

# A Symmetric Local Search Network for Emotion-Cause Pair Extraction

Zifeng Cheng, Zhiwei Jiang\*, Yafeng Yin, Hua Yu, Qing Gu

State Key Laboratory for Novel Software Technology,

Nanjing University, Nanjing 210023, China

chengzff@smail.nju.edu.cn, {jzw, yafeng}@nju.edu.cn

huayu.yh@smail.nju.edu.cn, guq@nju.edu.cn

## Abstract

Emotion-cause pair extraction (ECPE) is a new task which aims at extracting the potential clause pairs of emotions and corresponding causes in a document. To tackle this task, a two-step method was proposed by previous study which first extracted emotion clauses and cause clauses individually, then paired the emotion and cause clauses, and filtered out the pairs without causality. Different from this method that separated the detection and the matching of emotion and cause into two steps, we propose a Symmetric Local Search Network (SLSN) model to perform the detection and matching simultaneously by local search. SLSN consists of two symmetric subnetworks, namely the emotion subnetwork and the cause subnetwork. Each subnetwork is composed of a clause representation learner and a local pair searcher. The local pair searcher is a specially-designed cross-subnetwork component which can extract the local emotion-cause pairs. Experimental results on the ECPE corpus demonstrate the superiority of our SLSN over existing state-of-the-art methods.

## 1 Introduction

Emotion cause analysis is a research branch of sentiment analysis and has gained increasing popularity in recent years (Lee et al., 2010; Gui et al., 2016; Xia and Ding, 2019; Xia et al., 2019). Its goal is to identify the potential causes that lead to the certain emotion. This is very useful in fields such as electronic commerce, where the sellers may concern about users' emotions towards the products as well as the causes of users' emotions.

Previous studies on emotion cause analysis mainly focus on the task of emotion cause extraction (ECE), which is usually formalized as a clause-level sequence labeling problem (Chen et al., 2010; Gui et al., 2016; Li et al., 2018; Ding et al., 2019; Xia et al., 2019; Yu et al., 2019; Fan et al., 2019). Given an annotated emotion clause, for each clause in the document, the goal of ECE task is to identify whether the clause is the corresponding cause. However, in practice, the emotion clauses are naturally not annotated, which may limit the application of the ECE task in real-world scenarios. Motivated by this, Xia and Ding (2019) first proposed the emotion-cause pair extraction (ECPE) task, which aims to extract all potential pairs of emotion and corresponding cause in a document. As shown in Figure 1, the example document has 17 clauses, among which, the emotion clauses are  $c_4$ ,  $c_{13}$ , and  $c_{17}$  (marked as orange), and their corresponding cause clauses are  $c_3$ ,  $c_{12}$ , and  $c_{15}$  (marked as blue). The goal of ECPE task is to extract all emotion-cause pairs:  $(c_4, c_3)$ ,  $(c_{13}, c_{12})$ , and  $(c_{17}, c_{15})$ .

The ECPE task is a new and more challenging task. To tackle this task, Xia and Ding (2019) proposed a two-step method, which has been demonstrated to be effective. In the first step, they extracted emotion clauses and cause clauses individually. In the second step, they used Cartesian product to pair the clauses and then used a logistic regression to filter out the emotion-cause pairs without causality. In this method, the detection of emotion and cause, and the matching of emotion and cause are separately implemented in two steps.

---

\* Corresponding Author

This work is licensed under a Creative Commons Attribution 4.0 International Licence. Licence details: <http://creativecommons.org/licenses/by/4.0/>.

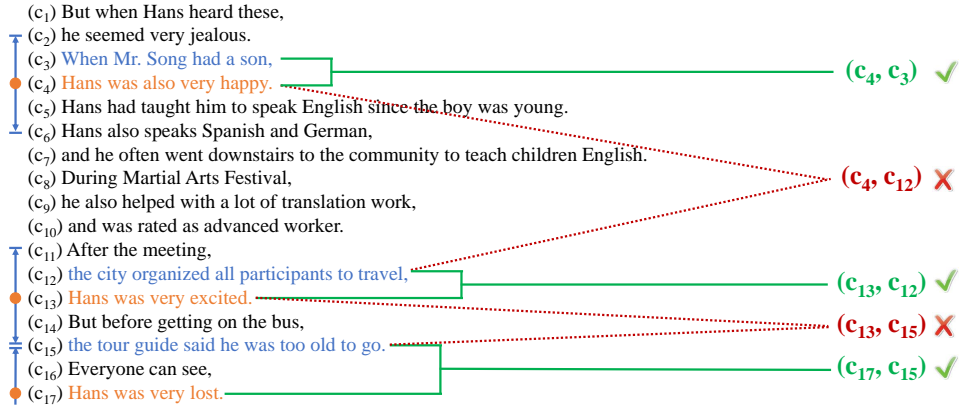


Figure 1: An example document from the ECPE corpus

However, when humans deal with the ECPE task, they usually consider the detection and matching problems at the same time. This is mainly achieved through the process of local search. For example, as shown in Figure 1, if a clause is detected as an emotion clause (e.g.,  $c_4$ ), humans will search its corresponding cause clause (i.e.,  $c_3$ ) within its local context (i.e.,  $c_2, c_3, c_4, c_5, c_6$ ). The advantage of local search is that the wrong pairs (e.g.,  $(c_4, c_{12})$ ) beyond the local context scope can be avoided. Additionally, when local searching the cause clause corresponding to the target emotion clause, humans not only judge whether the clause is a cause clause, but also consider whether it matches the target emotion clause. In this way, they can avoid extracting the pairs (e.g.,  $(c_{13}, c_{15})$ ) that are in the local context scope but mismatch. Similarly, when a cause clause is encountered, the corresponding emotion clause can also be searched within its local context scope.

Inspired by this local search process, we propose a Symmetric Local Search Network (SLSN) model. The model consists of two subnetworks with symmetric structures, namely the emotion subnetwork and the cause subnetwork. Each subnetwork consists of two parts: a clause representation learner and a local pair searcher (LPS). Among them, the clause representation learner is designed to learn the emotion or cause representation of a clause. The local pair searcher is designed to perform the local search of emotion-cause pairs. Specifically, the LPS introduces a local context window to limit the scope of context for local search. In the process of local search, the LPS first judges whether the target clause is emotion (cause), and then judges whether each clause within the local context window is the corresponding cause (emotion). Finally, SLSN will output the local pair labels (i.e., the labels of the target clause and the clauses within its local context window) for each clause in the document, from which we can get the emotion-cause pairs.

The main contributions of this work can be summarized as follows:

- We propose a symmetric local search network model, which is an end-to-end model and gives a new scheme to solve the ECPE task.
- We design a local pair searcher in SLSN, which allows simultaneously detecting and matching the emotions and causes.
- Experimental results on the ECPE corpus demonstrate the superiority of our SLSN over existing state-of-the-art methods.

## 2 Symmetric Local Search Network

In this section, we first present the task definition. Then, we introduce the SLSN model, followed by its technical details. Finally, we discuss the connection between the SLSN model and the previous two-step method.

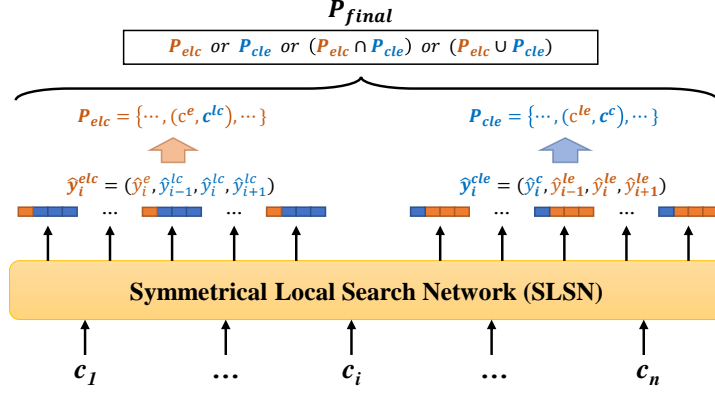


Figure 2: Overview of SLSN model

## 2.1 Task Definition

The task of emotion-cause pair extraction (ECPE) is first studied by Xia and Ding (2019). In the ECPE task, each document  $d$  in the dataset  $D$  consists of multiple clauses  $d = [c_1, c_2, \dots, c_n]$ . The clause with emotional polarity (such as happiness, sadness, fear, anger, disgust and surprise) is labeled as an emotion clause  $c^e$ . The clause that causes the emotion is called a cause clause  $c^c$ . The pair of emotion clause and its corresponding cause clause is called an emotion-cause pair  $(c^e, c^c)$ . The goal of ECPE task is to extract all emotion-cause pairs in  $d$ :

$$P = \{\dots, (c^e, c^c), \dots\}$$

Note that each document may contain several (at least one) emotion clauses, and each emotion clause may correspond to several (at least one) cause clauses. Besides, the emotion clause and its corresponding cause clause may be the same clause.

## 2.2 An Overview of SLSN

As shown in Figure 2, SLSN receives a sequence of clauses from a document as input and predicts the local pair labels for these clauses, which can be directly converted into the corresponding emotion-cause (E-C) pairs. For each clause  $c_i$ , SLSN predicts two types of local pair labels: E-LC label  $\hat{y}_i^{elc}$  and C-LE label  $\hat{y}_i^{cle}$ . The E-LC label  $\hat{y}_i^{elc}$  contains the emotion label (E-label)  $\hat{y}_i^e$  of the  $i$ -th clause and the local cause labels (LC-label)  $(\hat{y}_{i-1}^{lc}, \hat{y}_i^{lc}, \hat{y}_{i+1}^{lc})$  of the clauses near the  $i$ -th clause. Similarly, the C-LE label  $\hat{y}_i^{cle}$  contains the cause label (C-label)  $\hat{y}_i^c$  of the  $i$ -th clause and the local emotion labels (LE-label)  $(\hat{y}_{i-1}^{le}, \hat{y}_i^{le}, \hat{y}_{i+1}^{le})$  of the clauses near the  $i$ -th clause. Whether a clause is near the target clause is defined by the local context window, whose size is denoted as  $k$  (the case in Figure 2 is  $k = 1$ ). That is, for a target clause, the scope of its local context includes the previous  $k$  clauses, itself, and the following  $k$  clauses. Note that, both the E-LC label  $\hat{y}_i^{elc}$  and the C-LE label  $\hat{y}_i^{cle}$  can be converted into their corresponding emotion-cause (E-C) pairs. For example, the corresponding E-C pair of  $\hat{y}_i^{elc} = (1, 1, 0, 0)$  is  $(c_i, c_{i-1})$ , and the corresponding E-C pair of  $\hat{y}_i^{cle} = (1, 1, 0, 0)$  is  $(c_{i-1}, c_i)$ . We denote the E-C pair set corresponding to  $\hat{y}_i^{elc}$  as  $P_{elc}$ , and the E-C pair set corresponding to  $\hat{y}_i^{cle}$  as  $P_{cle}$ . Then the final E-C pair set of our method is the union of  $P_{elc}$  and  $P_{cle}$ . Of course,  $P_{elc}$ ,  $P_{cle}$  or the intersection of  $P_{elc}$  and  $P_{cle}$  is also an option for the final pair set.

## 2.3 Components of SLSN

As shown in Figure 3, SLSN contains two subnetworks, i.e., the emotion subnetwork referred as E-net which is mainly for the E-LC label prediction and the cause subnetwork referred as C-net which is mainly for the C-LE label prediction. E-net and C-net have similar structures in terms of word embedding, clause encoder, and hidden state learning. After the hidden state learning layer, E-net and C-net use two types of local pair searchers (LPS) with symmetric structures for the local pair label prediction. The local pair

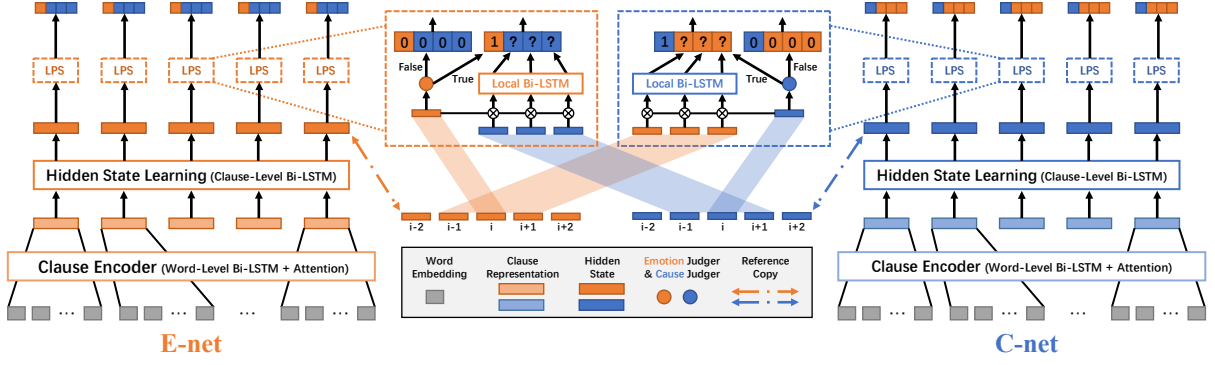


Figure 3: Framework of SLSN model

searcher is a specially designed cross-subnetwork module, which uses the hidden states of the clauses in both subnetworks for prediction. In the following, we introduce the components of SLSN in technical details.

### 2.3.1 Word Embedding

Before representing the clauses in the document, we first map each word in clauses into word embedding, which is a low-dimensional real-value vector. Formally, given a sequence of clauses  $d = [c_1, c_2, \dots, c_n]$ , the clause  $c_i = [w_i^1, w_i^2, \dots, w_i^{l_i}]$  consists of  $l_i$  words. We map each clause into its word-level representation  $v_i = [v_i^1, v_i^2, \dots, v_i^{l_i}]$ , where  $v_i^j$  is the word embedding of word  $w_i^j$ .

### 2.3.2 Clause Encoder

After word embedding, we use a Bi-LSTM layer followed by an attention layer as the clause encoder in both E-net and C-net to learn the representation of clauses. Formally, in E-net, given the word-level representation of the  $i$ -th clause  $v_i = [v_i^1, v_i^2, \dots, v_i^{l_i}]$  as the input, the word-level Bi-LSTM layer first maps it to the hidden states  $r_i = [r_i^1, r_i^2, \dots, r_i^{l_i}]$ . Then, the attention layer maps each  $r_i$  to the emotion representation of the clause  $s_i^e$  by weighting each word in the clause and then aggregating them through the following equations:

$$w_i^j = \tanh(W_w r_i^j + b_w) \quad (1)$$

$$a_i^j = \frac{\exp((w_i^j)^T u_s)}{\sum_t \exp((w_i^t)^T u_s)} \quad (2)$$

$$s_i^e = \sum_j a_i^j r_i^j \quad (3)$$

where  $W_w$ ,  $b_w$  and  $u_s$  are weight matrix, bias vector and context vector respectively.  $a_i^j$  is the attention weight of  $r_i^j$ . Similarly, in C-net, the cause representation of the  $i$ -th clause  $s_i^c$  is obtained using a clause encoder with a similar structure.

### 2.3.3 Hidden State Learning

After the clause encoder, we use a hidden state learning layer to learn the contextualized representation of each clause in the document. Formally, in E-net, given a sequence of emotion representations  $[s_1^e, s_2^e, \dots, s_n^e]$  as input, the clause-level Bi-LSTM layer is used to map it to a sequence of emotion hidden states  $[h_1^e, h_2^e, \dots, h_n^e]$ . Similarly, in C-net, the sequence of cause hidden states  $[h_1^c, h_2^c, \dots, h_n^c]$  is obtained from a sequence of cause representations.

### 2.3.4 Local Pair Searcher

After obtaining the two types of hidden states, we design two types of local pair searchers (LPS) with symmetric structures in E-net and C-net respectively, to predict the local pair labels of each clause.

In E-net, LPS predicts the E-LC label for each clause, which contains E-label and LC-label respectively. For the E-label prediction, LPS only uses the emotion hidden state of the clause. Formally, given

the emotion hidden state  $h_i^e$  of the  $i$ -th clause, LPS uses a softmax layer to predict its E-label  $\hat{y}_i^e$  through the following equation:

$$\hat{y}_i^e = \text{softmax}(W_e h_i^e + b_e) \quad (4)$$

where  $W_e$  and  $b_e$  are weight matrix and bias vector respectively.

For the LC-label prediction, there are two cases under consideration. If the predicted E-label of the  $i$ -th clause is false, the corresponding LC-label is a zero vector. Because it is unnecessary to predict LC-label. Otherwise, LPS predicts the LC-label for all the clauses within the local context of the  $i$ -th clause. We denote these clauses as local context clauses. Assuming that the local context window size is  $k = 1$  (the case in Figure 3), the local context clauses of the  $i$ -th clause are  $c_{i-1}$ ,  $c_i$ , and  $c_{i+1}$  respectively. For the LC-label prediction, both the emotion and cause hidden states of the clause are used. Formally, given the emotion hidden state  $h_i^e$  of the  $i$ -th clause and the cause hidden states  $[h_{i-1}^c, h_i^c, h_{i+1}^c]$  of the corresponding local context clauses, LPS first calculates an emotion attention ratio  $\lambda_j$  for each local context clause using the following formula:

$$\gamma(h_i^e, h_j^c) = h_i^e h_j^c \quad (5)$$

$$\lambda_j = \frac{\exp(\gamma(h_i^e, h_j^c))}{\sum_{j=i-k}^{i+k} \exp(\gamma(h_i^e, h_j^c))} \quad (6)$$

where  $\gamma(h_i^e, h_j^c)$  is an emotion attention function which estimates the relevance between the local cause and the target emotion. We choose the simple dot attention based on the experimental results (Luong et al., 2015). This emotion attention ratio  $\lambda_j$  is then used to scale the original cause hidden states as follows:

$$q_j^{lc} = \lambda_j \cdot h_j^c \quad (7)$$

where  $q_j^{lc}$  is the scaled cause hidden state of the  $j$ -th local context clause. The  $\otimes$  used in Figure 3 refers to Eq. (5), Eq. (6), and Eq. (7). We further use a local Bi-LSTM layer to learn the contextualized representation of each local context clause:

$$\vec{o}_j = \overrightarrow{LSTM}_{lc}(q_j^{lc}), j \in [i - k, i + k] \quad (8)$$

$$\overleftarrow{o}_j = \overleftarrow{LSTM}_{lc}(q_j^{lc}), j \in [i - k, i + k] \quad (9)$$

Finally, the LC-label  $\hat{y}_j^{lc}$  of the  $j$ -th local context clause is predicted through the following equation:

$$\hat{y}_j^{lc} = \text{softmax}(W_{lc} o_j + b_{lc}) \quad (10)$$

where  $o_j$  is the concatenation of  $\vec{o}_j$  and  $\overleftarrow{o}_j$ ,  $W_{lc}$  and  $b_{lc}$  are the weight matrix and bias vector respectively.

Similarly, in C-net, the LPS with symmetric structure in E-net is used to predict the C-LE label for each clause, which contains C-label  $\hat{y}_i^c$  and LE-label  $\hat{y}_i^{le}$  respectively.

## 2.4 Model Training

The SLSN model consists of two sub-networks, i.e., E-net and C-net. Given a sequence of clauses as input, the E-net is mainly used to predict their E-LC label, and the C-net is mainly used to predict their C-LE label. Thus, the loss of SLSN is a weighted sum of two components:

$$L = \alpha L^{elc} + (1 - \alpha) L^{cle} \quad (11)$$

where  $\alpha \in [0, 1]$  is a tradeoff parameter. Both  $L^{elc}$  and  $L^{cle}$  consist of two parts of the loss:

$$L^{elc} = \beta L^e + (1 - \beta) L^{lc} \quad (12)$$

$$L^{cle} = \beta L^c + (1 - \beta) L^{le} \quad (13)$$

where  $\beta \in (0, 1)$  is another tradeoff parameter.  $L^e$ ,  $L^{lc}$ ,  $L^c$  and  $L^{le}$  are the cross-entropy loss of the prediction of E-label  $\hat{y}_i^e$ , LC-label  $\hat{y}_i^{lc}$ , C-label  $\hat{y}_i^c$ , and LE-label  $\hat{y}_i^{le}$  respectively:

$$L^e = -\frac{1}{n} \sum_{i=1}^n \eta y_i^e \log(\hat{y}_i^e) \quad (14)$$

$$L^{lc} = -\frac{1}{p^e(2k+1)} \sum_{i=1}^n \mathbb{I}(\hat{y}_i^e = 1) \sum_{j=i-k}^{i+k} y_j^{lc} \log(\hat{y}_j^{lc}) \quad (15)$$

$$L^c = -\frac{1}{n} \sum_{i=1}^n \eta y_i^c \log(\hat{y}_i^c) \quad (16)$$

$$L^{le} = -\frac{1}{p^c(2k+1)} \sum_{i=1}^n \mathbb{I}(\hat{y}_i^c = 1) \sum_{j=i-k}^{i+k} y_j^{le} \log(\hat{y}_j^{le}) \quad (17)$$

where  $y_i^e$ ,  $y_j^{lc}$ ,  $y_i^c$  and  $y_j^{le}$  denote the ground-truth,  $\mathbb{I}(\cdot)$  is an indicator function,  $p^e$  and  $p^c$  denote the times that  $\mathbb{I}(\cdot)$  equals 1 in Eq. (15) and Eq. (17) respectively, and  $\eta$  is used to deal with the class imbalance problem.

## 2.5 Connection to the Two-Step Method

The two-step method for the ECPE task was proposed by Xia and Ding (2019), which first extracted the emotion clause set and cause clause set individually, and then generated emotion-cause pairs and finally filtered out irrelevant pairs. In their approach, Cartesian product was used to pair the two sets. While in our model, we use the local context window size  $k$  to control the scope of local pair search. Considering the extreme case of  $k = n$ , LPS treats all clauses in the document as the local context of the target clause. This is actually equivalent to making a Cartesian product to search emotion-cause pair globally. Therefore, in the extreme case  $k = n$ , our method is approximately an end-to-end version of the two-step method (Xia and Ding, 2019). In practice,  $k$  usually takes a much smaller value than  $n$ , which can reduce the complexity of the model while achieving better performance.

## 3 Experiments

### 3.1 Dataset and Experimental Settings

We evaluate our proposed model on a Chinese ECPE corpus (Xia and Ding, 2019) which is constructed based on the ECE corpus (Gui et al., 2016). In this paper, we use the same setting adopted by Xia and Ding (2019). The dataset is split into two parts, 90% for training, and the remaining 10% for testing. The results reported in the following experiments are an average of 10-fold cross-validation. We use Precision (P), Recall (R), and F1-score to measure the performance.

We use the word embedding provided by NLPCC which was pre-trained on a 1.1 million Chinese Weibo corpora with the word2vec toolkit (Mikolov et al., 2013) and the dimension of word embedding is 200. The number of hidden units of Bi-LSTM in this paper is set to be 100. The network is trained based on the Adam optimizer, where the mini-batch size and the learning rate are set to 32 and 0.005. The tradeoff parameters  $\alpha$ ,  $\beta$ , and the size of the local context window  $k$  are set to 0.6, 0.8, and 2 respectively. The parameter  $\eta$  is set to 5.

### 3.2 Comparison with Other Methods

We compare our model with the following three baseline methods proposed by Xia and Ding (2019): Indep, Inter-CE and Inter-EC. All these three methods are based on the two-step framework, and the differences of them mainly lie in the first step. Among them, Indep extracts emotion and cause individually, Inter-CE uses cause extraction to enhance emotion extraction, while Inter-EC uses emotion extraction to enhance cause extraction. For our model, we consider four variants of SLSN: SLSN-E, SLSN-C, SLSN-I, and SLSN-U, which correspond to the output of  $P_{elc}$ , the output of  $P_{cle}$ , the intersection of  $P_{elc}$  and  $P_{cle}$ , and the union of  $P_{elc}$  and  $P_{cle}$ , respectively.

	Emotion Extraction			Cause Extraction			Emotion-Cause Pair Extraction		
	Precision	Recall	F1-score	Precision	Recall	F1-score	Precision	Recall	F1-score
<b>Indep</b>	0.8375	0.8071	0.8210	0.6902	0.5673	0.6205	0.6832	0.5082	0.5818
<b>Inter-CE</b>	0.8494	<b>0.8122</b>	<b>0.8300</b>	0.6809	0.5634	0.6151	0.6902	0.5135	0.5901
<b>Inter-EC</b>	0.8364	0.8107	0.8230	0.7041	0.6083	0.6507	0.6721	0.5705	0.6128
<b>SLSN-E</b>	0.8580	0.7718	0.8118	0.7386	0.6246	0.6762	0.7090	0.6062	0.6529
<b>SLSN-C</b>	0.8563	0.7541	0.8014	0.7301	0.6205	0.6704	0.7087	0.5956	0.6465
<b>SLSN-I</b>	<b>0.8770</b>	0.7193	0.7896	<b>0.7696</b>	0.5866	0.6651	<b>0.7463</b>	0.5696	0.6454
<b>SLSN-U</b>	0.8406	0.7980	0.8181	0.6992	<b>0.6588</b>	<b>0.6778</b>	0.6836	<b>0.6291</b>	<b>0.6545</b>

Table 1: Comparisons with baselines on ECPE corpus. Averaged results over 10 runs are reported. The best results are in bold.

Case	Components of Local Pair Searcher		Emotion-Cause Pair Extraction		
	Attention Function	Local Encoder	Precision	Recall	F1-score
<b>SLSN-U</b>	Using Attention	Bi-LSTM	0.6836	0.6291	0.6545
<b>(a)</b>	Not Using Attention	Bi-LSTM	0.6322	0.6639	0.6462
<b>(b)</b>	Using Attention	Not Using Encoder	0.5549	0.5819	0.5631
<b>(c)</b>	Using Attention	FC	0.5531	0.6031	0.5730
<b>(d)</b>	Using Attention	CNN	0.6900	0.6045	0.6433
<b>(e)</b>	Using Attention	Transformer	0.6759	0.6198	0.6453

Table 2: Effect of the components in local pair searcher. Averaged results over 5 runs are reported.

As shown in Table 1, for the target ECPE task, our methods (i.e., SLSN-E, SLSN-C, SLSN-I, SLSN-U) all achieve better performance than the two-step methods (i.e., Indep, Inter-CE and Inter-EC) on F1-score, and SLSN-U achieves the best performance (0.6545 on F1-score). When compared with the best baseline method Inter-EC, SLSN-E, SLSN-C, SLSN-I, and SLSN-U achieve 0.0401, 0.0337, 0.0326, and 0.0417 improvement on F1-score, respectively. More specifically, by observing the results on emotion extraction and cause extraction, we can find that when compared with the baseline methods, our models perform better on cause extraction (increase about 0.02 on F1-score) and worse on emotion extraction (decrease about 0.02 on F1-score). This indicates that our model SLSN has no advantage in emotion extraction, and the good performance on ECPE task may be owed to the effectiveness of cause extraction or emotion-cause pairing.

### 3.3 Effect of Components in LPS

In this section, we explore how the components in LPS affect the performance of SLSN. We mainly study two components of LPS, i.e., the attention function and the local encoder. We conduct the experiments by ablating them from the model or substituting them with other components. The results are shown in Table 2.

By comparing SLSN-U and the case (a) in Table 2, we can find that the case using attention function outperforms the case not using attention function by about 0.0083 on F1-score. Similarly, by comparing SLSN-U and the case (b), we can find that the case using local encoder (Bi-LSTM) outperforms the case not using local encoder by about 0.091 on F1-score. This indicates that both the attention function and the local encoder are effective designs. For the local encoder, we also try to substitute Bi-LSTM with other kinds of transformation layers such as FC (i.e., Fully-Connected layer), CNN, and Transformer (Vaswani et al., 2017). Among these layers, we can find that Bi-LSTM achieves the best performance. Besides, CNN and transformer achieve similar performance which is only a little worse than Bi-LSTM (about 0.01 on F1-score). FC gets the worst performance which only brings a little improvement on the case not using encoder (about 0.01 on F1-score). This indicates that the context aware transformation layers (i.e., Bi-LSTM, CNN, and Transformer) are more suitable to be the local encoder, and Bi-LSTM is the best choice.

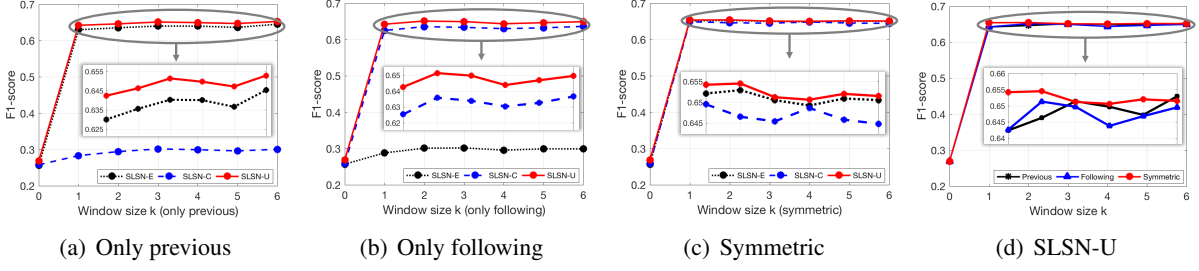


Figure 4: Effect of local context window

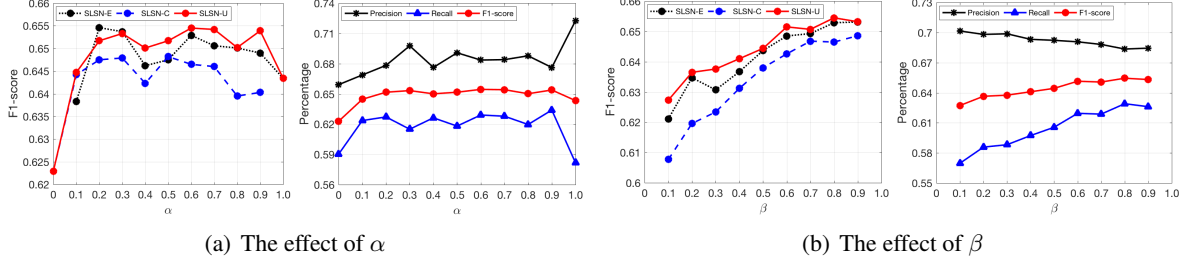


Figure 5: Effects of  $\alpha$  and  $\beta$

### 3.4 Effect of Local Context Window

In this section, we explore how the setting of local context window affects the performance of SLSN. We study three settings of local context window: only previous  $k$  (i.e., the target clause and its previous  $k$  clauses), only following  $k$  (i.e., the target clause and its following  $k$  clauses), and symmetric (i.e., the target clause, its previous  $k$  clauses, and its following  $k$  clauses). For each setting, we study its effect on the performance of two basic models (i.e., SLSN-E, SLSN-C) and the best model (i.e., SLSN-U). We conduct the experiments by varying  $k$  from 0 to 6 step by 1. The results are shown in Figure 4.

From Figure 4(a) and Figure 4(b), we can find that the performance of SLSN-C is poor under the setting of only previous  $k$  (about 0.3 on F1-score), while the performance of SLSN-E is poor under the setting of only following  $k$  (about 0.3 on F1-score). This implies that the cause clauses may tend to appear before the corresponding emotion clauses. From Figure 4(c), we can find that both SLSN-C and SLSN-E keep a good performance (about 0.63 on F1-score) under the setting of symmetric. This indicates that the setting of symmetric is more robust than other two settings. By observing the former three subfigures in Figure 4, we can find that SLSN-U keeps a stable performance (about 0.65 on F1-score), no matter which setting of local context window is adopted. This means that SLSN-U is more robust to the setting of local context window than SLSN-C and SLSN-E. From Figure 4(d), we can find that SLSN-U achieves the best performance under the setting of symmetric. This indicates that the best choice of the setting of local context window is symmetric. In addition, by observing the window size  $k$ , we can find that all the three models achieve a relatively stable performance as  $k$  varies, except for the case  $k = 0$ .

### 3.5 Effect of Tradeoff Parameters

In this section, we explore the effect of the tradeoff parameters  $\alpha$  and  $\beta$  on the performance of SLSN. For each parameter, we study its effect on the performance of three models (i.e., SLSN-E, SLSN-C, and SLSN-U). We conduct the experiments by varying  $\alpha$  from 0 to 1 step by 0.1, and varying  $\beta$  from 0.1 to 0.9 step by 0.1. The results are shown in Figure 5.

From the left subfigure of Figure 5(a), we can find that when  $\alpha = 0$  or  $\alpha = 1$ , the models get poor performance. When  $\alpha \in [0.1, 0.9]$ , the performance of SLSN-E and SLSN-C can be improved. This implies that both kinds of loss (i.e.,  $L^{elc}$  and  $L^{cle}$ ) are important to SLSN, and either E-net or E-net can benefit from each other. In most cases, SLSN-U achieves the best performance, and SLSN-E performs better than SLSN-C. From the right subfigure of Figure 5(a), we can find that the three metrics (i.e.,



Document	Ground-truth	Inter-EC	SLSN-U
Ms. Huang married a guy 20 years younger than her ( $c_1$ ). In order to avoid losing her wealth ( $c_2$ ), they have notarized their property before marriage ( $c_3$ ). However the man stole Ms. Huang’s money twice after marriage ( $c_4$ ), and Ms. Huang submitted a case to the court helplessly ( $c_5$ ).	( $c_5, c_4$ )	None	( $c_5, c_4$ )
Chen and his wife have two boys and one daughter ( $c_1$ ). The 18-year-old son works in Taizhong ( $c_2$ ). When he knew his mother had been killed by his father ( $c_3$ ), he was quite emotional ( $c_4$ ). Then he saw his younger brother and sister ( $c_5$ ), and they cried together ( $c_6$ ).	( $c_4, c_3$ )	( $c_4, c_3$ ), ( $c_4, c_5$ ), ( $c_6, c_5$ )	( $c_4, c_3$ )

Table 3: Two examples for the case study

precision, recall, and F1-score) of SLSN-U exhibit a steady trend when  $\alpha \in [0.1, 0.9]$ . This indicates that SLSN-U is robust to the setting of parameter  $\alpha$ .

From the left subfigure of Figure 5(b), we can find that when  $\beta \in [0.1, 0.9]$ , the performance of all three models exhibit an upward trend as  $\beta$  increases. This means that the prediction of E-label and C-label plays a more important role than the prediction of LC-label and LE-label in the training process of SLSN. Again, in most cases, SLSN-U achieves the best performance, and SLSN-E performs better than SLSN-C. From the right subfigure of Figure 5(b), we can find that the recall and F1-score of SLSN-U exhibit an upward trend and the precision of SLSN-U exhibits a downward trend as  $\beta$  increases. This implies that the LPS tends to extract more emotion-cause pairs as  $\beta$  increases.

### 3.6 Case Study

For the case study, we select two examples in the test dataset to demonstrate the effectiveness of our model. The ground-truth and the predicted results of Inter-EC and SLSN-U are given in Table 3.

For the first example, Inter-EC predicts None while SLSN-U predicts the correct emotion-cause pair ( $c_5, c_4$ ). For the Inter-EC method, it outputs the right pair only when the emotion clause set includes  $c_5$  and the cause clause set includes  $c_4$ . While for our method, we can extract emotion-cause pair according to emotion or cause clause. It is easier to establish a matching relationship.

For the second example, we can observe that many wrong answers are predicted by Inter-EC (e.g., ( $c_4, c_5$ ), ( $c_6, c_5$ )). Due to the use of Cartesian product operation, the connection between emotion clause and cause clause may be ignored, thus many irrelevant pairs will be introduced. It indicates that Cartesian product brings a lot of redundancy and the filter operation fails to filter out the irrelevant pairs. In our method, we can extract emotion-cause pair straightly in the local context window. So our method avoids the redundancy brought by Cartesian product. This is a reason that our method is better than Inter-EC method.

## 4 Related Work

Emotion cause analysis has been studied for about a decade (Lee et al., 2010; Gui et al., 2016; Xia and Ding, 2019). Previous studies on emotion cause analysis mainly focused on the emotion cause extraction (ECE) task (Ding et al., 2019; Xia et al., 2019; Fan et al., 2019). Recently, based on the ECE task, a new and more challenging task named emotion cause pair extraction (ECPE) was proposed (Xia and Ding, 2019).

The ECE task was first proposed by Lee et al. (2010) and was formalized as a word-level sequence labeling problem. But Chen et al. (2010) suggested that clause may be a more appropriate unit than word for detecting cause. Later, based on this idea, Gui et al. (2016) released a Chinese ECE corpus from a public SINA city news. In this corpus, the ECE task was defined as a clause-level sequence labeling problem, the objective of which is to predict the cause clauses in a document given the emotion. For the following studies on ECE task, this corpus has become a benchmark dataset. While early studies mainly adopted the rule-based methods (Chen et al., 2010; Gao et al., 2015; Gui et al., 2014) and machine learning methods (Ghazi et al., 2015) to deal with the ECE task, recent studies has begun to apply the deep learning methods to this task (Gui et al., 2017; Li et al., 2018; Chen et al., 2018; Ding et al., 2019; Yu et al., 2019; Li et al., 2019; Fan et al., 2019; Xia et al., 2019).

Although the ECE task is valuable in practice, its application in real-world scenarios is limited due to the reason that the emotion clauses are naturally not annotated. Considering this situation, Xia and

Ding (2019) proposed the ECPE task and released a corresponding ECPE dataset based on the ECE corpus. To tackle the task, they proposed a two-step method, which first extracted emotions and causes individually by multi-task framework, and then got the emotion-cause pairs by pairing and filtering.

## 5 Conclusion and Future Work

In this paper, we propose a symmetric local search network (SLSN) to perform end-to-end emotion-cause pair extraction. SLSN can straightly extract the emotion-cause pair through a process of local search. This is realized by designing a special component, i.e., local pair searcher, which allows simultaneously detecting and matching the emotions and causes. Experimental results on the ECPE corpus demonstrate the effectiveness of our model.

In the future, we will consider to further improve the performance of emotion extraction and cause extraction by employing more powerful pre-trained encoder (e.g., BERT (Devlin et al., 2019)) or designing some auxiliary tasks to utilize extra knowledge. Besides, we will further explore the process of local pair search, and seek for more advanced implementations of the local pair searcher.

## Acknowledgements

This work is supported by National Natural Science Foundation of China under Grant Nos. 61906085, 61802169, 61972192, 41972111; JiangSu Natural Science Foundation under Grant No. BK20180325; the Second Tibetan Plateau Scientific Expedition and Research Program under Grant No. 2019QZKK0204. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.

## References

- Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *COLING 2010, 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*, 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 Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019*, pages 4171–4186. Association for Computational Linguistics.
- Zixiang Ding, Huihui He, Mengran Zhang, and Rui Xia. 2019. From independent prediction to reordered prediction: Integrating relative position and global label information to emotion cause identification. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019*, pages 6343–6350.
- Chuang Fan, Hongyu Yan, Jiachen Du, Lin Gui, Lidong Bing, Min Yang, Ruifeng Xu, and Ruibin Mao. 2019. A knowledge regularized hierarchical approach for emotion cause analysis. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP*, pages 5613–5623.
- Kai Gao, Hua Xu, and Jiushuo Wang. 2015. Emotion cause detection for chinese micro-blogs based on ecoc model. In *Advances in Knowledge Discovery and Data Mining - 19th Pacific-Asia Conference, PAKDD*, pages 3–14.
- Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *Computational Linguistics and Intelligent Text Processing*, pages 152–165.
- Lin Gui, Li Yuan, Ruifeng Xu, Bin Liu, Qin Lu, and Yu Zhou. 2014. Emotion cause detection with linguistic construction in chinese weibo text. In *Natural Language Processing and Chinese Computing*, pages 457–464.
- 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, EMNLP*, pages 1639–1649.

- Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. A question answering approach for emotion cause extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP*, pages 1593–1602.
- 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. Association for Computational Linguistics.
- 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.
- Xiangju Li, Shi Feng, Daling Wang, and Yifei Zhang. 2019. Context-aware emotion cause analysis with multi-attention-based neural network. *Knowl.-Based Syst.*, 174:205–218.
- Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015*, pages 1412–1421. The Association for Computational Linguistics.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, pages 3111–3119.
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017*, pages 5998–6008.
- Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 1003–1012.
- Rui Xia, Mengran Zhang, and Zixiang Ding. 2019. RTHN: A rnn-transformer hierarchical network for emotion cause extraction. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, pages 5285–5291.
- 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*, 7:9071–9079.