

Leveraging Two-Stream Cause-Effect Relation for Emotion-Cause Analysis

Anonymous ACL submission

Abstract

Emotion Cause Analysis (ECA) is a task to analyze corresponding causes for certain emotions expressed in text, which heavily depends on the context as the model needs to find the deep cause-effect relations between emotions and their causes. Previous research typically focused on extracting emotions first and then their corresponding causes, or vice versa. However, these approaches fail to integrate these two streams of thought into a unified model, so we propose a novel two-stream reasoning model to unify them for better performance. We leverage discourse connectives as bridges between these two streams, incorporating their discourse information to reveal cause-effect relations and enhance the reasoning ability of our model. Further, we employ the connectives predicted by ChatGPT to help our model achieve better results, and our research demonstrates that our model achieves SOTA results in ECA and proves the superiority of our model.

1 Introduction

Emotion-cause analysis (ECA) aims to extract emotional expressions and identify the reasons behind the emotions, which is helpful to the application of human-computer interaction, companionship, depression treatment, etc. As an example shown in Table 1, C_2 and C_3 are corresponding cause clauses of emotion clause C_4 . In this work, we mainly focus on the Emotion-Cause Pair Extraction (ECPE) (Xia and Ding, 2019) task, which intends to extract all possible emotion-cause pairs consisting of emotions and their corresponding causes.

Previous methods on ECA can be divided into two categories, i.e., clause pairing matrix methods and sequence labeling methods. The clause pairing matrix methods identify the emotion-cause pairs from a matrix of all clause combinations (Ding et al., 2020a,b; Wei et al., 2020), causing a sparse matrix and high computational cost. The sequence

C_1	yesterday morning	-
C_2	(because) a policeman visited the old man with the lost money	Cause 1
C_3	and (because) told him that the thief was caught	Cause 2
C_4	(so) the old man was very happy	Emotion
C_5	and deposited the money in the bank	-
Emotion Extraction Result		C_4
Cause Extraction Result		C_2, C_3
ECPE Result		$(C_4, C_2), (C_4, C_3)$

Table 1: An example text with the connectives predicted by ChatGPT and its results of ECA. Words in “()” represent predicted implicit connectives. Blue ones are connectives marking reasons and red ones are connectives marking results. These connectives are not allowed to coexist simultaneously, so we only consider one situation at a time.

labeling methods (Chen et al., 2020a; Yuan et al., 2020) explore some novel tagging schemes for identifying emotions and causes, but cannot well model the relations between emotions and their corresponding causes.

Though these works have made progress in ECA, they to some extent ignore that ECA in essence is a causal reasoning problem with consideration of emotional cause-effect relations. The cause and effect are in essence the two sides of a coin (Humphreys, 2005), so we need to explore “what is the cause of a given emotional effect” and “what is the emotional effect of a given cause” simultaneously. To this end, we design a two-stream reasoning model including emotion-cause stream (ECS) and cause-emotion stream (CES) as shown in Figure 1(a). These two streams share parameters for identifying emotions, causes, and emotion-cause pairs, which can be seen as a multi-task learning

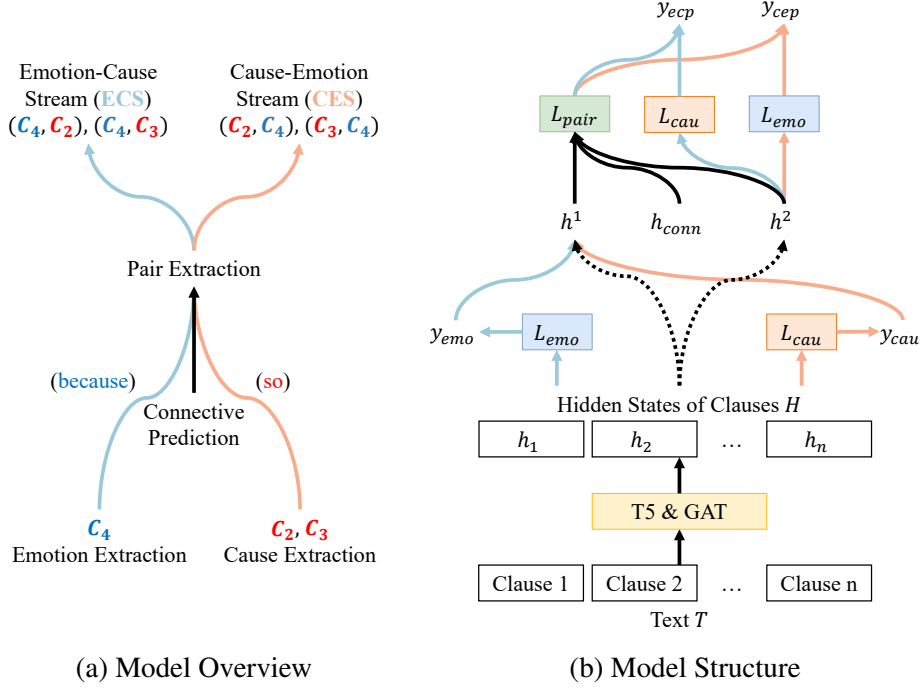


Figure 1: Model architecture. We use the example shown in Table 1 to describe our model. Cyan lines and orange lines denote **emotion-cause stream (ECS)** and **cause-emotion stream (CES)**, respectively. Modules with the same color and name share parameters with each other. L_{emo} , L_{cau} , L_{pair} denote linear layer for emotion, cause and pair classification, respectively. y_{emo} , y_{cau} , y_{ecp} , y_{cep} denote probability vector of emotions, causes, emotion-cause pairs, and cause-emotion pairs respectively. h_{conn} denotes the hidden state of connective. h^1 , h^2 are the hidden states of the emotion clause and candidate cause clause for ECS, and of the cause clause and candidate emotion clause for CES. Three lines pointing to h^1 means using predictions y_{emo} or y_{cau} to extract corresponding hidden states of predicted emotion clauses or cause clauses from overall hidden states H . The dotted line pointing from hidden states H to h^2 denotes the process of sequentially choosing the hidden state of each clause as the hidden state of a candidate clause.

process.

To enhance the causal reasoning capability of our two-stream model, we leverage predicted discourse connectives to serve as indicators with discourse information to assist in the identification of cause-effect relations. This approach draws inspiration from previous research that discourse relations can be classified with more than 94% accuracy from the discourse connectives alone (Pitler and Nenkova, 2009). Table 1 shows one example: with “because” or “so” predicted by ChatGPT, which highlights the links between emotions and causes, the cause-effect relations are obvious.

Experiments on the ECPE benchmark dataset (Xia and Ding, 2019) verify the effectiveness of our method, which exhibits good causal capability for the ECA task and outperforms other baselines. The main contributions of this paper can be summarized as follows:

- To our best knowledge, we are the first to consider the ECA task as an emotional reasoning

task and concentrate on the cause-effect relations to extract emotion-cause pairs.

- We propose a novel two-stream reasoning model that considers emotion-to-cause and cause-to-emotion relations in a unified way and intuitively utilizes discourse connectives predicted by ChatGPT to strengthen its analysis ability of cause-effect relations in the ECA task.
- The experiment results show that our method incorporating ChatGPT has achieved SOTA results and outperforms original ChatGPT results on the ECA task. It demonstrates the potential of employing reasoning methods to achieve better performance in the ECA task.

2 Methodology

2.1 Model Overview

Given a text with n clauses $\{C_1, \dots, C_n\}$, where C_i represents the i -th clause. ECPE aims to identify

each emotion-cause pair (C_{emo}, C_{cau}) , where the clause C_{cau} includes the reason causing the emotion occurring in clause C_{emo} . For this task, we design a two-stream approach, as shown in Figure 1, which unifies the emotion-cause stream (ECS) and the cause-emotion stream (CES).

As shown in Figure 1(b), our proposed method is a four-stage process, which sequentially consists of aggregation layers to obtain hidden states, joint linear layers L_{emo} and L_{cau} for emotion extraction or cause extraction, a pairing and connective prediction module, and joint linear layer L_{pair} for pair extraction.

First, we leverage aggregation layers built up with a T5 model (Raffel et al., 2020) and a graph attention network (GAT) (Veličković et al., 2018) to obtain the aggregated hidden states of clauses H in text T . The T5 model is utilized to get hidden states efficiently and GAT combines graph neural networks and attention mechanisms to enable adaptive aggregation on graph structures and therefore could build up the interaction among clauses and fuse contextual information.

Second, with them, for emotion-cause stream (ECS) and cause-emotion stream (CES), we use L_{emo} and L_{cau} to obtain probability vectors y_{emo} , y_{cau} of emotions and causes, respectively, and extract possible emotion and cause clauses. Our method leverages these joint linear layers to assist in predicting true emotion-cause pairs in the final stage.

In the third stage, the extracted clauses are paired up with other candidate clauses, and our T5 or ChatGPT module predicts their discourse connectives and gets the hidden states h_{conn} of the connectives.

In the final stage, our model concatenates the hidden states of paired clauses and the predicted connectives to judge which pairs are true emotion-cause pairs or cause-emotion pairs using L_{pair} with the help of the cause extraction linear layer L_{cau} or emotion extraction linear layer L_{emo} , where we consider all these linear classification modules as joint linear layers. Our method shares parameters on the T5 model, GAT, and joint linear layers L_{emo} , L_{cau} , and L_{pair} as a unified two-stream framework.

2.2 Pairing and Connective Prediction

We discovered that about two-thirds of the emotion-cause pairs do not contain any connectives in the Chinese ECPE benchmark dataset (Xia and Ding, 2019), indicating that it is necessary to explore an effective way to predict connectives in order

to leverage their discourse information. We show how to build up pairs and predict connectives in the emotion-cause stream (ECS) as a demonstration. After getting the extracted emotion clauses with linear layer L_{emo} , for each one of them, we pair up this emotion clause with each clause in the same text. These pairs have a format like (C_{emo}, C_{cand}) , where C_{emo} denotes emotion clause and C_{cand} denotes possible candidates cause clause.

For connectives prediction, We utilize ChatGPT in the evaluation and a (Raffel et al., 2020) T5 masked language model in the training, considering its ability to construct bidirectional context relationships. For the T5 module, we insert the special masking [MASK] tokens or sentinel tokens and estimate the probability of connectives using the probability vocabulary of T5 masked language models.

$$\begin{aligned}
 P_{conn}(c_i|C_{emo}, C_{cand}) & \\
 &= P_{T5}([\text{MASK}] = c_i|C_{pair})/P_{sum}, \\
 P_{sum} &= \sum_i P_{T5}([\text{MASK}] = c_i|C_{pair}), \\
 C_{pair} &= (C_{emo}, [\text{MASK}] C_{cand}),
 \end{aligned}$$

where c_i represents each possible connective, and P_{conn} , P_{T5} denote the probability of connectives and T5 masked language model likelihood, respectively. We choose the connective $conn$ with the highest probability among all c_i as our predicted connective.

But for some pairs, causal connectives may already exist and we need only to extract them, therefore we set some rules for this situation; see Appendix A. These connectives revealing cause-effect relations enable our model to get more discourse information to deal with the pair classification work. Also, we do not intend these newly predicted connectives to affect the results directly, so we use the same model and leverage the hidden states of connectives.

$$h_{conn} = T5(C_{emo}, conn C_{cand})|_{index},$$

where $index$ represents position index of connective and h_{conn} represents the hidden state of connective.

2.3 Aggregation Layers

To address the ECA task, we need to obtain hidden states of clauses with aggregated contextual representation, and our method utilizes a T5 (Raffel

et al., 2020) model, a clause-level attention layer, and a graph attention network (GAT) (Veličković et al., 2018) to do so. We first use the T5 model to get hidden states of all tokens in the text T . Then our model employs a clause-level attention layer to build up the hidden states of clauses. As the final step, we leverage a graph attention network (GAT) to build up the interaction among clauses and get our desired aggregated hidden states H . The whole process can be concluded as:

$$H = \text{GAT}(\text{Attn}(\text{T5}(T))),$$

where T5, Attn and GAT denote the T5 model, the attention layer, and the graph attention network, respectively.

2.4 Joint Linear Layers

For our two-stream method, we use three joint linear layers as the classifier as shown in Figure 1(b). For the emotion-cause stream (ECS), to incorporate knowledge learned from the cause-emotion stream (CES), on the one hand, we use L_{cau} on the hidden state of cause candidate h^2 to get a score of this candidate being a cause. On the other hand, we concatenate three hidden states $h_{pair} = (h^1, h_{conn}, h^2)$, where h^1 represents the hidden state of emotion clause, and use pair classifier L_{pair} on h_{pair} to get a score of this candidate being the cause of corresponding emotions. Considering it as a multi-task process, we add output vectors of L_{cau} and L_{pair} . With this sum, we utilize the sigmoid function to gain the final probability of whether this pair is an emotion-cause pair.

$$y_{ecp} = \text{sigmoid}(L_{cau}(h^2) + L_{pair}(h_{pair})),$$

where y_{ecp} is the final probability of emotion-cause pairs with ECS.

We can get the probability of cause-emotion pairs similarly.

$$y_{cep} = \text{sigmoid}(L_{emo}(h^2) + L_{pair}(h_{pair})),$$

where y_{cep} is the final probability of cause-emotion pairs with CES.

2.5 Answer Prediction

We summarize our approach to the answer prediction and define the loss of our model. We simply use Binary Cross-Entropy (BCE) Loss for back-propagation. Let’s take the emotion-cause stream (ECS) as an example. It starts with getting the

hidden states of clauses H . We first use linear layer L_{emo} and sigmoid function to predict emotion clauses directly.

$$y_{emo} = \text{sigmoid}(L_{emo}(h_i)),$$

$$\mathcal{L}_{emo} = \text{BCE}(y_{emo}, y_{Temo}),$$

where y_{emo} denotes the predicted probability of each clause being an emotion clause, while y_{Temo} denotes the true probability; \mathcal{L}_{emo} is the loss of emotion extraction step. We use a common threshold 0.5 for y_{emo} to determine whether the predicted clause is an emotion clause.

Then, we pair up clauses and obtain hidden states of connectives h_{conn} . For emotion-cause pair extraction, we consider it as a multi-task process and predict emotion-cause pairs utilizing L_{cau} and L_{pair} .

$$y_{ecp} = \text{sigmoid}(L_{cau}(h^2) + L_{pair}(h_{pair})),$$

$$\hat{y}_{ecp} = y_{emo} \cdot y_{ecp},$$

$$\mathcal{L}_{ecp} = \text{BCE}(y_{ecp}, y_{Tecp}),$$

where y_{ecp} is the predicted probability of emotion-cause pairs with ECS, while y_{Tecp} denotes the true probability; \hat{y}_{ecp} represents overall probability of emotion-cause pairs for evaluation; \mathcal{L}_{ecp} is the loss of emotion-cause pair extraction step. We also use the common threshold of 0.5 for \hat{y}_{ecp} to determine whether a pair is an emotion-cause pair.

Similarly, we can obtain y_{cau} , y_{cep} as the predicted probability of each clause being a cause clause and of cause-emotion pairs, \hat{y}_{cep} as the overall probability of cause-emotion pairs for evaluation, and \mathcal{L}_{cau} , \mathcal{L}_{cep} as the loss of cause extraction step and cause-emotion pair extraction step. Our total loss \mathcal{L} of model can be represented as

$$\mathcal{L} = \mathcal{L}_{emo} + \mathcal{L}_{ecp} + \mathcal{L}_{cau} + \mathcal{L}_{cep}.$$

3 Experiments

3.1 Dataset

We use the ECPE benchmark dataset (Xia and Ding, 2019). It contains 1941 texts with indices of emotion-cause pairs. We follow its classic 10-fold dataset separation. We use precision P , recall R , and F_1 score defined in (Xia and Ding, 2019) as our metrics to evaluate the performance of our model.

3.2 Experimental Settings

We use T5 (Raffel et al., 2020) Chinese model Pegasus T5 (Su, 2021) to obtain hidden states and

Method	Emotion Extraction			Cause Extraction			Pair Extraction		
	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)
Indep (Xia and Ding, 2019)	82.10	83.75	80.71	62.05	69.02	56.73	58.18	68.32	50.82
Inter-CE (Xia and Ding, 2019)	83.00	84.94	81.22	61.51	68.09	56.34	59.01	69.02	51.35
Inter-EC (Xia and Ding, 2019)	82.30	83.64	81.07	65.07	70.41	60.83	61.28	67.21	57.05
PairGCN (Chen et al., 2020b)	83.75	88.57	79.58	73.75	79.07	69.28	72.02	76.92	67.91
IE-CNN+CRF (Chen et al., 2020a)	81.88	86.14	78.11	64.96	73.48	58.41	66.86	71.49	62.79
ECPE-2D (Ding et al., 2020a)	89.10	86.27	92.21	71.23	73.36	69.34	68.89	72.92	65.44
ECPE-MLL (Ding et al., 2020b)	88.86	86.08	91.91	76.30	73.82	79.12	74.52	77.00	72.35
ECPE-Fan (Fan et al., 2020)	84.74	87.16	82.44	69.74	75.62	64.71	67.99	73.74	63.07
RANKCP (Wei et al., 2020)	90.57	91.23	89.99	76.15	74.61	77.88	73.60	71.19	76.30
ECPE-Yuan (Yuan et al., 2020)	77.39	81.96	73.29	70.18	74.90	66.02	67.76	72.43	63.66
UTOS (Cheng et al., 2021)	85.56	88.15	83.21	74.71	76.71	73.20	72.03	73.89	70.62
MGSAG (Bao et al., 2022)	82.87	87.21	79.11	70.80	75.10	67.13	68.46	72.43	65.07
RSN (Chen et al., 2022a)	87.55	86.14	89.22	75.45	77.27	73.98	73.93	76.01	72.19
A^2 Net (Chen et al., 2022b)	90.80	90.67	90.98	78.35	77.62	79.20	76.34	75.03	77.80
UECA-Prompt (Zheng et al., 2022)	88.16	84.75	91.95	77.55	76.24	79.16	74.70	71.82	77.99
ECPE-MM-R [†] (Zhou et al., 2022)	93.70	97.38	90.38	81.35	83.28	79.64	80.62	82.18	79.27
EPO-ECPE (Hu et al., 2023)	95.00	97.87	92.32	76.20	77.11	75.43	75.64	76.21	75.19
ChatGPT-Wang (Wang et al., 2023b)	-	-	-	-	-	-	52.44	54.13	50.86
ChatGPT-CoT (ours)	23.11	14.62	55.19	15.57	10.00	35.14	4.10	2.59	9.73
TSCER (ours)	93.12	98.42	88.46	83.28	87.63	79.44	79.73	82.55	77.27

Table 2: ECA main results. P , R and F_1 denote precision, recall and F_1 score, respectively. ECPE-MM-R[†] uses a revised truncated dataset.

predict connectives during training. An AdamW optimizer is employed for training, with a weight decay of 0.01, an initial learning rate of 1e-5, and a warm-up rate of 0.1. For regularization, we use the dropout technique with a dropout rate of 0.1. We have experimented with various batch sizes and get the best results when the batch size is set to 4.

Additionally, we follow (Wei et al., 2020) to use a sentimental dictionary to assist the model in identifying emotion clauses. Besides, We incorporate the connectives predicted by ChatGPT in the evaluation for better performance. For direct experiments on ChatGPT, We utilize gpt-3.5-turbo API to conduct preliminary research on its performance of ECA task; see Appendix C for detailed information.

3.3 Main Results

Our results are shown in Table 2. To show the effectiveness of our model, we compare our results with other baselines, which are Indep, Inter-CE, Inter-EC (Xia and Ding, 2019), PairGCN (Chen et al., 2020b), IE-CNN+CRF (Chen et al., 2020a), ECPE-2D (Ding et al., 2020a), ECPE-MLL (Ding et al., 2020b), ECPE-Fan (Fan et al., 2020), RANKCP (Wei et al., 2020), ECPE-Yuan (Yuan et al., 2020), UTOS (Cheng et al., 2021), MGSAG (Bao et al., 2022), RSN (Chen et al., 2022a), A^2 Net (Chen et al., 2022b), UECA-Prompt (Zheng et al., 2022), ECPE-MM-R[†] (Zhou et al., 2022) and EPO-ECPE

(Hu et al., 2023).

For more baseline information, see Appendix B. Nowadays, ChatGPT is considered a strong baseline for various NLP tasks, so we also conduct experiments on it. We get our own ChatGPT evaluation results based on the Chain-of-Thought (CoT) prompt, which is represented as ChatGPT-CoT in the table. We also refer to other ChatGPT researches on ECA (Wang et al., 2023b). Due to the lack of English datasets, we conducted English experiments on the self-built toy dataset and the results are in Appendix B.

Our TSCER model outperforms all the baselines and achieves improvements in precision and almost all in F_1 score in terms of all three tasks. This validates the utility of our proposed two-stream structure with predicted connectives. The performance improvements highlight the importance of connectives and cause-effect relations in analyzing emotion-cause relations in ECA.

Other than the above, we have two interesting findings in Table 2.

- Considering all three tasks, our TSCER models have significant improvements in the F_1 score and precision, but we observe a slight increase or even decrease in recall when compared to other methodologies. This phenomenon may be attributed to our models' stringent criteria for identifying cause-effect

345
346
347

348
349
350
351
352
353
354
355
356

357

358
359
360
361
362
363
364
365
366

367
368
369
370
371
372
373

374
375

376
377
378
379
380

381
382
383
384
385

386
387
388
389
390
391

relationships, potentially leading to the exclusion of genuine emotion-cause pairs with less clearly defined cause-effect connections.

- The results of ChatGPT-Wang (Wang et al., 2023b) and ChatGPT-CoT (ours) show that ChatGPT could not handle this task with a zero-shot CoT prompt. Our ChatGPT approach points out many possible emotion-cause pairs with low-degree emotions and weakly associated causes, so it ends up with an exceptionally poor performance in terms of all three metrics in all three tasks.

4 Analysis

4.1 Ablation Study

The results in Figure 2 have shown the effectiveness of our two-stream model. We are curious about how the performance changes when altering the two-stream structure of our model and without connective prediction. Thus, we design an ablation study and build up three more variants based on TSCER to check the effectiveness of our proposed method.

- The CES-Only model only leverages the cause-emotion stream, and the ECS-Only model only leverages the emotion-cause stream. They no longer share parameters of L_{emo} , L_{cau} and L_{pair} . In addition, these two models don't use the connective prediction module.
- The TSCER (w/o connectives) model sets the connective prediction module unused.

These three new models have some adaptive changes in their model structure and they are easy to understand, so we don't elaborate on them here. Our results are shown in Table 3 and our insight can be organized as follows.

- Comparing ECS-Only with CES-Only, emotion extraction first is more accessible and has better results than cause extraction first, while ECS-Only steadily outperforms CES-Only in most metrics.
- Comparing TSCER with CES-Only and ECS-Only shows the superiority of our proposed two-stream structure. Two-stream cause-effect relations can instruct the model to think thoroughly to obtain higher precision in all three tasks.

- Compared with TSCER (w/o connectives), TSCER (w connectives) gains more improvement in all precision, recall, and the F_1 score with the assistance of connectives. It shows that connectives, as indicators of cause-effect relations, can sift some pairs with ambiguous cause-effect relations to gain higher precision, and unveil those hidden cause-effect relations to obtain higher recall.

4.2 Time Efficiency

Our TSCER model builds pairs like prior pairing matrix methods, but our model builds much fewer pairs as shown in Table 4. We organized a preliminary time complexity analysis to elaborate on this.

Denote the number of clauses, emotion clauses, cause clauses, and window size of the sliding window are n , p , q , and w , respectively. The complexity of former pairing matrix methods is $O(n^2)$ since they need to build all possible pairs and is $O(wn)$ for those who use sliding windows. The time complexity of our method is $O(n + pq)$ for we use a strong pipeline method to greatly reduce meaningless pairs.

However, mathematical formulas linking n , p , q and w still pose challenges. Therefore, we compare the pairs built in the whole ECPE benchmark dataset to reveal their differences. Ours only form 62618 pairs under the best circumstances, which is much less than prior pairing matrix methods. Consequently, we deduce that our method tends to outperform the former pairing matrix approach in complexity in most cases.

4.3 Error Analysis

Our model relies heavily on predicted connectives and wrong connectives usually lead to wrong results. We make an error analysis to state this and the results are shown in Figure 2.

The mistakes are mainly due to inappropriate predicted connectives. Under some circumstances, especially when two clauses in a pair are far apart and the link between them is weak, our method may predict poisoned connectives resulting directly from language model likelihood instead of reasoning. It greatly stops us from achieving better performance due to the lack of a particular Chinese reasoning dataset for connective prediction fine-tuning.

Method	Emotion Extraction			Cause Extraction			Pair Extraction		
	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$	$F_1(\%)$	$P(\%)$	$R(\%)$
CES-Only (w/o connectives)	86.48	90.83	82.64	76.01	81.99	70.92	71.22	74.77	68.10
ECS-Only (w/o connectives)	90.71	96.53	85.63	77.33	81.28	73.95	74.54	76.62	72.74
TSCER (w/o connectives)	90.77	97.28	85.23	77.45	82.59	73.12	75.07	78.56	72.21
TSCER (w connectives)	93.12	98.42	88.46	83.28	87.63	79.44	79.73	82.55	77.27

Table 3: ECA ablation study results. Connective prediction is unused for models tagged “w/o connective”.

Method	Capability of Long-distance Pairs	Time Complexity	Estimated Pairing Count
Pairing Matrix	✓	$O(n^2)$	490367
Pairing Matrix w/ Sliding Window	✗	$O(wn)$	132003
TSCER (ours)	✓	$O(n + pq)$	62,618

Table 4: Time efficiency analysis results. The number of clauses, emotion clauses, cause clauses, and window size of the sliding window are n , p , q , and w , respectively. The estimated pairing count is evaluated by counting all pairs built in the whole ECPE benchmark dataset (Xia and Ding, 2019) under the best circumstances.

4.4 Case Study

We analyze a text shown in Table 5 for demonstration and show the internal steps of how our model solves the ECA problem.

In the emotion-cause stream (ECS), our model first predicts the right emotion clause C_{14} and then predicts connectives between C_{14} and each clause in the text. For (C_{14}, C_{11}) , according to our rules shown in Appendix A, our method first deletes “but” and predicts connective “because” for this pair, which helps model to figure out that this is an emotion-cause pair. It happens the same to (C_{14}, C_{12}) , but our model understands that C_{12} is just an analogy leveraging semantic information and does not consider it as a cause. C_{14} does not have any event or action so it could not be the cause of itself. We could get the emotion-cause pair answer (C_{14}, C_{11}) from ECS.

In the cause-emotion stream (CES), our model first predicts the right cause clause C_{11} and then predicts connectives between C_{11} and each clause in the text. Similarly, C_{11} does not have any emotion expressed so it cannot be the emotion of itself. Considering pair (C_{11}, C_{14}) , with “so” predicted as an indicator, our reasoning model successfully links them as a cause-emotion pair.

Finally, combining these two answers of two streams, we could get the correct emotion-cause pair answer (C_{14}, C_{11}) .

5 Related Work

5.1 Emotion-Cause Analysis (ECA)

Xia and Ding (2019) brought forward the Emotion-

Cause Analysis (ECA) task and published the Chinese Emotion-Cause Pair Extraction (ECPE) benchmark dataset, along with three original proposed methods Indep, Inter-CE, and Inter-EC. Inter-CE and Inter-EC made a preliminary attempt to enhance emotion extraction with cause extraction and vice versa, obtaining limited results with immature model architecture. Previous methods of ECA can be divided into two categories.

One approach is to utilize a clause pairing matrix. Ding et al. (2020a) consider emotion-cause pair extraction as representation, interaction, and prediction of joint two-dimensional clause vectors. Ding et al. (2020b) use a sliding window approach to identify emotion-cause pairs in text, which is based on a multi-label learning method. Wei et al. (2020) propose an end-to-end extraction model focused on effective inter-clause modeling and use it to improve the pair extraction process from a ranking perspective. However, they all suffer from sparse label space and high computational costs.

Another approach is to utilize sequence labels. Chen et al. (2020a) use a unified sequence labeling method for emotion-cause pairs and use a unified model to identify them. Yuan et al. (2020) also consider ECA as a sequence labeling task and use a unique labeling scheme to identify emotion-cause pairs. However, once they model the task as a sequence labeling task, they do not take the relation between emotions and their corresponding causes into consideration.

Some recent works try to use thoughtfully crafted prompts to instruct models to extract emotion-cause pairs. Zheng et al. (2022) develop

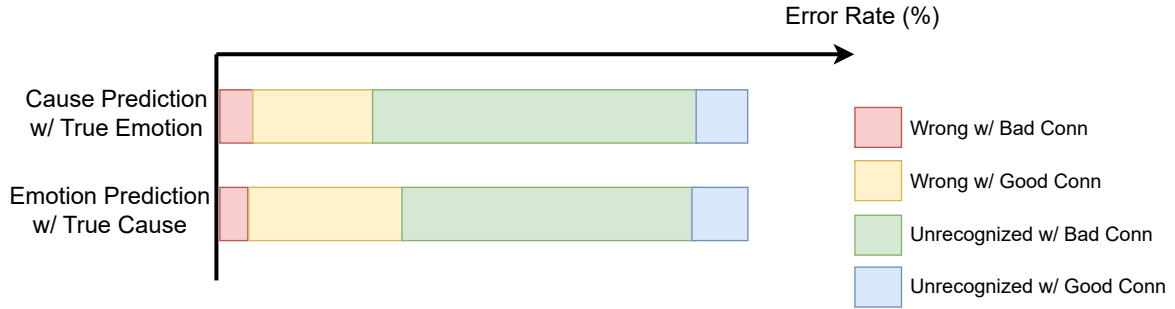


Figure 2: Error analysis breakdown results. “Conn” represents connectives for short.

Clause	Content	ECS Conn Prediction	CES Conn Prediction	ECS Pair Prediction	CES Pair Prediction
C_{10}	I bought these oysters for only 5 yuan	namely	and	-	-
C_{11}	but I got 5 pearls	but because	but and	(C_{14}, C_{11})	-
C_{12}	like buying a lottery ticket and winning a prize	like because	like and	-	-
C_{13}	mentioning this	if	if	-	-
C_{14}	Mr. Zhang is very excited	so	so	-	(C_{11}, C_{14})

Table 5: Case study. Text No.2025. Blue clauses denote predicted emotion clauses in ECS and red clauses denote predicted cause clauses in CES. “Conn” is short for connective.

a universal prompt tuning method to solve different ECA tasks in a unified framework. Zhou et al. (2022) use a multi-turn machine comprehension framework, which uses a multi-turn approach to understand the context information and identify emotion-cause pairs on a revised truncated dataset, while the rethink mechanism allows the model to adjust its predictions. Though these works have made progress in ECA with additional model instructions, they are limited by input length and to some extent ignore that ECA in essence is a causal reasoning problem that needs more than a simple query or prompt to deal with.

5.2 Implicit Discourse Relation Recognition

The publication of the PDTB dataset (Prasad et al., 2008) has stimulated much research in discourse analysis, including implicit discourse relation recognition (IDRR), which inspires us to resolve ECA using implicit discourse information. With the development of deep learning, many talented researchers have come up with novel ways to deal with IDRR. Some of them try to make use of implicit connectives in many different ways.

Braud and Denis (2016) try to learn connective-based word representations and use them to deal with IDRR. Qin et al. (2017) use adversarial connective-exploiting networks for implicit discourse relation classification. They propose a feature imitation framework, in which an implicit

relation network learns from another neural network with access to connectives to extract similarly salient features. Nguyen et al. (2019) use a multi-task learning framework to enable relations and connectives to be simultaneously predicted. Kurfali and Östling (2021) generate candidate explicit discourse markers between sentences and score the resulting segments using a large language model.

6 Conclusions

In conclusion, our research presents a novel two-stream reasoning model that unifies the emotion-cause and cause-emotion streams of thought, thereby enhancing performance in emotion-cause analysis (ECA). By leveraging discourse connectives predicted by ChatGPT, our model uncovers hidden cause-effect relationships, bridging the gap between emotions and their causes. Consequently, our model exhibits a superior understanding of bidirectional cause-effect relationships between emotions and causes. Our methodology offers improved interpretability and aligns more closely with human thought processes than conventional approaches. This approach not only achieves state-of-the-art (SOTA) performance but also introduces a significant innovation by treating the ECA task as an emotional reasoning challenge. It is exciting that our research highlights the potential of applying reasoning methods to advance the field of ECA.

563 Limitations

564 We list some of our main limitations.

- 565 • Still High Computational Cost

566 Compared with models using clause pairing ma-
567 trix (Ding et al., 2020a,b; Wei et al., 2020) consid-
568 ering all possible pair combinations, our method
569 build up much fewer pairs; see Section 4.2. How-
570 ever, we still consider all clauses in the text as can-
571 didate cause clauses in the emotion-cause stream
572 (ECS) and as candidate emotion clauses in the
573 cause-emotion stream (CES). Though our model is
574 much better than previous ones, it still suffers from
575 high computational cost, since our method needs to
576 build 62618 pairs for the ECPE benchmark dataset,
577 while only 2167 emotion-cause pairs are true ones.

- 578 • Insufficient Hyper-parameters Selection Exper-
579 iment

580 Apart from the hyper-parameters of the model,
581 our model has set many thresholds as hyper-
582 parameters for judgment of extraction results; see
583 Section 2.5. This leads to various possible choices
584 of them in the space of hyper-parameters. Unfor-
585 tunately, we don't have so much computational
586 resources to search for the optimized combination
587 of hyper-parameters. Consequently, there is still
588 room for improvement and the efficiency of our
589 model is yet to be revealed.

- 590 • Lack of English Dataset and Bad Performance
591 on Self-built Toy Dataset

592 Due to the lack of English datasets, we con-
593 ducted English experiments on the self-built toy
594 dataset and the results are in Appendix B. This toy
595 dataset is built by directly translating the ECPE
596 benchmark dataset (Xia and Ding, 2019) using a
597 translator. Due to poor translation, we get poor
598 performance on it. So this task urgently needs a
599 high-quality dataset to push forward the English
600 study of it.

601 References

602 Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wen-
603 liang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei
604 Ji, Tiezheng Yu, Willy Chung, Quyet V. Do, Yan
605 Xu, and Pascale Fung. 2023. A multitask, multilin-
606 gual, multimodal evaluation of chatgpt on reasoning,
607 hallucination, and interactivity.

608 Yinan Bao, Qianwen Ma, Lingwei Wei, Wei Zhou, and
609 Songlin Hu. 2022. Multi-granularity semantic aware
610 graph model for reducing position bias in emotion
611 cause pair extraction. In *Findings of the Associa-*
612 *tion for Computational Linguistics: ACL 2022*, pages

1203–1213, Dublin, Ireland. Association for Compu-
tational Linguistics. 613 614

Chloé Braud and Pascal Denis. 2016. Learning
connective-based word representations for implicit
discourse relation identification. In *Proceedings of
the 2016 Conference on Empirical Methods in Nat-
ural Language Processing*, pages 203–213, Austin,
Texas. Association for Computational Linguistics. 615 616 617 618 619 620

Fang Chen, Ziwei Shi, Zhongliang Yang, and Yongfeng
Huang. 2022a. Recurrent synchronization network
for emotion-cause pair extraction. *Know.-Based Syst.*,
238(C). 621 622 623 624

Shunjie Chen, Xiaochuan Shi, Jingye Li, Shengqiong
Wu, Hao Fei, Fei Li, and Donghong Ji. 2022b. Joint
alignment of multi-task feature and label spaces for
emotion cause pair extraction. In *Proceedings of
the 29th International Conference on Computational
Linguistics*, pages 6955–6965, Gyeongju, Republic
of Korea. International Committee on Computational
Linguistics. 625 626 627 628 629 630 631 632

Xinhong Chen, Qing Li, and Jianping Wang. 2020a. A
unified sequence labeling model for emotion cause
pair extraction. In *Proceedings of the 28th Inter-
national Conference on Computational Linguistics*,
pages 208–218, Barcelona, Spain (Online). Interna-
tional Committee on Computational Linguistics. 633 634 635 636 637 638

Ying Chen, Wenjun Hou, Shoushan Li, Caicong Wu,
and Xiaoqiang Zhang. 2020b. End-to-end emotion-
cause pair extraction with graph convolutional net-
work. In *Proceedings of the 28th International Con-
ference on Computational Linguistics*, pages 198–
207, Barcelona, Spain (Online). International Com-
mittee on Computational Linguistics. 639 640 641 642 643 644 645

Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Na Li, and
Qing Gu. 2021. A unified target-oriented sequence-
to-sequence model for emotion-cause pair extraction.
*IEEE/ACM Transactions on Audio, Speech, and Lan-
guage Processing*, 29:2779–2791. 646 647 648 649 650

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020a. ECPE-
2D: Emotion-cause pair extraction based on joint two-
dimensional representation, interaction and predic-
tion. In *Proceedings of the 58th Annual Meeting of
the Association for Computational Linguistics*, pages
3161–3170, Online. Association for Computational
Linguistics. 651 652 653 654 655 656 657

Zixiang Ding, Rui Xia, and Jianfei Yu. 2020b. 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)*, pages 3574–3583,
Online. Association for Computational Linguistics. 658 659 660 661 662 663

Chuang Fan, Chaofa Yuan, Jiachen Du, Lin Gui, Min
Yang, and Ruifeng Xu. 2020. Transition-based di-
rected graph construction for emotion-cause pair ex-
traction. In *Proceedings of the 58th Annual Meet-
ing of the Association for Computational Linguistics*,
pages 3707–3717, Online. Association for Computa-
tional Linguistics. 664 665 666 667 668 669 670

671	Guimin Hu, Yi Zhao, and Guangming Lu. 2023. Emotion prediction oriented method with multiple supervisions for emotion-cause pair extraction . <i>IEEE/ACM Transactions on Audio, Speech, and Language Processing</i> , 31:1141–1152.	728
672		729
673		730
674		731
675		732
676	Christmas Humphreys. 2005. <i>Karma and rebirth: The karmic law of cause and effect</i> . Routledge.	733
677		
678	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners .	734
679		735
680		
681	Murathan Kurfalı and Robert Östling. 2021. Let’s be explicit about that: Distant supervision for implicit discourse relation classification via connective prediction . In <i>Proceedings of the 1st Workshop on Understanding Implicit and Underspecified Language</i> , pages 1–10, Online. Association for Computational Linguistics.	736
682		737
683		738
684		
685		739
686		740
687		741
688	Md Tahmid Rahman Laskar, M Saiful Bari, Mizanur Rahman, Md Amran Hossen Bhuiyan, Shafiq Joty, and Jimmy Xiangji Huang. 2023. A systematic study and comprehensive evaluation of chatgpt on benchmark datasets .	742
689		743
690		744
691		745
692		
693	Andrew Nedilko and Yi Chu. 2023. Team bias busters at WASSA 2023 empathy, emotion and personality shared task: Emotion detection with generative pre-trained transformers . In <i>Proceedings of the 13th Workshop on Computational Approaches to Subjectivity, Sentiment, & Social Media Analysis</i> , pages 569–573, Toronto, Canada. Association for Computational Linguistics.	746
694		747
695		748
696		749
697		
698		750
699		751
700		752
701	Linh The Nguyen, Linh Van Ngo, Khoat Than, and Thien Huu Nguyen. 2019. Employing the correspondence of relations and connectives to identify implicit discourse relations via label embeddings . In <i>Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics</i> , pages 4201–4207, Florence, Italy. Association for Computational Linguistics.	753
702		754
703		755
704		
705		756
706		757
707		758
708		759
709	Emily Pitler and Ani Nenkova. 2009. Using syntax to disambiguate explicit discourse connectives in text . In <i>Proceedings of the ACL-IJCNLP 2009 Conference Short Papers</i> , pages 13–16, Suntec, Singapore. Association for Computational Linguistics.	760
710		761
711		762
712		763
713		764
714	Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltakaki, Livio Robaldo, Aravind Joshi, and Bonnie Webber. 2008. The Penn Discourse TreeBank 2.0 . In <i>Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC’08)</i> , Marrakech, Morocco. European Language Resources Association (ELRA).	765
715		766
716		767
717		768
718		769
719		770
720		771
721	Lianhui Qin, Zhisong Zhang, Hai Zhao, Zhiting Hu, and Eric Xing. 2017. Adversarial connective-exploiting networks for implicit discourse relation classification . In <i>Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 1006–1017, Vancouver, Canada. Association for Computational Linguistics.	772
722		773
723		774
724		775
725		776
726		777
727		778
	Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer . <i>Journal of Machine Learning Research</i> , 21(140):1–67.	779
	Jianlin Su. 2021. T5 pegasus - zhuiyai . Technical report, zhuiyi.ai.	780
	Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph attention networks .	781
	Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models .	782
	Zengzhi Wang, Qiming Xie, Zixiang Ding, Yi Feng, and Rui Xia. 2023b. Is chatgpt a good sentiment analyzer? a preliminary study . <i>arXiv preprint</i> .	783
	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models .	784
	Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction . In <i>Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics</i> , pages 3171–3181, Online. Association for Computational Linguistics.	785
	Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts . In <i>Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics</i> , pages 1003–1012, Florence, Italy. Association for Computational Linguistics.	786
	Kailai Yang, Shaoxiong Ji, Tianlin Zhang, Qianqian Xie, Ziyang Kuang, and Sophia Ananiadou. 2023. Towards interpretable mental health analysis with chatgpt .	787
	Chaofa Yuan, Chuang Fan, Jianzhu Bao, and Ruifeng Xu. 2020. Emotion-cause pair extraction as sequence labeling based on a novel tagging scheme . In <i>Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)</i> , pages 3568–3573, Online. Association for Computational Linguistics.	788
	Xiaopeng Zheng, Zhiyue Liu, Zizhen Zhang, Zhaoyang Wang, and Jiahai Wang. 2022. UECA-prompt: Universal prompt for emotion cause analysis . In <i>Proceedings of the 29th International Conference on Computational Linguistics</i> , pages 7031–7041, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.	789
	Changzhi Zhou, Dandan Song, Jing Xu, and Zhijing Wu. 2022. A multi-turn machine reading comprehension framework with rethink mechanism for emotion-cause pair extraction . In <i>Proceedings of the 29th</i>	790
		791
		792

A Discussion on Connective Prediction

A.1 Model Choice

We could choose unidirectional language models like GPT to predict connectives. We can get the probability of connectives by directly estimating the language model likelihood of the complete pair with predicted connectives.

$$P_{conn}(c_i|C_{emo}, C_{cand}) = P_{LM}(C_{emo}, c_i, C_{cand})/P_{sum},$$
$$P_{sum} = \sum_i P_{LM}(C_{emo}, c_i, C_{cand}),$$

where c_i represents each possible connective, and P_{conn} , P_{LM} denote the probability of connectives and language model likelihood, respectively. However, in this preliminary method, these generation models only concern information about the previous token sequence, so the information of candidates does not affect the prediction of connectives, and thus this model couldn't represent the reasoning process of connective prediction. Therefore, we choose to use a T5 masked language model for connective prediction at last, considering its ability to construct bidirectional context relationships.

A.2 Rules for Connective Prediction

Here are the rules we follow when predicting connectives.

- We first search for several continuous sequences of connectives from the beginning of the candidate clause to the end. For example, "but because" is a continuous sequence of connectives. We define sequence starting at the beginning of the candidate clause as the head sequence.
- If the clause forms a pair with itself, we delete the head sequence if it exists. If there is a single-token causal connective in other sequences of connectives, we directly choose it as our predicted connectives. If not, we predict the connectives with the T5 model. And if there are multiple choices, we choose the first one.

B Baseline Comparison

B.1 Baselines

The results of baselines are shown in Table 2, and here we conduct a brief introduction of them.

- **Indep, Inter-CE, Inter-EC (Xia and Ding, 2019)** are three original methods proposed with the ECPE benchmark dataset. Inter-CE and Inter-EC made a preliminary attempt to enhance emotion extraction with cause extraction and vice versa.

- **PairGCN (Chen et al., 2020b)** uses a graph convolutional network with a sliding window approach.

- **IE-CNN+CRF (Chen et al., 2020a)** uses a unified sequence labeling method for emotion-cause pairs and uses a unified model to identify them.

- **ECPE-2D (Ding et al., 2020a)** considers emotion-cause pair extraction as representation, interaction, and prediction of joint two-dimensional clause vectors.

- **ECPE-MLL (Ding et al., 2020b)** uses a sliding window approach to identify emotion-cause pairs in text, which is based on a multi-label learning method.

- **ECPE-Fan (Fan et al., 2020)** transforms the ECA task into a parsing-like directed graph construction procedure and generates labeled edges to extract emotion-cause pairs.

- **RANKCP (Wei et al., 2020)** proposes an end-to-end extraction model focused on effective inter-clause modeling and uses it to improve the pair extraction process from a ranking perspective.

- **ECPE-Yuan (Yuan et al., 2020)** considers ECA as a sequence labeling task and uses a unique labeling scheme to identify emotion-cause pairs.

- **UTOS (Cheng et al., 2021)** uses a unified target-oriented sequence labeling scheme.

- **MGSAG (Bao et al., 2022)** considers reducing the position bias in the ECPE dataset.

- **RSN (Chen et al., 2022a)** uses emotion extraction and cause extraction to reinforce the result of ECPE through explicit information transmission.

- **A²Net (Chen et al., 2022b)** develop a novel parsing matrix method with alignment mechanisms leveraging Kullback Leibler (KL) divergence.

- **ECPE-MM-R (Zhou et al., 2022)** uses a multi-turn machine comprehension framework, which

Method	Emotion Extraction			Cause Extraction			Pair Extraction		
	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)	F_1 (%)	P (%)	R (%)
RANKCP (Chinese)	90.57	91.23	89.99	76.15	74.61	77.88	73.60	71.19	76.30
TSCER (ours)	93.12	98.42	88.46	83.28	87.63	79.44	79.73	82.55	77.27
RANKCP (English)	72.77	81.76	65.57	56.12	61.11	51.89	49.14	51.81	56.73
TSCER (English, ours)	74.87	84.12	67.45	56.71	61.20	52.83	49.76	51.50	48.13

Table 6: ECA English results on the self-built toy dataset. P , R and F_1 denote precision, recall and F_1 score, respectively.

阅读下列段落: **TEXT**. 找出其中含有情感的句子和其对应的能解释其产生原因的句子, 用列表表示其对应关系并输出, 比如“[1,4], [2,5]”。对应序号已在段落中做过标记。让我们一步一步地思考。

Read the following paragraph: **TEXT**. Find the sentences containing emotions and their corresponding sentences that can explain the reasons for their occurrence, express their correspondence with a list and output, such as “[1,4], [2,5]”. The corresponding ordinal number is marked in the paragraph. Let’s think step by step.

Figure 3: ChatGPT-CoT prompt. “TEXT” denotes a text in the corpus without labels and emotional annotations.

uses a multi-turn approach to understand the context information and identify emotion-cause pairs on a revised truncated dataset, while the rethink mechanism allows the model to adjust its predictions.

- UECA-Prompt (Zheng et al., 2022) develops a universal prompt tuning method to solve different ECA tasks in a unified framework.

- EPO-ECPE (Hu et al., 2023) uses multiple supervisions with a sliding window approach, which originated from ECPE-MLL (Ding et al., 2020b).

B.2 English Study

To the best of our knowledge, English datasets for this specific Emotion-Cause Pair Extraction (ECPE) problem do not exist. Consequently, our study has been confined to Chinese as a result of this limitation. Due to a lack of English datasets, we have made an effort to translate the ECPE benchmark dataset (Xia and Ding, 2019) into English. But the translation quality is far from satisfaction as shown in Table 6, rendering the utilization of the translated data impractical.

C Discussion on ChatGPT Approach

C.1 Our ChatGPT Approach

To explore the reasoning ability of LLM, we conducted a preliminary attempt to utilize ChatGPT

and chain-of-thought (CoT) (Wei et al., 2023; Kojima et al., 2023; Wang et al., 2023a) method to identify emotions and their corresponding causes directly. For experiments on ChatGPT, we utilize gpt-3.5-turbo API and use the prompt as shown in Figure 3. We set every parameter by default.

We don’t tell ChatGPT how to deal with the task but only use a zero-shot CoT prompt to ask questions. Figure 4 shows that ChatGPT understands what to do without further instruction, thereby it proves that our prompt should suffice. Figure 5 demonstrates the ChatGPT answer of the example shown in Table 1.

C.2 Rules for Evaluation

We set some rules for the evaluation of the ChatGPT-CoT method.

- If the output of ChatGPT for a text is irrelevant to our task after three attempts, we simply ignore this section and don’t use it for evaluation.
- If its final answer is right, but with a wrong reasoning step, inappropriate reasoning, or hallucination, we consider it as a false positive instance.
- For ChatGPT to give a lot of wrong answers without any reasoning step with zero-shot CoT guidance, we simply judge them as false negative instances.

C.3 ChatGPT Performance Analysis

The performance of ChatGPT has already been shown in Table 2. We also refer to other ChatGPT researches on ECA (Wang et al., 2023b). Their results are much higher than ours since they only use texts with one emotion-cause pair and build finely crafted prompts based on this. It may also result from the performance gap between ChatGPT API and ChatGPT itself. But in conclusion, these results are still much worse than SOTA results.

940 We have some conjectures about the poor per-
941 formance of ChatGPT. ChatGPT works badly on
942 some complex reasoning tasks, so it is within con-
943 sideration that it works poorly for ECA, for this
944 task involves emotion recognition, reasoning, and
945 long-distance relation extraction.

946 [Nedilko and Chu \(2023\)](#) demonstrate that the
947 best macro F1 score of ChatGPT or GPT-4 is even
948 lower than the baseline XGBoost Classifier (0.46
949 compared to 0.51-0.56). LLMs perform much infe-
950 rior to humans when it comes to more complex
951 causal structures involving multiple events and
952 multi-hop reasoning, it may also have a halluci-
953 nation problem ([Bang et al., 2023](#)).

954 [Laskar et al. \(2023\)](#) have hypothesized that Chat-
955 GPT is prone to hallucination, and tends to answer
956 based on memorization of the original task learned
957 during its pre-training stage, instead of answering
958 with proper reasoning when no step-by-step instruc-
959 tion to solve a new task is provided. ChatGPT out-
960 performs traditional neural networks such as CNN
961 and GRU, showing its potential in mental health
962 analysis and emotional reasoning in conversations.
963 However, it significantly underperforms advanced
964 supervised methods on all tasks, highlighting the
965 challenges of emotion-related subjective tasks for
966 ChatGPT ([Yang et al., 2023](#)).

967 Compared to ChatGPT, our proposed method
968 leveraging two-stream cause-effect relations and
969 connectives successfully modeled the emotion-
970 cause reasoning process, even for long-distance
971 situations.



Certainly, let's break down the paragraph step by step to identify sentences containing emotions and their corresponding sentences that explain the reasons for their occurrence.

Step 1: Identify Sentences Containing Emotions

To start, we need to identify sentences in the paragraph that express emotions. Emotions are often conveyed through words and phrases that indicate feelings, such as "happy," "sad," "excited," "angry," "frustrated," and so on. Please provide the paragraph you'd like me to analyze for emotions, and I'll help you identify the emotional sentences.

Step 2: Identify Reasons for Emotions


Once we've identified the emotional sentences, we'll then look for the sentences that provide reasons or explanations for why those emotions are occurring. These reasons may involve events, circumstances, or triggers that elicit the emotions.

Step 3: Output Corresponding Sentence Pairs

After identifying both the emotional sentences and their corresponding explanations, we'll provide a list of pairs indicating which sentences go together. These pairs will be in the format you mentioned, such as "[1,4], [2,5]," where the first number represents the emotional sentence's ordinal number, and the second number represents the ordinal number of the sentence explaining the emotion.

Figure 4: ChatGPT understanding of the prompt. We input the prompt without a text in the corpus and ChatGPT seems to understand what to do without further instruction.

1, yesterday morning
2, a policeman visited the old man with the lost money
3, told him that the thief was caught
4, the old man was very happy
5, and deposited the money in the bank



Thank you for providing the paragraph. Let's identify the sentences containing emotions and their corresponding sentences explaining the reasons for their occurrence:

Emotional Sentence 1: "the old man was very happy" (Sentence 4)
Explanation Sentence 1: "a policeman visited the old man with the lost money" (Sentence 2)

Based on the paragraph you provided, the emotional sentence "the old man was very happy" (Sentence 4) is explained by the preceding sentence, which is "a policeman visited the old man with the lost money" (Sentence 2). Therefore, the corresponding sentence pair is "[4,2]."

Figure 5: ChatGPT answer. We use the example shown in Table 1 for demonstration.