Applied Intelligence (2023) 53:3519–3529
https://doi.org/10.1007/s10489-022-03637-7

An effective multi-task learning model for end-to-end emotion-cause pair extraction

Chenbing Li[1] · Jie Hu[1] · Tianrui Li[1] · Shengdong Du[1] · Fei Teng[1]

Accepted: 14 April 2022 / Published online: 1 June 2022

Abstract

Emotion-cause pair extraction (ECPE), as an extended research direction of emotion cause extraction, aims to extract emotion and its corresponding causes for a given document. Previous methods solved this problem in a two-stage fashion. Nevertheless, these methods suffered from the problem of error propagation. Moreover, there exists the problem of label imbalance for the ECPE task. In order to solve the above problems, in this paper, we propose a novel end-to-end multi-task learning model which contains a shared module and a task-specific module to simultaneously perform emotion extraction, cause extraction, and emotion-cause pair extraction. The above three tasks share the shallow sharing module, and the shared information among mining tasks is realized to achieve mutual benefit. Then each task generates task-specific features and completes the corresponding tasks in the task-specific module. In addition, we propose a sampling-based method to construct the training set for the ECPE task to alleviate the problem of label imbalance and enable our model to focus on extracting the pairs with the corresponding emotion-cause relationship. Experimental results show that our model outperforms many strong baselines with 75.48%, 75.57%, and 75.03% in P, R, and F1 score, respectively.

Keywords Emotion-cause pair extraction · End-to-end · Multi-task learning · Label imbalance

1 Introduction

Emotion cause extraction (ECE), as a branch of emotion analysis, aims at discovering the corresponding causes for a certain emotion expressed in a document. This task was first defined as a word-level sequence labeling problem [1]. Afterward, Chen et al. [2] found that emotion causes are often expressed in phrases or sentences, and then they changed the extraction granularity of this task from word-level to clause-level. Gui et al. [3] treated the ECE task as a clause-level binary classification problem, which aims to detect clause-level causes towards a certain emotion expressed in the document. Following the formulation of the ECE task in [3], many methods [4–8] have been proposed to address the ECE task. However, these models have an obvious disadvantage. These emotions should be manually pointed out before performing the ECE task, which significantly limits their practical application scenarios.

To solve the problem in the above task, Xia et al. [9] proposed a new task called emotion-cause pair extraction (ECPE), which seeks to discover all emotion-cause pairs in a document. A specific example is shown in Fig. 1. The document contains 4 clauses. Clause c_4 is an emotion clause, and its corresponding cause clause is clause c_3. The goal of the ECPE is to find all latent emotion-cause pairs, e.g., (c_4, c_3).

Meanwhile, Xia et al. [9] put forward a two-stage model to extract emotion-cause pairs. In the first stage, the emotion clauses and cause clauses are extracted. In the second stage, a classification model is used to extract the emotion-cause

✉ Jie Hu
jiehu@swjtu.edu.cn

Chenbing Li
ChenbingLi@my.swjtu.edu.cn

Tianrui Li
trli@swjtu.edu.cn

Shengdong Du
sddu@swjtu.edu.cn

Fei Teng
fteng@swjtu.edu.cn

[1] School of Computing and Artificial Intelligence, Southwest Jiaotong University, Chengdu 611756, China

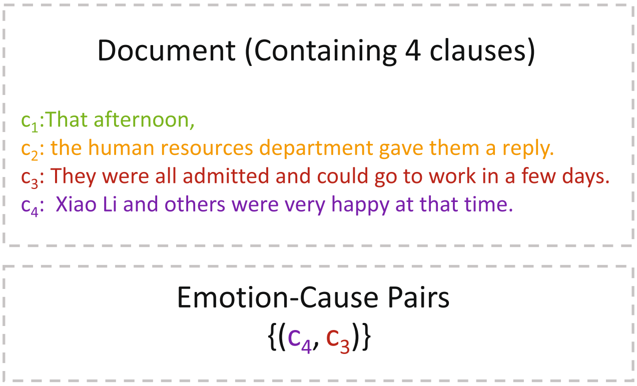

Document (Containing 4 clauses)
c1:That afternoon,
c2: the human resources department gave them a reply.
c3: They were all admitted and could go to work in a few days.
c4: Xiao Li and others were very happy at that time.
Emotion-Cause Pairs
{(c4, c3)}

Fig. 1 An example used to illustrate the emotion-cause pair extraction task

pairs from all candidate pairs by applying the Cartesian product of the emotion clauses and the cause clauses. However, this model has an obvious shortcoming that the error generated at the first stage will be propagated to the second stage. Therefore, many efforts have been paid to handle the ECPE task in an end-to-end manner [10–17].

Most of these end-to-end methods use multi-task learning to complete ECPE task. But some of them [10, 14, 17] cannot share feature information among different tasks. Therefore, we consider employing a unified framework to jointly perform the emotion clause extraction (EE), the cause clause extraction (CE), and the ECPE tasks. Specially, the three tasks use the same shallow structure to share information between tasks. Then, each specific task extracts the characteristics for the task and completes the corresponding task.

In addition, there is a label imbalance problem for the ECPE task, that is to say, only a few of these pairs are emotion-cause pairs, and most of them are not emotion-cause pairs. This means that the class distribution in the training dataset is uneven, which is called label imbalance. Unfortunately, almost all existing approaches treat all pairs into the training set and then classify positive cases, i.e., emotion-cause pairs.

Based on the above considerations, we propose a new multi-task learning framework for ECPE task (ECPE-MTL) to solve the EE, CE, and ECPE tasks simultaneously. It is composed of two modules: a shared module and a task-specific module. The shared module is responsible for generating clause representations, which fully mines the shared information among tasks. Based on the shared module, there is a task-specific module that contains three independent parts. Each part is responsible for generating the task-specific representation and performing the corresponding task, i.e., EE, CE, and ECPE. Generally, the corresponding causes often appear in the context of this specific emotion, and vice versa. Thus, towards the label imbalance of emotion-cause pairs, we construct the training set by sampling a subset from all candidate pairs according to the absolute distance between two clauses in the candidate pairs. The main contributions of our work can be summarized as follows:

1. We design a multi-task learning framework to jointly perform EE, CE, and ECPE with the aim of gaining mutual benefit.
2. To handle the imbalanced class distribution in the ECPE task, we design a strategy to construct a training set, which is conceptually simple and effective.
3. The experimental results on the benchmark dataset demonstrate that our method achieves a better performance on the ECPE task.

2 Related work

2.1 Emotion cause extraction

Gui et al. [3] employed a multi-kernel SVM classifier to perform ECE task based on the publicly available dataset constructed by themselves. Afterwards, this dataset becomes a benchmark dataset for the ECE research. With the development of deep representation learning and the extensive application of the attention mechanism, a set of deep learning based methods [4–8, 18] were proposed for the ECE task. These methods seek to model text sequence information and the relationship between emotional words and clauses to improve emotion cause extraction. For example, Li et al. [5] held that the context of the specific emotion is also a valuable clue to find the corresponding causes, and designed a co-attention module to make use of the context of emotion for the ECE task. Yu et al. [6] believed that the relationships among clauses are also important and proposed a hierarchical framework, which not only takes semantic information between emotion description and clause into consideration, but also considers the relationships among clauses. In addition to the content of the document, Ding et al. [7] found that the label information and relative position information between emotion description and clause are important for emotion cause extraction. Xia et al. [8] employed the Transformer [21] as the clause encoder to model the relations between the clauses and further extracted emotion causes. Hu et al. [18] employed graph convolutional networks to encode the semantic and structure information of the clause and achieved superior performance on extracting emotion causes. However, the ECE task has an obvious disadvantage: emotions should be labeled manually before extracting emotion causes. As a result, Xia et al. [9] proposed the ECPE task.

2.2 Emotion-cause pair extraction

In recent years, extensive efforts have been paid to address the ECPE task in an end-to-end fashion. For example, Wei et al. [10] emphasized the importance of the relationships between clauses. They adopted a ranking perspective to deal with this task and selected pairs with high confidence as the emotion-cause pairs. Ding et al. [12] designed a 2D Transformer to model the interaction between pairs. They integrated the emotion-cause pair representation learning, emotion-cause pair interaction, and emotion-cause pair prediction into a unified framework to complete emotion-cause pair extraction. Additionally, Ding et al. [13] pointed out that extracting the causes without specifying the emotion is unreasonable, and vice versa. Thus, they proposed two dual frameworks for the ECPE task. The first framework takes every clause in the document as an emotion clause, and then employs multi-label learning to extract the corresponding cause clauses in the context of the emotion clause, named EMLL. The second framework regards every clause in the document as a cause clause and then employs multi-label learning to extract the corresponding emotion clauses in the context of the cause clause, named CMLL. Yuan et al. [11] devised a novel tagging scheme, and proposed a sequence labeling model based on Bi-LSTM to extract emotion-cause pairs. Tang et al. [14] believed that the current research failed to detect the relationship between emotion detection (ED) and ECPE, and thus proposed a multi-task learning framework for ED and ECPE tasks. Wu et al. [15] proposed a multi-task learning neural network to perform emotion extraction, cause extraction, and emotion-cause relation classification tasks jointly, which explores the interactions among these tasks. Song et al. [16] tackled the ECPE task as predicting directional links between emotion and cause. They designed a multi-task learning model to perform ECPE tasks with the help of auxiliary tasks, i.e., EE and CE. Yu et al. [19] proposed a mutually auxiliary multi-task model, which adds two auxiliary tasks to build the interaction between emotion extraction and cause extraction. However, the label imbalance issue had not been solved in the above methods. In this paper, we propose an end-to-end multi-task learning model which employs a sampling-based strategy to construct the training set to alleviate the above problem.

3 Approach

3.1 Problem definition

The input of the ECPE task is a document composed of multiple clauses $D = \left[c_1, c_2, \dots, c_{|D|}\right]$, where $|D|$ is the number of clauses in it. Every clause c_i $(i = 1, 2, \dots, |D|)$ consists of several words $c_i = \left[w_1^i, w_2^i, \dots, w_{|c_i|}^i\right]$, where $|c_i|$ is the length of clause c_i. The target of the ECPE task is to find all potential emotion-cause pairs in the document D:

$$P = \left\{\dots, (c_j^{emo}, c_j^{cau}), \dots\right\}, \qquad (1)$$

where $\left(c_j^{emo}, c_j^{cau}\right)$ is the jth emotion-cause pair, c_j^{emo} and c_j^{cau} are emotion clause and the corresponding cause clause, respectively.

3.2 Overall architecture

We propose a novel multi-task learning framework to perform the EE, CE, and ECPE tasks simultaneously (shown in Fig. 2), which mainly consists of two modules. The module below, termed the shared module, is to generate clause representations. The module above is a task-specific module, which includes three independent parts. These three parts generate task-specific representation based on the outputs of the shared module, and then perform the corresponding task, i.e., EE, CE, and ECPE tasks.

3.3 Shared module

3.3.1 Clause encoder

BERT [20] is a bidirectionally pre-trained language model, which shocked the deep learning world when it led to excellent improvement on the downstream task in NLP. Therefore, our model generates clause representations based on BERT. Specifically, given a document $D = \left[c_1, c_2, \dots, c_{|D|}\right]$ consisting of $|D|$ clauses and each clause $c_i = \left[w_1^i, w_2^i, \dots, w_{|c_i|}^i\right]$ containing $|c_i|$ words, the input of BERT is composed of c_i and two additional tokens, formulated as $\left([CLS], w_1^i, w_2^i, \dots, w_{|c_i|}^i, [SEP]\right)$. The $[CLS]$ token is added at the beginning of each clause, where its final hidden state can be used as a semantic representation of the whole clause. The $[SEP]$ token is added at the ending of each clause to distinguish other clauses. We take the final hidden state of $[CLS]$ as raw clause representation $h_i \in \mathbb{R}^{d_B}$ for clause c_i.

Then we employ one fully connected layer for dimension reduction. Finally, we obtain all clause hidden representations of the document D and formulate them as $\left[h_1, h_2, \dots, h_{|D|}\right] \in \mathbb{R}^{d_b \times |D|}$, which will be fed into the Transformer layer.

3.3.2 Learning correlations between clauses with Transformer

As we know, clauses in a document do not exist independently, and the correlations between the clauses are

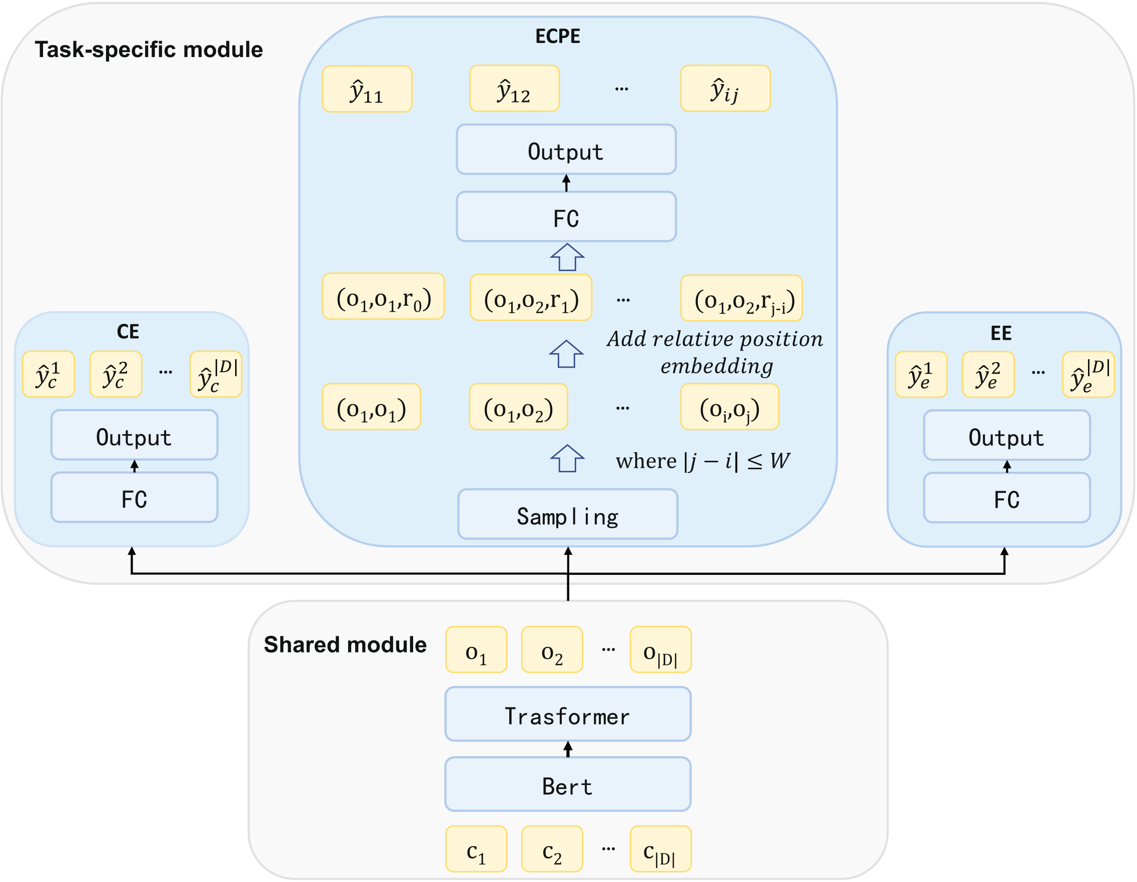


Fig. 2 The overall architecture of ECPE-MTL. The shared module is responsible for generating task invariant clause embeddings $[o_1, o_2, ..., o_{|D|}]$, where $|D|$ is the number of clauses in the document and o_i is the embedding of clause i. The three blue modules in the task-specific module are responsible for the EE, CE, and ECPE tasks. The left module CE in the task is to extract the cause clauses, and $\hat{y}_c^i$ is the probability of clause c_i being a cause clause. The middle module ECPE is to extract emotion-cause pairs, and $\hat{y}_{ij}$ denotes the probability of candidate pairs (c_i, c_j) being an emotion-cause pair. The right module EE is to extract the emotion clauses, and $\hat{y}_e^i$ is the probability of clause c_i being an emotion clause

helpful information. Generally, grasping contextual cues can help us understand the current clause better. Therefore, we apply an encoder module of Transformer [21] to generate an updated clause representation by incorporating other clauses' information into the current clause, which enables us to understand the current clause from the perspective of the document. The standard encoder of the Transformer includes a stack of N identical layers, where each layer has two sub-layers. The first sublayer is a multi-head self-attention layer, and the second sublayer is a fully connected feed-forward network.

multi-head self-attention layer One-head attention is the foundation of single-head attention, and in our setting, we adopt single-head attention. Concretely, for each clause c_i, we first feed its clause representation $h_i \in \mathbb{R}^{db}$ into three distinct fully connected layers to produce the query, key, and value vectors, represented as q_i, k_i, and v_i as follows:

$$q_i = W_q h_i, \tag{2}$$

$$k_i = W_k h_i, \tag{3}$$

$$v_i = W_v h_i, \tag{4}$$

where $W_q \in \mathbb{R}^{d_q \times d_b}$, $W_k \in \mathbb{R}^{d_k \times d_b}$ and $W_v \in \mathbb{R}^{d_v \times d_b}$ are trainable parameters, and d_k, d_q, d_v are the dimension of key, query, and value vectors, respectively.

After that, the query vector q_i of clause c_i does dot product with all key vectors k_j $(j = 1, 2, \ldots, |D|)$ to produce a score vector $Score_i \in \mathbb{R}^{|D|}$ as follows:

$$Score_i = [Score_{i,1}, Score_{i,2}, \ldots, Score_{i,|D|}]^\top, \tag{5}$$

$$Score_{i,j} = q_i^T \cdot k_j, j = 1, 2, \ldots, |D|. \quad (6)$$

Finally, we normalize score vector $Score_i$ to get the attention weights vector $A_i \in \mathbb{R}^{|D|}$ and get an output vector $z_i \in \mathbb{R}^{d_v}$ by calculating the weighted sum of all the value vectors, where z_i is an updated clause representation for clause c_i:

$$A_i = softmax(\frac{Score_i}{\sqrt{d_k}}), \quad (7)$$

$$z_i = \sum_j A_{i,j} v_j. \quad (8)$$

Intuitively, we employ self-attention to map all input clause embeddings into an updated clause embedding that holds the learned information of the whole document.

fully connected feed-forward network layer The attention sublayer is then followed by a fully connected feed-forward network sublayer:

$$e_i = W_2(ReLU(W_1 z_i + b_1)) + b_2, \quad (9)$$

where $W_1 \in \mathbb{R}^{d_{ff} \times d_v}$, $W_2 \in \mathbb{R}^{d_v \times d_{ff}}$, $b_1 \in \mathbb{R}^{d_{ff}}$, and $b_2 \in \mathbb{R}^{d_v}$ are trainable parameters.

To help the Transformer training, we add a residual connection followed by layer normalization at the output of above each sublayer:

$$o_i = LayerNorm(x_i + Sublayer(x_i)), \quad (10)$$

where x_i is the input of Sublayer and Sublayer(x_i) is the output of the sublayer.

As noted above, the standard encoder of Transformer is composed of N identical layers. We take the output of the previous layer as the input of the next layer:

$$H^{(l+1)} = O^{(l)}, \quad (11)$$

where l denotes the index of encoder layers.

Finally, the encoder of Transfomer outputs a set of clause embeddings represented as $\left[o_1^{(N)}, o_2^{(N)}, \ldots, o_{|D|}^{(N)}\right]$, and we formulate them as $\left[o_1, o_2, \ldots, o_{|D|}\right]$.

3.4 Task-specific module

3.4.1 Multi-task setting

Since ECPE task is related to the EE and CE tasks, we want to apply multi-task learning to bring improvement on extracting emotion-cause pairs with the help of auxiliary tasks, i.e., the EE and CE tasks. As with some previous work in [22], we generate task-specific representations for each task. Specifically, since the extraction granularity of these tasks is clause-level, the shared module generates clause representations. Then the three parts in the task-specific module all share the outputs of the shared module and generate task-specific features for the desired tasks.

In detail, upon the clause representations $[o_1, o_2, \ldots, o_{|D|}]$ output by shared module, we use three different fully connected layers to generate three task-specific feature vectors for EE, CE, and ECPE.

3.4.2 Emotion clause extraction and cause clause extraction

Given a document $D = \left[c_1, c_2, \ldots, c_{|D|}\right]$, EE aims to predict whether the clause c_i $(i = 1, 2, \ldots, |D|)$ is an emotion clause, and CE aims to predict whether clause c_i $(i = 1, 2, \ldots, |D|)$ is a cause clause. For each clause c_i, we feed its clause representation $o_i \in \mathbb{R}^{d_v}$ into two distinct fully connected layers to get the task-specific clause representations h_e^i for the EE task and h_c^i for the CE task:

$$h_e^i = W_e o_i + b_e, \quad (12)$$

$$h_c^i = W_c o_i + b_c, \quad (13)$$

where W_e, $W_c \in \mathbb{R}^{d_h \times d_v}$ and b_e, $b_c \in \mathbb{R}^{d_h}$ are trainable parameters.

After obtaining the task-specific clause representations h_e^i and h_c^i, we feed h_e^i into a connected layer followed by a logistic function $\sigma(\cdot)$ to predict the probability of clause c_i being an emotion clause. Similarly, we feed h_c^i into another connected layer followed by a logistic function $\sigma(\cdot)$ to predict the probability of clause c_i being an emotion cause clause. The formulas are as follows:

$$\hat{y}_e^i = \sigma\left(\hat{W}_e h_e^i + \hat{b}_e\right), \quad (14)$$

$$\hat{y}_c^i = \sigma\left(\hat{W}_c h_c^i + \hat{b}_c\right), \quad (15)$$

where $\hat{W}_e \in \mathbb{R}^{1 \times d_h}$, $\hat{b}_e \in \mathbb{R}$, $\hat{W}_c \in \mathbb{R}^{1 \times d_h}$, and $\hat{b}_c \in \mathbb{R}$ are trainable parameters.

3.4.3 Emotion-cause pair extraction

For the ECPE task, an intuitive method is to take all candidate pairs as the training set. However, in most cases, there is only one pair with the corresponding emotion-cause relationship in a document. That means there exists the problem of class imbalance. Moreover, according to the dataset constructed by Gui et al. [3], emotion clauses appear mostly in the context of their corresponding cause clause and vice versa. Based on the above analysis, we sample a subset from all candidate pairs and take this subset as the training set.

Concretely, if the absolute distance between clause c_i and clause c_j is less than or equal to a specific positive value W, we treat it as a training sample for ECPE. Consequently, we construct the training set $\mathcal{P}$ for ECPE:

$$\mathcal{P} = \left\{\left(c_i, c_j\right) | \, |j - i| \leq W\right\}, \; i, j = 1, 2, \ldots, |D|. \quad (16)$$

For each clause pair candidate $\left(c_i, c_j\right) \in \mathcal{P}$, we construct its representation via concatenating three vectors, i.e., the clause embedding $o_i \in \mathbb{R}^{d_v}$ of clause c_i, the clause embedding $o_j \in \mathbb{R}^{d_v}$ of clause c_j, and their relative position embedding $r_{j-i} \in \mathbb{R}^{d_r}$ encoding the distance between clause c_i and c_j. Then we use a fully connected layer followed by a $ReLU$ function to get a task-specific pair representation p_{ij} for ECPE:

$$h_{ij} = \left(o_i \oplus o_j \oplus r_{j-i}\right), \tag{17}$$

$$p_{ij} = \mathrm{Re}LU\left(W_p h_{ij} + b_p\right), \tag{18}$$

where the adopted relative position embedding r_{j-i} is the same as in RANKCP [10], d_r represents the dimension of relative position embedding, $W_p \in \mathbb{R}^{(2\times d_v+d_r)\times(2\times d_v+d_r)}$ and $b_p \in \mathbb{R}^{(2\times d_v+d_r)}$ are trainable parameters.

Then pair representation p_{ij} is fed into softmax classifier to get the $\hat{y}_{ij}$, which denotes the probability of candidate pair (c_i, c_j) being an emotion-cause pair:

$$\hat{y}_{ij} = \sigma\left(\hat{W}_p p_{ij} + \hat{b}_p\right), \tag{19}$$

where $\hat{W}_p \in \mathbb{R}^{1\times(2\times d_v+d_r)}$ and $\hat{b}_p \in \mathbb{R}$ are trainable parameters.

In the testing phase, we aim to predict whether the candidate pair in $\mathcal{P}' = \left\{\ldots, \left(c_i, c_j\right), \ldots\right\}$ $(i, j = 1, 2\ldots, |D|)$ is an emotion-cause pair. Specifically, for each candidate pair $\left(c_i, c_j\right)$ in $\mathcal{P}'$, we first construct its pair representation by concatenating the representations of c_i, c_j and their distance embedding r_{j-i}. But for those pairs whose relative distance is larger than W, we set r_w as its position embedding. For those pairs whose relative distance is less than -W, we set $r_{\text{-}w}$ as its position embedding. Then we feed its representation into the ECPE-part of the task-specific module to get the probability y_{ij} of $\left(c_i, c_j\right)$ of being an emotion-cause pair. Finally, we get $\{\ldots, \hat{y}_{ij}, \ldots\}$ and select the candidate pair with the highest probability as the emotion-cause pair.

In addition, there may be multiple emotion-cause pairs in some documents. In order to deal with this kind of problem, we take the same approach as in RANKCP [10]. Concretely, we select the top$-N$ candidate pairs $\{p_1, p_2, \ldots, p_N\}$ from $\mathcal{P}'$ according to their probabilities of being an emotion-cause pair and take p_1 as an emotion-cause pair by default. For the rest $p = (c_i^1, c_j^2) \in \{p_2, \ldots, p_N\}$, if its probability is larger than a threshold η and the clause c_i^1 contains sentiment word according to a sentiment lexicon [23], we extract it as an emotion-cause pair.

3.5 Objective function

As shown in (20), (21) and (22), $\mathcal{L}_e$ is the loss of the EE task, $\mathcal{L}_c$ is the loss of the CE task, and $\mathcal{L}_p$ is the loss of the ECPE task:

$$\mathcal{L}_e = -\sum_k \left(y_e^k \log\left(\hat{y}_e^k\right) + \left(1-y_e^k\right)\log\left(1-\hat{y}_e^k\right)\right), \tag{20}$$

$$\mathcal{L}_c = -\sum_k \left(y_c^k \log\left(\hat{y}_c^k\right) + \left(1-y_c^k\right)\log\left(1-\hat{y}_c^k\right)\right), \tag{21}$$

$$\mathcal{L}_p = -\sum_{\forall(c_i, c_j)\in\mathcal{P}} \left(y_{ij}\log\left(\hat{y}_{ij}\right) + \left(1-y_{ij}\right)\log\left(1-\hat{y}_{ij}\right)\right). \tag{22}$$

Since our model trains EE, CE, and ECPE tasks jointly, the objective function is the combination of cross-entropy loss of these tasks with the L_2-norm regularization term, which is formulated as follows:

$$\mathcal{L} = \mathcal{L}_e + \mathcal{L}_c + \mathcal{L}_p + \lambda\,\mathrm{reg}\,\|\theta\|_2, \tag{23}$$

where θ is the parameters set for L_2-norm regularization and λ is a coefficient for L_2-norm regularization.

4 Experiments

4.1 Dataset

We evaluate our ECPE-MTL model on the benchmark dataset published by Xia et al. [9], which is also the only dataset for the ECPE task. This dataset is constructed based on an open dataset for emotion cause extraction released by Gui et al. [3]. The statistical information about the benchmark dataset is shown in Table 1. It is noticed that nearly 90% of documents have only one emotion-cause

Table 1 Statistics of benchmark dataset for ECPE

	Number	Percentage
# of documents	1945	100%
# of documents with 1 emotion-cause pair	1746	89.7%
# of documents with over 1 emotion-cuase pairs	199	10.3%
# of emotion-cause pairs	2167	100%
# of emotion-cause pairs with absolute distance equal to 0	511	23.6%
# of emotion-cause pairs with absolute distance equal to 1	1342	61.9%
# of emotion-cause pairs with absolute distance equal to 2	224	10.3%
# of emotion-cause pairs with absolute distance equal to 3	50	2.3%
# of emotion-cause pairs with absolute distance large than 3	40	1.9%

pair. Moreover, the ratio of emotion-cause pairs where the relative distance between emotion clause c_i and cause clause c_j less than 3 accounts for 95.8%.

4.2 Evaluation metrics

Following previous work, we adopt the P, R, and F1 scores as evaluation metrics for the ECPE task:

$$P = \frac{\sum correct\_pairs}{\sum predicted\_pairs}, \tag{24}$$

$$R = \frac{\sum correct\_pairs}{\sum actual\_pairs}, \tag{25}$$

$$F1 = \frac{2 \times P \times R}{P + R}, \tag{26}$$

where $predicted\_pairs$ is the number of pairs predicted as emotion-cause pairs, $correct\_pairs$ is the number of pairs predicted as emotion-cause pairs correctly, and $autual\_pairs$ is the number of actual emotion-cause pairs in the dataset.

Similarly, we use the same evaluation metrics in [3] to evaluate the performance of our model on the EE and CE tasks.

4.3 Experimental setup

Our model is trained based on the Adam optimizer, where the batch size is set to 4, and the learning rate is set to 2e-5. In our objective function, the L_2 regularization coefficient λ is set to 1e-5. We adopt BERT-Chinese as the clause encoder in our work. The dimension of clause representations d_b is set to be 200. In the Transformer, the number of layers N of the encoder module is set to 1. The dimensions of query, key, value vectors $\left(d_q, d_k, d_v\right)$ are all set to 200, and the dimension d_{ff} of the hidden states is set to 400. The relative distance W is set to 4. Similarly, we set relative position embedding to 50. The parameter η we set is 0.5. In our experiment, we divide our data into 10 parts and employ 10-fold cross-validation. The average result of ten folds is recorded as the final result.

4.4 Baselines

To prove the effectiveness of our ECPE-MTL model on the ECPE task, we compared it with the following state-of-the-art methods.

Indep [9] is a two-stage method. It extracts all emotion clauses and cause clauses at the first step. In the second step, emotion-cause pairs are extracted based on the results of the first step.

Inter-CE/Inter-EC [9] are different from Indep in the first stage. Inter-CE utilizes the result of causes prediction to help emotions prediction. Inter-EC utilizes the prediction results of emotions to predict the emotion causes. The second step of Inter-CE and Inter-EC is the same as Indep.

RANKCP [10] tackles the ECPE task from a ranking perspective, which emphasizes inter-clause modeling and enhances the pair representations. RANKCP/BERT is based on BERT embeddings.

LAE-MANN [14] emphasizes the connection between EE and ECPE. It employs a multi-level attention module to model word-level interaction between two clauses in the candidate pair. LAE-MANN/BERT is based on BERT embeddings.

MTNECP [15] performs EE, CE, and ECPE tasks jointly and exploits the interaction between these tasks.

E2EECPE [16] is a multi-task learning model that jointly executes EE, CE, and ECPE tasks. It regards EE and CE as auxiliary tasks to improve the performance of ECPE.

MAM-SD [19] is a mutually auxiliary multi-task model that aims to model the mutual interaction between emotion extraction and cause extraction to improve the performance of the ECPE task.

ECPE-tagging [11] is a tagging method. It tackles ECPE as a tagging task and builds a novel tagging scheme.

ECPE-MLL [13] integrates two joint frameworks named CMLL and EMLL to solve the ECPE task. CMLL assumes that each clause in the document is an emotion-oriented clause and then finds the corresponding cause clauses in its context. EMLL and CMLL are the same in general, but the steps are opposite. The final result is obtained by integrating the results of CMLL and EMLL. ECPE-MLL/BERT is based on BERT embeddings.

4.4.1 Results on emotion-cause pair extraction

The experimental results are shown in Table 2. Obviously, our ECPE-MTL model achieves the best performance in terms of F1 score on ECPE. In addition, the P score and R score are higher than most models, which proves the effectiveness of ECPT-MTL. Secondly, we compare our model with RANKCP/BERT because we use the same method to extract sentiment-cause pairs. Overall, our model improves the F1 score and P of ECPE tasks by 1.43% and 4.29%, respectively. Besides, in terms of performance metrics, ECPE-MLL/BERT and

Table 2 The experimental results on emotion-cause pair extraction

Models	Emotion-Cause Pair Ext.			Emotion Extraction			**Cause Extraction**		
	P	R	F1	P	R	F1	P	R	F1
Indep	0.6832	0.5082	0.5818	0.8375	0.8071	0.8210	0.6902	0.5673	0.6205
Inter-CE	0.6902	0.5135	0.5901	0.8494	0.8122	0.8300	0.6809	0.5634	0.6151
Inter-EC	0.6721	0.5705	0.6128	0.8364	0.8107	0.8230	0.7041	0.6083	0.6507
RANKCP	0.6698	0.6546	0.6610	0.8703	0.8406	0.8548	0.6927	0.6743	0.6824
LAE-MANN	0.6990	0.5960	0.6440	0.8810	0.7810	0.8260	-	-	-
MTNECP	0.6828	0.5894	0.6321	0.8662	0.8393	0.852	0.7400	0.6378	0.6844
E2EECPE	0.6491	0.6195	0.6315	0.8552	0.8024	0.8275	0.7048	0.6159	0.6581
MAM-SD	0.6963	0.5799	0.6320	0.8554	0.8141	0.8339	0.7202	0.6375	0.6751
ECPE tagging/BERT	0.7243	0.6366	0.6776	0.8196	0.7329	0.7739	0.7490	0.6602	0.7018
ECPE-MLL/BERT	**0.7700**	0.7235	**0.7452**	0.8608	**0.9191**	0.8886	0.7382	**0.7912**	0.7630
RANKCP/BERT	0.7119	**0.7630**	0.7360	**0.9123**	**0.8999**	**0.9057**	**0.7461**	**0.7788**	0.7615
LAE-MANN/BERT	0.7110	0.6070	0.6550	0.8990	0.8000	0.8470	-	-	-
ours	**0.7548**	**0.7557**	**0.7503**	**0.9093**	0.8922	**0.9004**	**0.7769**	0.7739	**0.7749**

The top 2 results are shown in bold. The model with suffix /BERT indicates that BERT is used for word embeddings

RANKCP/BERT are currently the best two models. It can be clearly observed that ECPE-MLL/BERT achieves the highest score in metric P, but it performs less well in metric R, which indicates that it extracts are fewer positive cases (i.e., emotion-cause pairs) than RANKCP/BERT and our model. Toward the model MAM-SD, our model improves 5.85%, 17.58%, 11.83% on P, R and F1 score. The main reason is that it is still a two-stage model, which exists the problem of error propagation. LAE-MANN/BERT performs ED and ECPE jointly. Compared with it, our model improves 4.38%, 14.87%, 9.53% on P, R, and F1 score. Our model exceeds it a lot on R in that we employ a sampling method for constructing pairs before executing the ECPE task, which helps the model focus on finding the pairs with the emotion-cause relationship. Especially, our model also performs well on EE and CE. We attribute such high performance to the multi-task setting. Based on the clause representation output by the shared module, each task generates task-specific representation, which includes helpful information for the current task.

4.4.2 Results on emotion cause extraction

Before ECPE was proposed, ECE tasks had been widely discussed and studied in previous work. Therefore, we compare our model with some ECE models, including a traditional machine learning method (Multi-kernel [3]) and several deep learning based methods (MemNet [4], CANN [5], PAE-DGL [7], and RTHN [8]). It should be noted that these models need to label emotions manually before performing the ECE task. But in our model, we don't need any annotation in advance. RHTN-APE and CANN-E are the models which remove emotion annotations during the training phase.

As seen in Table 3, our ECPE-MTL model performs better than these models, which relies on emotion annotations to extract emotion causes. We think it can be attributed to two reasons. On the one hand, the CE task benefits from multi-task learning because our approach performs EE, CE, and ECPE jointly. On the other hand, we utilize BERT to generate powerful clause embeddings.

4.5 Ablation studies

In this section, we conduct several ablation experiments to verify the effects of several components in our approach.

Table 3 Comparison of the performance of our ECPE-MTL model with other models on emotion cause extraction

Models	Emotion Cause Ext.			
	P	R	F1	emotion annotations
Multi-kernel	0.6588	0.6927	0.6752	✓
MemNet	0.7076	0.6838	0.6955	✓
CANN	0.7721	0.6891	0.7266	✓
PAE-DGL	0.7619	0.6908	0.7242	✓
RHTN	0.7697	0.7662	0.7677	✓
CANN-E	0.4826	0.3160	0.3797	✗
RTHN-APE	0.5800	0.5618	0.5694	✗
RANKCP	0.7461	**0.7788**	0.7618	✗
ours	**0.7769**	0.7739	**0.7749**	✗

Table 4 Effectiveness of multi-task learning

Approachs	Emotion-Cause Pair Ext.		
	P	R	F1
ECPE-MTL w/o aux	0.6667	0.7321	0.6974
ECPE-MTL	**0.7458**	**0.7557**	**0.7503**

4.5.1 Effectiveness of multi-task learning

To explore the effect of auxiliary tasks, we conduct an ablation experiment on ECPE-MTL. Concretely, we remove $\mathcal{L}_e + \mathcal{L}_c$ in our objective function and train our model with only $\mathcal{L}_p$, which we name ECPE-MTL $w/o\ aux$. The results are shown in Table 4. We notice that the performance of ECPE-MTL $w/o\ aux$ drops a lot compared with ECPE-MTL, which demonstrates that ECPE actually benefits from the joint learning of the three tasks.

4.5.2 Effectiveness of transformer

We apply the Transfomer to learn the correlations between clauses, which enables our model to understand the current clause from the perspective of the document. In addition, we seek to use the graph attention network to model the correlations between clauses. We conduct an ablation study by designing the following ECPE-MTL variant:

ECPE-MTL w/o Transformer+GAT It's a model that we replace the Transformer with Graph Attention Network [24] to learn the correlations between clauses.

The results are shown in Table 5. Compared with ECPE-MTL w/o Transformer+GAT, ECPE-MTL performs better on the P, R, and F1 score. This is intuitive that Transformer does better at learning the correlations between clauses.

4.5.3 Effectiveness of constructing the training set

Towards ECPE-MTL, we construct the training set for the ECPE task by selecting candidate pairs (c_i, c_j) $(i, j = 1, 2, \ldots, |D|)$ with relative distance $|j - i|$ less than or equals to a specific value W instead of taking all candidate pairs as the training set. We explore the effectiveness of the training set construction method by an ablation study.

Table 5 Effectiveness of Transformer. We employ Graph Attention Network to learn the correlations between clauses

Approachs	Emotion-Cause Pair Ext.		
	P	R	F1
ECPE-MTL w/o Transformer+GAT	0.7365	0.7534	0.7445
ECPE-MTL	**0.7458**	**0.7557**	**0.7503**

Table 6 Effectiveness of constructing training set

Approachs	Emotion-Cause Pair Ext.		
	P	R	F1
ECPE-MTL w/o Sampling	0.7277	0.7258	0.7260
ECPE-MTL	**0.7458**	**0.7557**	**0.7503**

ECPE-MTL w/o Sampling In this model, we do not use the sampling strategy, but directly take all candidate pairs into the training set.

From Table 6, we can find that ECPE-MTL performs better than ECPE-MTL w/o sampling in P, R, and F1 score.

It proves that our sampling-based method to construct the training set is effective, because it alleviates the problem of label imbalance effectively and enables our model to focus on finding the candidate pairs with the emotion-cause relationship. We construct the training set according to hyperparameter W. We further explore the influence of different values of hyperparameter W on our model, and the results are shown in Fig. 3. Obviously, our model achieves the best performance when W is set to 4.

5 Conclusion

For the ECPE task, the existing works usually tackle the problems in a two-step fashion, and the problem of label imbalance for the ECPE task is ignored. To solve the above shortcomings, in this paper, we proposed an end-to-end model that employs multi-task learning to jointly perform the EE, CE, and ECPE tasks. In addition, targeting the

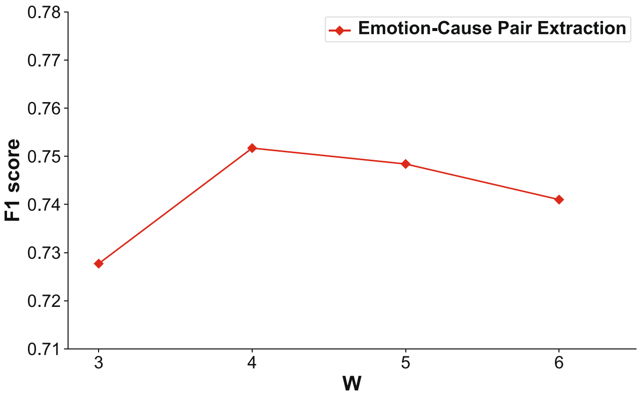


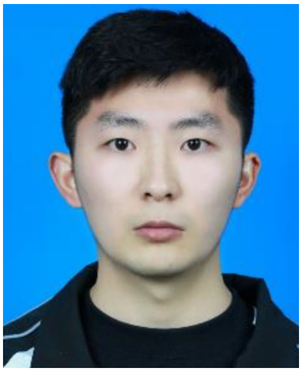
Fig. 3 The influence of hyperparameter W. We construct the training set of ECPE according to W. If the absolute distance between the two clauses in a candidate pair is less than W, we treat it as a training sample

problem of imbalanced classes distribution for the ECPE task, we proposed a sampling-based method to construct the training set. The experimental results showed that our multi-task learning model and the sampling-based method to construct the training set are effective. In future work, we will work in the following two aspects. Firstly, we will attempt to design a unified framework that extracts emotion firstly and then extract its corresponding causes around the context of emotion and vice versa, which is also more in line with the human thinking process. Secondly, because usually the emotions and causes are expressed by several words rather than the whole clause, we will consider using more fine-grained extraction, such as span-level extraction rather than clause-level extraction, to discover emotions and causes.

Acknowledgements This work is supported by the National Key Research and Development Program of China under Grant number: 2020AAA0105101, the National Natural Science Foundation of China (No. 61976182), and the Sichuan Key R&D project (Nos. 2022YFH0020, 2021YFG0136).

References

1. Lee SYM, Chen Y, Huang CR (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. Association for Computational Linguistics, pp 45-53
2. Chen Y, Lee SYM, Li S, Huang CR (2010) Emotion cause detection with linguistic constructions. In: Proceedings of the 23rd international conference on computational linguistics (COLING), pp 179–187
3. Gui L, Wu D, Xu R, Lu Q, Zhou Y (2016) Event-driven emotion cause extraction with corpus construction. In: Proceedings of the 2016 conference on empirical methods in natural language processing (EMNLP). Association for Computational Linguistics, pp 1639–1649
4. Gui L, Hu J, He Y, Xu R, Lu Q, Du J (2017) A question answering approach for emotion cause extraction. In: Proceedings of the 2017 conference on empirical methods in natural language processing (EMNLP), pp 1593–1602
5. Li X, Song K, Feng S, Wang D, Zhang Y (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), pp 4752–4757
6. Yu X, Rong W, Zhang Z, Ouyang Y, Xiong Z (2019) Multiple level hierarchical network-based clause selection for emotion cause extraction. IEEE Access 7:9071–9079
7. Ding Z, He H, Zhang M, Xia R (2019) From independent prediction to reordered prediction: Integrating relative position and global label information to emotion cause identification. In: Proceedings of the AAAI conference on artificial intelligence (AAAI), vol 33, pp 6343–6350
8. Xia R, Zhang M, Ding Z (2019) RTHN: A RNN-Transformer hierarchical network for emotion cause extraction. In: Proceedings of the 28th international joint conference on artficial intelligence (IJCAI), pp 5285–5291
9. Xia R, Ding Z (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 (ACL). Association for Computational Linguistics, pp 1003–1012
10. Wei P, Zhao J, Mao W (2020) Effective inter-clause modeling for endto-end emotion-cause pair extraction. In: Proceedings of the 58th annual meeting of the association for computational linguistics (ACL), pp 3171–3181
11. Yuan C, Fan C, Bao J, Xu R (2020) Emotion-cause pair extraction as sequence labeling based on a novel tagging scheme. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 3568–3573
12. Ding Z, Xia R, Yu J (2020) ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In: Proceedings of the 58th annual meeting of the association for computational linguistics (ACL), pp 3161–3170
13. Ding Z, Xia R, Yu J (2020) End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 3574–3583
14. Tang H, Ji D, Zhou Q (2020) Joint multi-level attentional model for emotion detection and emotion-cause pair extraction. Neurocomputing 409:329–340
15. Wu S, Chen F, Wu F, Huang Y, Li X (2020) A multi-task learning neural network for emotion-cause pair extraction. In: Proceedings of the 24th european conference on artificial intelligence (ECAI), pp 1–8
16. Song H, Song D (2021) An end-to-end multi-task learning to link framework for emotion-cause pair extraction. In: Proceedings of the 2021 international conference on image, video processing and artificial intelligence (ICPAI), vol 12076, pp 13–21
17. Chen Y, Hou W, Li S, Wu C, Zhang X (2020) End-to-end emotion-cause pair extraction with graph convolutional network. In: Proceedings of the 28th international conference on computational linguistics (COLING), pp 198–207
18. Hu G, Lu G, Zhao Y (2021) FSS-GCN: A graph convolutional networks with fusion of semantic and structure for emotion
19. Yu J, Liu W, He Y, Zhang C (2021) A mutually auxiliary multitask model with self-distillation for emotion-cause pair extraction. IEEE Access 9:26811–26821
20. Devlin J, Chang MW, Lee K, Toutanova K (2019) 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), vol 1, pp 4171–4186
21. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Kaiser L, Polosukhin I (2017) Attention is all you need. In: Proceedings of the 31st international conference on neural information processing systems (NIPS), pp 6000–6010
22. Zhang C, Li Q, Song D, Wang B (2020) A multi-task learning framework for opinion triplet extraction. In: Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP), pp 819–828
23. Wang SM, Ku LW (2016) ANTUSD: A large Chinese sentiment dictionary. In: Proceedings of the 10th international conference on language resources and evaluation (LREC), pp 2697–2702
24. Veličković P, Cucurull G, Casanova A, Romero A, Liò P, Bengio Y (2018) Graph Attention Networks. International conference on learning representations (ICLR)



Chengbing Li born in1998, is now a postgraduate at the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His main research interests include natural language processing and deep learning.

Jie Hu born in 1978, is an Associate professor at the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His research interests include machine learning and data mining.

Tianrui Li born in 1969, is a Professor at the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His research interests include big data, data mining, granular computing, and rough sets. He is a fellow of the IRSS and a senior member of the ACM and IEEE.

Shengdong Du born in 1981, is an Associate professor at the School of Computing and Artificial Intelligence, Southwest Jiaotong University. His research interests include data mining and deep learning.

Fei Teng born in 1984, is an Associate professor at the School of Computing and Artificial Intelligence, Southwest Jiaotong University. Her research interests include cloud computing and industrial big data mining. She is a member of the ACM and IEEE.