E.g., compared to the best pipeline system (i.e., Inter-EC), the F_1 score of PairGCN rises from 61.28% to 63.21%. Specifically, this performance gain mainly comes from the improvement on the precision score. Furthermore, PairGCN achieves lower recall and yet higher precision than Inter-EC on both emotion extraction and emotion cause extraction. This indicates that although less correct emotion cases and cause cases are detected by our PairGCN model, they are more likely to be matched with each other so as to lead to a significant increasing in the precision score of ECPE. Moreover, in terms of emotion extraction, though PairGCN is not trained with only emotion

[1] https://www.weibo.com/

Model	Emotion Extraction			Cause Extraction			EC Pair Extraction		
	P	R	F_1	P	R	F_1	P	R	F_1
Indep*	83.75	80.71	82.10	69.02	56.73	62.05	68.32	50.82	58.18
Inter-CE*	84.94	**81.22**	**83.00**	68.09	56.34	61.51	69.02	51.35	59.01
Inter-EC*	83.64	81.07	82.30	70.41	**60.83**	65.07	67.21	57.05	61.28
Hier-BiLSTM	**86.16**	66.29	74.80	72.27	55.32	62.48	69.25	53.71	60.30
PairGCN	85.87	72.08	78.29	**72.83**	59.53	**65.41**	**69.99**	**57.79**	**63.21**
Hier-BiLSTM-BERT	**88.80**	74.70	81.00	78.03	65.35	70.96	75.37	64.34	69.26
PairGCN-BERT	88.57	**79.58**	**83.75**	**79.07**	**69.28**	**73.75**	**76.92**	**67.91**	**72.02**

Table 3: Experimental results of different models. "EC Pair Extraction" denotes the ECPE task. * denotes that experimental results are cited from Xia & Ding (2019).

labels, it still shows competitive performance with a F_1 score at 78.29%, compared to Inter-CE with the highest performance 83.0%.

Compared to the end-to-end baseline Hier-BiLSTM, our PairGCN model has great improvement over the three tasks with two types of embeddings. As shown in Table 3, compared to Hier-BiLSTM (or Hier-BiLSTM-BERT), the F_1 scores of PairGCN (or PairGCN-BERT) on emotion extraction and emotion cause extraction rise by $\sim$3%, and as a result, the F_1 score on ECPE increases $\sim$3%. This performance gain mainly comes from the significant improvement in recall scores on the three tasks. This means that PairGCN is capable of detecting more emotion-cause pairs with the help of the contextual features extracted by the pair-level context encoder.

4.5 Ablation Study

To further explore the effects of the three types of edges in our full model (i.e., PairGCN and PairGCN-BERT), we perform an ablation study and show the results in Table 4.

First of all, we investigate the impact of the pair-level context encoder by removing GCN from our full model (i.e., PairGCN w/o GCN and PairGCN-BERT w/o GCN). In other words, only features extracted by the clause-level context encoder (see Eq. 5) are feed to classification. As we can see from Table 4, compared to our full model, their F_1 scores of ECPE drop significantly ($\sim$2%) with the two types of embeddings. The decreasing performance indicates that the GCN-based feature transformation process in the pair-level context encoder can effectively augment the effects of the features extracted by the clause-level context encoder on ECPE. This is also reflected by the improved performances of the other two tasks, i.e., emotion extraction and emotion cause extraction.

Secondly, from Table 4, we observe that after removing one type of edges (i.e., either $D1$ or $D2$) from our full model, the overall performance of ECPE degrades. E.g., for models removing $D1$ (i.e., PairGCN w/o D1 and PairGCN-BERT w/o D1), their F_1 score drops 0.9% and 0.7% with Word2Vec embeddings and BERT embeddings respectively. For models removing $D2$ (i.e., PairGCN w/o D2 and PairGCN-BERT w/o D2), their F_1 score drops $\sim$1.4% with the two types of embeddings. This indicates that the contextual information which is propagated either through $D1$ edges or through $D2$ edges in the pair-level context encoder is very useful for ECPE. Furthermore, compared to models removing $D1$, models removing $D2$ perform worse. Compared to $D1$ edges, $D2$ edges allows contextual information propagate more straightforward because of their greater distance and their own weight matrices (see Section 3.3.2), and therefore, the pair-level contextual information is effectively captured for ECPE.

Finally, compared to models removing $D2$ (i.e., PairGCN w/o D2 and PairGCN-BERT w/o D2) , the performances of models removing both $D1$ and $D2$ (i.e., PairGCN w/o D1&D2 and PairGCN-BERT w/o D1&D2) decrease slightly. This also confirms that it is necessary to distinguish $D1$ edges and $D2$ edges in a pair graph because of their different ways to propagate contextual information. Although information propagation through $D1$ edges and information propagation through $D2$ edges are relevant, they are not interchangeable. E.g., in Pair Graph 3 in Fig. 2, information propagation between $c^p_{3,1}$ and $c^p_{3,3}$ can be made through either of the two paths: the combination of two $D1$ edges (i.e., $c^p_{3,1} \leftrightarrow c^p_{3,2}$

Model	Emotion Extraction			Cause Extraction			EC Pair Extraction		
	P	R	F_1	P	R	F_1	P	R	F_1
PairGCN	85.87	72.08	78.29	72.83	59.53	65.41	69.99	57.79	63.21
PairGCN w/o D1	85.90	70.89	77.56	72.34	58.45	64.52	69.23	56.59	62.15
PairGCN w/o D2	85.79	72.03	78.20	71.35	58.85	64.36	68.18	56.83	61.85
PairGCN w/o D1&D2	86.44	68.13	76.10	73.19	57.14	64.06	70.27	55.35	61.81
PairGCN w/o GCN	87.31	66.93	75.72	73.73	56.00	63.57	70.90	54.23	61.37
PairGCN-BERT	88.57	79.58	83.75	79.07	69.28	73.75	76.92	67.91	72.02
PairGCN-BERT w/o D1	89.66	77.01	82.66	80.25	67.14	72.88	78.07	66.01	71.31
PairGCN-BERT w/o D2	87.91	78.93	82.99	77.79	68.69	72.69	75.17	67.16	70.66
PairGCN-BERT w/o D1&D2	88.39	77.50	82.45	78.21	67.87	72.49	75.68	66.46	70.59
PairGCN-BERT w/o GCN	89.64	75.18	81.60	79.08	65.46	71.43	76.77	64.39	69.85

Table 4: Ablation analysis on PairGCN and PairGCN-BERT.

$\leftrightarrow c^p_{3,3}$), and a $D2$ edge (i.e., $c^p_{3,1} \leftrightarrow c^p_{3,3}$). During propagation, the first path brings more information because of passing more nodes (e.g., $c^p_{3,2}$), and the second path is more straightforward.

5 Conclusion and Future Work

In this paper, we propose a novel end-to-end Pair Graph Convolutional Network (PairGCN) to extract pair-level contextual features for emotion-cause pair extraction. Experimental results indicate the capability of our PairGCN in capturing dependency among local neighborhood candidate pairs. In the future, we would like to tackle the problem of imbalanced data by reducing non-emotion-cause pairs.

Acknowledgments

We thank the anonymous reviewers for their helping comments. This work was partially supported by the National Key Research and Development Program of China (No. 2016YFB0501805) and the Chinese Universities Scientific Fund (No. 2018XD003).

References

Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and ChuRen Huang. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 179–187.

Ying Chen, Wenjun Hou, Xiyao Cheng, and Shoushan Li. 2018. Joint learning for emotion classification and emotion cause detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 646–651.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL*.

Kai Gao, Hua Xu, and Jiushuo Wang. 2015a. Emotion Cause Detection for Chinese Micro-blogs Based on ECOCC Model. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD)*, 3–14.

Kai Gao, Hua Xu, and Jiushuo Wang. 2015b. A Rule-based Approach to Emotion Cause Detection for Chinese Micro-blogs. *Expert Systems with Applications*, 42(9):4517–4528.

Deepanway Ghosal, Navonil Majumder, Soujanya Poria, Niyati Chhaya, and Alexander Gelbukh. 2019. DialogueGCN: A Graph Convolutional Neural Network for Emotion Recognition in Conversation. In *Proceedings of ACL*.

Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016. Event-Driven Emotion Cause Extraction with Corpus Construction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1639–1649, Austin, Texas.

Lin Gui, Jiannan Hua, Yulan Hec, Ruifeng Xua, Qin Lue, and Jiachen Du. 2017. A Question Answering Approach to Emotion Cause Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1593–1602, Copenhagen, Denmark.

Diederik P. Kingma and Jimmy Ba. 2015. ADAM: A Method for Stochastic Optimization. In *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*.

Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *Proceedings of ICLR*.

Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A Text-driven Rule-based System for Emotion Cause Detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 45–53, Los Angeles, California.

Weiyuan Li and Hua Xu. 2014. Text-based Emotion Classification Using Emotion Cause Extraction. *Expert Systems with Applications*, 41(4):503–512.

Xiangju Li, Kaisong Song, Shi Feng, Daling Wang, and Yifei Zhang. 2018. A Co-attention Neural Network Model for Emotion Cause Analysis with Emotional Context Awareness. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4752–4757.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed Representations of Words and Phrases and Their Compositionality. In *Advances in Neural Information Processing Systems (NIPS)*, pages 3111–3119.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on International Conference on Machine Learning (ICML)*, pages 807–814.

Alena Neviarouskaya and Masaki Aono. 2013. Extracting Causes of Emotions from Text. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 932–936.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. In *Journal of Machine Learning Research*, pages 1929–1958.

Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. 2019. Aspect-Level Sentiment Analysis Via Convolution over Dependency Tree. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, pages 5678–5687, Hong Kong, China.

Rui Xia and Zixiang Ding. 2019. Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1003–1012, Florence, Italy.

Bo Xu, Hongfei Lin, Yuan Lin, Yufeng Diao, Liang Yang, and Kan Xu. 2019. Extracting Emotion Causes Using Learning to Rank Methods from an Information Retrieval Perspective. *IEEE Access*.

Shuntaro Yada, Kazushi Ikeda, Keiichiro Hoashi, and Kyo Kageura. 2017. A Bootstrap Method for Automatic Rule Acquisition on Emotion Cause Extraction. In *IEEE International Conference on Data Mining Workshops*, 414–421.

Liang Yao, Chengsheng Mao, and Yuan Luo. 2019. Graph Convolutional Networks for Text Classification. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.

Xinyi Yu, Wenge Rong, Zhuo Zhang, Yuanxin Ouyang, and Zhang Xiong. 2019. Multiple Level Hierarchical Network-based Clause Selection for Emotion Cause Extraction. *IEEE Access*, 7(1):9071–9079.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph Convolution over Pruned Dependency Trees Improves Relation Extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium.