A Consistent Dual-MRC Framework for Emotion-cause Pair Extraction

ZIFENG CHENG, ZHIWEI JIANG, YAFENG YIN, CONG WANG, SHIPING GE, and QING GU, State Key Laboratory for Novel Software Technology, Nanjing University, China

Emotion-cause pair extraction (ECPE) is a recently proposed task that aims to extract the potential clause pairs of emotions and its corresponding causes in a document. In this article, we propose a new paradigm for the ECPE task. We cast the task as a two-turn machine reading comprehension (MRC) task, i.e., the extraction of emotions and causes is transformed to the task of identifying answer clauses from the input document specific to a query. This two-turn MRC formalization brings several key advantages: First, the QA manner provides an explicit pairing way to identify causes specific to the target emotion; second, it provides a natural way of jointly modeling the emotion extraction, the cause extraction, and the pairing of emotion and cause; and third, it allows us to exploit the well-developed MRC models. Based on the two-turn MRC formalization, we propose a dual-MRC framework to extract emotion-cause pairs in a dual-direction way, which enables a more comprehensive coverage of all pairing cases. Furthermore, we propose a consistent training strategy for the second-turn query, so the model is able to filter the errors produced by the first turn at inference. Experiments on two benchmark datasets demonstrate that our method outperforms previous methods and achieves state-of-the-art performance. All the code and data of this work can be obtained at https://github.com/zifengcheng/CD-MRC.

CCS Concepts: • **Information systems** → **Sentiment analysis**; **Data mining**; • **Applied computing** → **Document analysis;**

Additional Key Words and Phrases: Emotion-cause pair extraction, sentiment analysis, machine reading comprehension

ACM Reference format:
Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Cong Wang, Shiping Ge, and Qing Gu. 2023. A Consistent Dual-MRC Framework for Emotion-cause Pair Extraction. *ACM Trans. Inf. Syst.* 41, 4, Article 105 (April 2023), 27 pages. https://doi.org/10.1145/3558548

1 INTRODUCTION

Identifying the causes (stimulus) behind emotions is an interesting research direction in the field of sentiment analysis [55, 56] and has received growing attention in recent years [5, 30, 31, 35, 42,

This work was supported by the National Science Foundation of China under Grants 61906085, 62172208, 61972192, 41972111; the Second Tibetan Plateau Scientific Expedition and Research Program under Grant 2019QZKK0204. This work is partially supported by Collaborative Innovation Center of Novel Software Technology and Industrialization.
Authors' address: Z. Cheng, Z. Jiang (corresponding author), Y. Yin, C. Wang, S. Ge, and Q. Gu, State Key Laboratory for Novel Software Technology, Nanjing University, 163 Xianlin Ave, Nanjing, Jiangsu, 210023, China; emails: chengzf@smail.nju.edu.cn, {jzw, yafeng, guq}@nju.edu.cn, {cw, shipingge}@smail.nju.edu.cn.

1046-8188/2023/04-ART105 $15.00
https://doi.org/10.1145/3558548

47, 51]. Understanding why the emotions occur is commercially valuable for empathetic chatbot [35], web applications such as product reviews mining, and user feedback analysis. Besides, it also provides a way to analyze and understand web texts (e.g., news, posts, and responses) more deeply.

Early studies on emotion cause analysis mainly focused on the clause-level **emotion cause extraction (ECE)** task, which aims to extract the cause clauses of a given emotion clause. Recently, considering that the need of emotion annotation greatly limits the practical application of the ECE task, the **emotion-cause pair extraction (ECPE)** task has been proposed [51], which aims to extract the potential clause pairs of emotions and their corresponding causes in a document. Figure 1 shows an example of the ECPE task. The input is a document, which contains five clauses. There are two clauses carrying emotion (called emotion clause) in the document, which are clause c_2 ("but I was tired of") and c_5 ("which makes me disgusted"), respectively. For the emotion clause c_2, its corresponding cause clause is c_2 itself ("going to the same restaurant always"). For the emotion clause c_5, its corresponding cause clause is c_4 ("but my friend says that it is affordable"). The output is a set of all emotion-cause pairs in the document: (c_2,c_2), (c_5,c_4).

To address the ECPE task, researchers have proposed many approaches, which can roughly fall into four major categories: the pipelined approach [51], which first uses tagging models to identify emotions and causes individually, and then pairs them and filters out the invalid pairs; the pair filtering approach [8, 14, 18, 45, 49, 50], which filters out invalid pairs from all candidate pairs based on the representation learning of pairs; the unified labeling approach [6, 54], which identifies both emotion and cause, as well as how they pair, by one pass of unified sequence labeling; and the sliding window-based approach [11, 15], which identifies causes/emotions within the local context (defined by a sliding window) of the target emotion/cause.

Although these approaches have demonstrated their effectiveness, there are still several key issues with them. First, the pipelined approach separates the three subtasks (i.e., emotion extraction, cause extraction, and the pairing of emotion and cause) into two steps, which prevents these subtasks from benefiting each other. Hence, how to optimize these subtasks jointly and make them mutually beneficial is a challenge. Second, the pair filtering approach reformulates the ECPE task as a pair filtering problem and validates the pair merely based on the pair representation. But such pair representation can hardly capture all the lexical, semantic, and syntactic cues in the context, especially when one clause contains both emotion and cause or is involved in multiple pairs. Thus, how to adequately utilize the context for pair extraction is another challenge. Third, the last two approaches simplify the ECPE task by setting up some assumptions (e.g., A1: each clause has at most one unified label, or A2: emotion and cause are often paired locally). These assumptions are practically useful but will lead to incomplete coverage of all potential emotion-cause pairs, since the relation of emotion and cause can be complicated, such as one-to-many, many-to-one, and even overlapped or located far away. Thus, it is challenging to ensure a full coverage of all pairing cases.

To address the aforementioned challenges, we formalize the ECPE task as a **machine reading comprehension (MRC)** task. Given a query and a document, MRC task aims to capture the interaction between them and extract specific clauses from the document as the answer. Due to the complexity of ECPE, we devise two-turn queries to identify all emotion-cause pairs. By introducing the answers of the first turn into the second turn as the target, the pairing process can be completed and the three subtasks can be optimized jointly. For example, given the document in Figure 1, we can identify the emotion clause c_5 in the first turn and introduce it into the second-turn query to jointly identify its corresponding cause clause c_4. In this way, the pairing process can adequately utilize the whole document as context for pair extraction. Besides, since each turn takes the whole document as input, a full coverage of all pairing cases can be ensured.

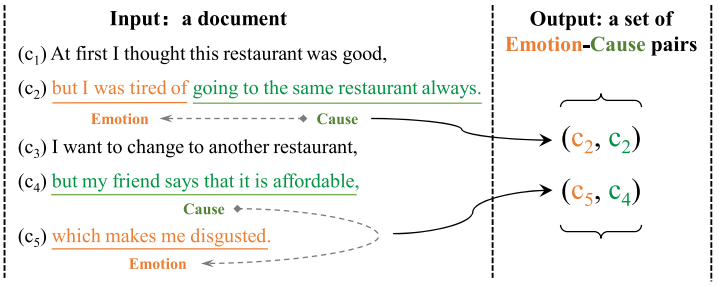


Fig. 1. An example of the ECPE task.

Based on the two-turn MRC formalization, we propose a dual-MRC framework to address the ECPE task. Specifically, we conduct the two-turn MRC in a dual-direction way. In one direction, we first identify all emotion clauses based on the emotion extraction query and then identify the corresponding cause clauses for each emotion clause based on the emotion-specific cause extraction query. Similarly, in the other direction, we first identify all cause clauses and then identify the corresponding emotion clauses for each cause clause. Then, we propose four combination strategies to combine the results of both directions to get the final set of emotion-cause pairs.

In this dual-MRC framework, to train a two-turn MRC model, a natural training strategy is to generate two turns of queries and corresponding results directly from the ground-truth pairs. However, such training strategy is easy to bring a gap between training and inference. At training time, the second-turn queries are generated based on ground-truth emotion/cause clauses, while at inference time, the second-turn queries are generated based on the predicted emotion/cause clauses of the first turn. As a result, if the first turn predicts some error emotion/cause clauses, then such inconsistency between training and inference will make the model unable to handle the error in the second-turn query and thus lead to error accumulation. This issue is also called *exposure bias* [43], which means that the model is never exposed to its own errors during training. To address this issue, we propose a consistent training strategy for the second-turn query, so the model is able to filter the errors produced by the first turn at inference. The consistent training strategy is realized by training model with some extra negative second-turn queries.

To verify the effectiveness of our approach, we conduct experiments on two public ECPE datasets and answer a number of research questions:

- **RQ1:** Does our proposed CD-MRC method outperform existing ECPE methods?
- **RQ2:** How do each of the components of our CD-MRC method contribute to the final performance?
- **RQ3:** What is effect of the query designment on the performance of our method?
- **RQ4:** What is effect of the combination strategy on the performance of our method?
- **RQ5:** What is effect of the consistent training strategy on the performance of our method?
- **RQ6:** How does our method perform in concrete cases compared to the baseline methods and ablation methods?

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

- We formalize the emotion-cause pair extraction task as a two-turn MRC task. Based on this formalization, the three subtasks of ECPE can be addressed in a unified framework and optimized jointly. The QA manner provides an explicit pairing way to identify causes specific to the target emotion.
- We propose a **Consistent Dual-MRC (CD-MRC)** framework to extract emotion-cause pairs in a dual-direction way, which enables a more comprehensive coverage of all pairing

cases. To comprehensively explore how to better combine the predicted pairs from two directions in the inference phase, we propose four combination strategies, including *intersection*, *union*, *harmonic*, and *complementary*.
- We develop a consistent training strategy for the second-turn query to mitigate exposure bias in our MRC framework. To the best of our knowledge, our CD-MRC is the first attempt to consider the inconsistency problem of multi-turn MRC framework between training and inference.
- We conduct extensive experiments on two benchmark datasets. The experimental results demonstrate that our method outperforms all previous ECPE methods and each component of our method is effective.

2 RELATED WORK

In this section, we introduce the following three research topics (i.e., emotion-cause analysis, emotion-cause pair extraction, and machine reading comprehension) relevant to our work.

2.1 Emotion Cause Extraction

Lee et al. [26] first proposed **emotion cause extraction (ECE)** task, which aims to extract the causes behind a given emotion expression. They constructed a small-scale dataset for the ECE task and formulated the task as a word-level sequence labeling problem. Based on this setting, there are some rule-based methods [9, 20] and machine learning methods [22] to deal with the ECE task.

Chen et al. [9] suggested that the ECE task may be more suitable to be addressed at the clause level than word level and extracted causes through six groups of manually constructed linguistic cues. Following this task setting, Gui et al. [25] extended the rule-based features to 25 linguistics cues, then trained classifiers on SVM and CRFs to detect causes. Afterwards, Gui et al. [24] released a Chinese ECE corpus collected from SINA city news that became a benchmark corpus of the latter studies on the ECE task. Inspired by the success of deep learning [1], recent studies have begun to apply the deep learning methods to solve this task [7, 13, 17, 23, 30–33, 46, 52, 53]. Existing models can roughly fall into two major categories: position-insensitive models [7, 33] and position-aware models [13, 17, 52].

Position-insensitive models do not consider position information and predict clauses independently. Li et al. [33] proposed a CNN-based model with co-attention mechanism. Chen et al. [7] proposed use emotion classification task to enhance cause extraction. Considering that the distance between emotion and cause clause is relatively close in practice, many methods consider position information. Ding et al. [13] reordered clauses based on their distances from the emotion clause and transformed the task from an independent prediction problem into a reordered prediction problem. Xia et al. [52] encoded the relative position and global predication information into the transformer framework. Fan et al. [17] proposed a hierarchical neural network and introduced a regularizer biased by relative position information to supervise the representation learning of text.

2.2 Emotion-Cause Pair Extraction

Recently, considering that the emotions are often not given in practice, Xia and Ding [51] proposed the **emotion-cause pair extraction (ECPE)** task, which has attracted a lot of attention [6, 8, 10, 14, 15, 19, 54]. To address the ECPE task, Xia and Ding [51] proposed a pipelined method, which first extracts the emotion and cause individually and then pairs and filters them. In this method, the detection of emotion and cause and the matching of emotion and cause are separately implemented in two steps. Although this method is very effective, it has two shortcomings: (1) The errors from

the first step will affect the performance of second step. (2) The training of the model is also not directly aimed at extracting the final emotion-cause pair. Afterwards, many joint models are proposed through end-to-end training, which can roughly fall into three major categories: the pair filtering approach, the unified labeling approach, and the sliding window-based approach.

The first type of approach directly constructs the representation of candidate emotion-cause pair for prediction [8, 14, 18, 45, 49, 50]. Among these approaches, Ding et al. [14] integrated the emotion-cause pair representation, interaction, and prediction into a joint a framework. Wei et al. [49] emphasized inter-clause modeling from a ranking perspective. Both Wu et al. [50] and Song et al. [45] proposed multi-task framework to extract emotion-cause pairs, but Wu et al. [50] selected a subset of all possible emotion-clause pairs according to distance. Fan et al. [18] constructed pair representation through transition-based model. Chen et al. [8] modeled dependency relations among the candidate pairs in a local neighborhood through graph network. This type of method provides an end-to-end way to train neural networks to directly extract pairs. However, the pair representation hardly captures all the lexical, semantic, and syntactic cues in the context, and such methods often require the construction of a large number of candidate pairs.

Different from the above approaches, the second type of approach transforms the ECPE task into a unified sequence labeling problem. Yuan et al. [54] encoded the relative distance of emotion and cause in the unified labels for pairing. Fan et al. [19] further refined the labels of emotion-cause pair extraction through the results of emotion extraction and cause extraction. Chen et al. [6] encoded the emotion types in the unified labels for pairing and proposed a stacked neural network. Cheng et al. [10] encoded the pair index in the unified labels for pairing and proposed a unified target-oriented sequence-to-sequence model for sequence labeling. The sequence labeling method is very concise and can accomplish the three subtasks by simply assigning a label to each clause. However, the sequence labeling method cannot ensure a full coverage of all potential pairs. For example, if a clause is an emotion clause corresponding to one clauses and also a cause clause corresponding to another clause, then only one of these two pairs can be extracted by the sequence labeling method.

For the third type of approach, the emotion-cause pairs are extracted based on sliding window mechanism. Cheng et al. [11] searched the emotion or cause corresponding to the center clauses of the sliding window. Ding et al. [15] supposed each clause to be an emotion or cause and identify whether there is the corresponding cause or emotion in the sliding window. The sliding window method makes very effective assumptions (i.e., the distance of emotion clause and its corresponding cause clauses is close) and the ISML proposed by Ding et al. [15] achieves the state-of-the-art performance. However, the sliding window methods also cannot cover all potential pairs. When the emotion clause and its corresponding cause clause are not in the same window, these pairs cannot be extracted.

Different from these methods, we formalize the ECPE task as a two-turn MRC task, which allows the pairing of emotion and cause to be realized in a QA manner. Compared with the three types of end-to-end methods, our method can capture the lexical, semantic, and syntactic cues in context more easily, thus is more promising to well identify the emotions and corresponding causes. Besides, our method ensures a full coverage of all potential pairs theoretically and gets rid of constructing a large number of candidate pairs. Note that although our method exhibits to be a two-turn pipelined method, it has several significant differences from the two-step pipelined method proposed by Xia and Ding [51]. First, our method optimizes the query tasks of two turns jointly and trains the MRC model end-to-end, while the two-step pipelined method builds two different models for two stages, and these models are optimized separately. Second, our MRC method can better utilize the context for the pairing of emotion and cause, while the two-step pipelined method filters candidate pairs only based on the features of corresponding clauses in candidate

pairs, which may easily neglect the context. Third, our method tries to mitigate the error accumulation problem based on the proposed consistent training strategy, while the two-step pipelined method does not take the problem into consideration.

2.3 Machine Reading Comprehension

Machine reading comprehension (MRC) aims to give the answer based on the given passage of text and corresponding question [3]. According to the answer type, existing machine reading comprehension tasks can be roughly divided into four categories: cloze style, multiple choice, span prediction, and free-form answer.

In recent years, many tasks in natural language processing have been framed as machine reading comprehension problem. Levy et al. [27] transformed the relation extraction into an MRC problem in zero-shot setting and achieved improvement. McCann et al. [41] cast 10 NLP tasks (e.g., machine translation, summarization, sentiment analysis, semantic role labeling, goal-oriented dialogue, and semantic parsing) as reading comprehension problem. Li et al. [29] reformalized the NER task as an MRC task to address the nested entities problem. Ma et al. [39] proposed a distant supervision-based MRC model for extractive summarization.

In addition, multi-turn MRC has also been introduced to solve various NLP tasks. Li et al. [34] cast the entity-relation extraction as a multi-turn MRC problem and encoded class information into question query. Li et al. [28] performed trigger identification, trigger classification, and argument extraction as multi-turn MRC in a pipelined fashion to solve event extraction task. Du and Cardie [16] explored different question generation strategies for event extraction. Liu et al. [36] explored an unsupervised question generation process for event extraction. Mao et al. [40] constructed two MRC problems to solve the aspect-based sentiment analysis problem through end-to-end training. Chen et al. [4] proposed a bidirectional machine reading comprehension framework to solve the aspect sentiment triplet extraction task.

While multi-turn MRC methods have been demonstrated to be effective for the event extraction task and aspect-based sentiment analysis task, it has not been applied to the ECPE task. Compared with these multi-turn MRC methods, our method has several technical differences. First, due to the task difference, our method is designed specific to the clause level, while previous methods work on word level. Thus, the structure of model is different and we validate the effectiveness of the clause-level MRC model for the ECPE task. Second, previous methods do not consider the error accumulation problem in multi-turn MRC framework, while we consider this problem and propose a consistent training strategy to deal with it. Third, different from previous bidirectional MRC methods, we comprehensively explore how to better combine the results from two directions in the inference phase.

3 APPROACH

We first present the task definition of ECPE. Then, we introduce the proposed **Consistent Dual-MRC (CD-MRC)** framework, followed by its technical details.

3.1 Task Definition

In the ECPE task, the input is a document d, which consists of multiple clauses $d = [c_1, c_2, \ldots, c_N]$ and each clause consists of multiple tokens $c_i = [t_1^i, t_2^i, \ldots, t_{n_i}^i]$. N and n_i are the number of clauses in document d and tokens in clause c_i, respectively. The goal of ECPE task is to extract all emotion-cause pairs in the document d at the clause level:

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

where c^e is an emotion clause and c^c is the corresponding cause clause.

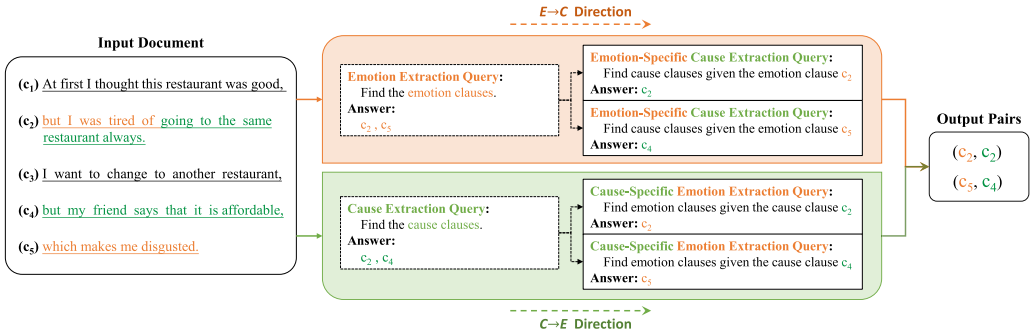


Fig. 2. The illustration of our proposed consistent dual-MRC framework.

3.2 Overview of Consistent Dual-MRC Framework

To deal with the ECPE task, we cast it as a two-turn **machine reading comprehension (MRC)** task and propose a consistent dual-MRC framework to extract emotion-cause pairs in a dual-direction way. As shown in Figure 2, we extract emotion-cause pairs from two directions, i.e., E-C direction and C-E direction. In either direction, we conduct two-turn MRC to identify emotion clauses and cause clauses from the context (i.e., the input document) in a question-answering fashion. Specifically, in E-C direction, we first extract emotion clauses based on the emotion extraction query. Then, for each extracted emotion clause, we extract its corresponding cause clauses based on the emotion-specific cause extraction query. Similarly, in C-E direction, we first extract cause clauses based on the cause extraction query. Then, for each extracted cause clause, we extract its corresponding emotion clauses based on the cause-specific emotion extraction query. To ensure these four subtasks can be learned jointly, we perform them on the same MRC model. Finally, the emotion-cause pairs extracted in both directions are combined to form the final set of emotion-cause pairs.

3.3 Query Construction

In our consistent dual-MRC framework, we need to construct four types of queries corresponding to the aforementioned four subtasks: emotion extraction query, cause extraction query, emotion-specific cause extraction query, and cause-specific emotion extraction query. These four types of queries fall into two forms: target-free static query and target-specific dynamic query. Among them, the target-free static query is used in the first turn and only contains static template text and no target. Both the emotion extraction query and cause extraction query belong to the target-free static query. The target-specific dynamic query is used in the second turn and contains both static template text and dynamic target content. Both the emotion-specific cause extraction query and cause-specific emotion extraction query belong to the target-specific dynamic query. All these queries can be constructed by the query templates.

Specifically, as shown in Figure 2, we construct the emotion extraction query and emotion-specific cause extraction query to extract emotion-cause pairs in the E-C direction:

- **Emotion extraction query:** The template of emotion extraction query is a static sentence: "Find the emotion clauses."
- **Emotion-specific cause extraction query:** The template of emotion-specific cause extraction query is a dynamic sentence: "Find cause clauses given the emotion clause c_i," where c_i refers to the content of the ith clause. For example, the complete query of the top emotion-specific cause extraction query in Figure 2 is "Find cause clauses given the emotion clause *but I was tired of going to the same restaurant always.*"

Similarly, the cause extraction query and cause-specific emotion extraction query in the C-E direction are designed as follows:

- **Cause extraction query:** The template of cause extraction query is a static sentence "Find the cause clauses."
- **Cause-specific emotion extraction query:** The template of cause-specific emotion extraction query is a dynamic sentence "Find emotion clauses given the cause clause c_i," where c_i refers to the content of the ith clause. For example, the complete query of the top cause-specific emotion extraction query in Figure 2 is "Find cause clauses given the emotion clause *but I was tired of going to the same restaurant always*."

For the above queries, their template texts are all natural language texts, so we can denote their templates as natural language query templates. Considering that the template text may be not necessarily grammatical, we can also construct some ungrammatical query templates by replacing the template text of natural language query template with ungrammatical text. For example, the natural language template text in the above four kinds of queries can be replace by "emotion," "emotion cause," "cause," and "cause emotion," respectively, to form the corresponding ungrammatical query templates.

3.4 Consistent Dual-MRC Model

We then describe the details of the consistent dual-MRC model adopted in the framework, which is structured with an encoder and the prediction layers. Based on the four types of queries, the model can be jointly trained, which can be finally used to infer the ECPE results.

3.4.1 Encoder. We adopt BERT [12] as model backbone to get the representation of each clause. Following the previous studies [15, 37, 49], we feed the entire document $d = (c_1, c_2, \ldots, c_N)$ into BERT to get the representation of each clause. As shown in Figure 3, to get the representation of each clause, we insert a [CLS] and [SEP] token for every clause $c_i = (t^i_1, t^i_2, \ldots, t^i_{n_i})$ to get the input of BERT (i.e., c_i = ([CLS],$t^i_1, t^i_2, \ldots, t^i_{n_i}$,[SEP])). To distinguish clauses in a document, we assign interval segment embedding (E_A, E_B, E_A, ...) to each clause where E_A is assigned to clauses at odd positions and E_B to those at even positions. Thus, for each token in the input, its input representation is the sum of the corresponding token embedding, segment embedding, and position embedding. The position embedding is same as BERT. Then, we use attention mechanism [2] to get the clause representation. Specifically, the clause representation of the clause c_i is denoted as h_i:

$$h_i = \sum_j a^i_j h^i_j, \tag{1}$$

$$a^i_j = \frac{exp((h^i_j)^T w_t)}{\sum_p exp((h^i_p)^T w_t)}, \tag{2}$$

where h_i is representation of clause c_i, h^i_j is hidden state representation of token t^i_j, a^i_j is the attention weight of token t^i_j, and w_t is a randomly initialized weight vector.

As shown in Figure 3, the BERT encoder takes the query and a document as input and outputs the representation matrix $H = \{h_0, h_1, \ldots, h_N\} \in \mathbb{R}^{(N+1)\times m}$, where N is the number of clauses in the document and m is the vector dimension of the last layer of BERT. Then, we feed the representation to BiLSTM to get a contextualized representation. Since there are four types of queries in our method, we use $H^e = \{h^e_0, h^e_1, \ldots, h^e_N\}$, $H^c = \{h^c_0, h^c_1, \ldots, h^c_N\}$, $H^{ec} = \{h^{ec}_0, h^{ec}_1, \ldots, h^{ec}_N\}$, $H^{ce} = \{h^{ce}_0, h^{ce}_1, \ldots, h^{ce}_N\} \in \mathbb{R}^{(N+1)\times m}$ to denote outputs of BiLSTM corresponding to four types of

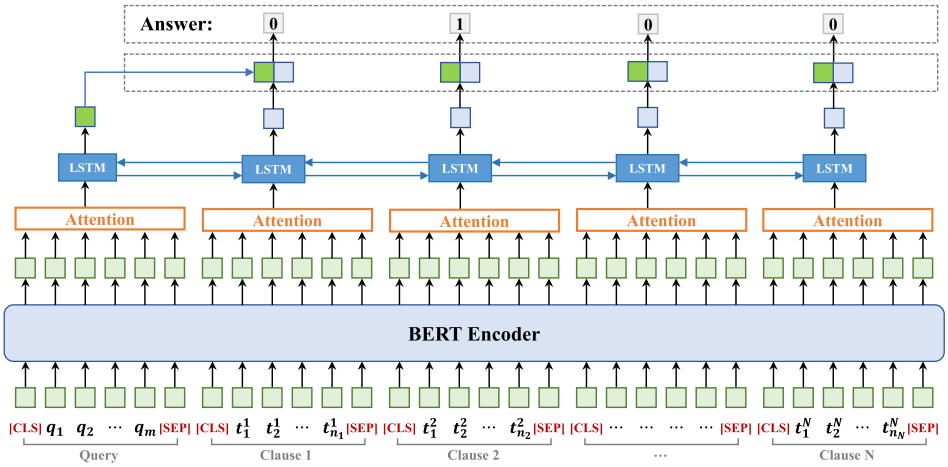


Fig. 3. The illustration of the proposed consistent dual-MRC model.

queries, where h_i^e, h_i^c, h_i^{ec}, and h_i^{ce} refer to the contextualized representation of the ith clause corresponding to the case of using emotion extraction query, cause extraction query, emotion-specific cause extraction query, and cause-specific emotion extraction query, respectively.

3.4.2 Prediction. After obtaining the contextualized representations in the document, we make a prediction on each clause to answer the input query. In the E-C direction, we concatenate query representation and each clause representation and use a binary classifier to predict emotion clauses and corresponding cause clauses. Specifically, the probability of the ith clause to be an emotion clause is denoted as p_i^e and the probability of the ith clause to be a cause clause of the jth emotion clause is denoted as $p_i^{ec_j}$:

$$\begin{aligned} p_i^e &= softmax(g_i^e) \\ g_i^e &= W^1(h_i^e \oplus h_0^e) + b^1, \end{aligned} \tag{3}$$

$$\begin{aligned} p_i^{ec_j} &= softmax(g_i^{ec_j}) \\ g_i^{ec_j} &= W^1(h_i^{ec_j} \oplus h_0^{ec_j}) + b^1, \end{aligned} \tag{4}$$

where $W^1 \in \mathbb{R}^{2\times 2m}$, $b^1 \in \mathbb{R}^2$ are model parameters, $\oplus$ is the concatenation operator, $h_0^e \in H^e$ is the contextualized representation of emotion extraction query, and $h_0^{ec_j} \in H^{ec}$ is the contextualized representation of emotion-specific cause extraction query.

Similarly, in the C-E direction, we also concatenate query representation and each clause representation and use a binary classifier to predict emotion clauses and corresponding cause clauses. Specifically, the probability of the ith clause to be a cause clause is denoted as p_i^c and the probability of the ith clause to be an emotion clause of the jth cause clause is denoted as $p_i^{ce_j}$:

$$\begin{aligned} p_i^c &= softmax(g_i^c) \\ g_i^c &= W^1(h_i^c \oplus h_0^c) + b^1, \end{aligned} \tag{5}$$

$$\begin{aligned} p_i^{ce_j} &= softmax(g_i^{ce_j}) \\ g_i^{ce_j} &= W^1(h_i^{ce_j} \oplus h_0^{ce_j}) + b^1, \end{aligned} \tag{6}$$

where $h_0^c \in H^c$ is the contextualized representation of cause extraction query, $h_0^{ce_j} \in H^{ce}$ is the contextualized representation of cause-specific emotion extraction query. It is worth noting that

for all four types of queries, we use the same model (i.e., BERT, LSTM, and a binary classifier) to classify all clauses.

3.5 Joint Training

We then introduce the details of two loss functions (i.e., dual loss and consistent loss) and the overall loss function, and describe the training flow of how to jointly train our model.

3.5.1 Dual Loss. We first use dual loss to train the model by combining the loss of both directions. For a document, the loss function in the E-C direction $\mathcal{L}^{EC}$ is the sum of two losses corresponding to the emotion extraction query and emotion-specific cause extraction query:

$$\mathcal{L}^{EC} = -\sum_{i=1}^{N} y_i^e \log(p_i^e) - \sum_{j \in L_e} \sum_{i=1}^{N} y_i^{ec_j} \log(p_i^{ec_j}), \tag{7}$$

where N is the number of clauses in this document, y_i^e is emotion label of clause c_i, L_e is the index set of ground-truth emotion clauses, $y_i^{ec_j}$ is the corresponding cause label of clause c_i given emotion clause c_j.

Similarly, the loss function in the C-E direction $\mathcal{L}^{CE}$ is the sum of two losses corresponding to the cause extraction query and cause-specific emotion extraction query:

$$\mathcal{L}^{CE} = -\sum_{i=1}^{N} y_i^c \log(p_i^c) - \sum_{j \in L_c} \sum_{i=1}^{N} y_i^{ce_j} \log(p_i^{ce_j}), \tag{8}$$

where N is the number of clauses in this document, y_i^c is cause label of clause c_i, L_c is the index set of ground-truth cause clauses, and $y_i^{ce_j}$ is the corresponding emotion label of clause c_i given cause clause c_j.

The dual loss of our model is the sum of E-C direction and C-E direction:

$$\mathcal{L}^{DUAL} = \mathcal{L}^{EC} + \mathcal{L}^{CE}. \tag{9}$$

3.5.2 Consistent Loss. It is worth noting that, during training, for the second turn of MRC, the ground-truth sets of emotion clauses and cause clauses are available for the generation of emotion-specific cause extraction query and cause-specific emotion extraction query. However, during inference, the ground-truth sets of emotion clauses and cause clauses are unavailable and emotion-specific cause (cause-specific emotion) extraction queries should be generated based on the results of the first turn. Thus, the second-turn queries at training and inference are drawn from different distributions. This discrepancy, called *exposure bias* [43], leads to a gap between training and inference. This gap will make the model unable to handle the first-turn error in the second-turn query at inference, and thus easily lead to error accumulation.

To address the *exposure bias* issue, we propose a consistent training strategy based on negative sampling, which can make the training and inference phases more consistent. Specifically, for each document, we set a probability α to randomly sample a non-emotion clause as a pseudo emotion clause to generate a pseudo emotion-specific cause extraction query. For example, for the input document in Figure 2, a pseudo emotion-specific cause extraction query can be "Find cause clauses given the emotion clause *but my friend says that it is affordable*," where the clause "*but my friend says that it is affordable*" is not an emotion clause. This step simulates the situation that an incorrect emotion clause is extracted in the first turn and then used to generate the second-turn query. To teach the model how to handle such situation, the pseudo emotion clause is set to

have no corresponding cause clauses. Similarly, we also set a probability α[1] to randomly sample a non-cause clause as a pseudo cause clause to generate a pseudo cause-specific emotion extraction query. So the consistent loss of our model is:

$$\mathcal{L}^{CON} = -\sum_{j \in L_e^{'}} \sum_{i=1}^{N} y_i^{ec_j} \log(p_i^{ec_j}) - \sum_{j \in L_c^{'}} \sum_{i=1}^{N} y_i^{ce_j} \log(p_i^{ce_j}), \tag{10}$$

where $L_e^{'}$ is the index set of pseudo emotion clause, $L_c^{'}$ is the index set of pseudo cause clause, $y_i^{ec_j}$ is the corresponding cause label of clause c_i given pseudo emotion clause c_j (i.e., no corresponding cause clauses), and $y_i^{ce_j}$ is the corresponding emotion label of clause c_i given pseudo cause clause c_j (i.e., no corresponding emotion clauses).

3.5.3 Overall Loss Function. The final loss of our model for a document is the combination of the dual loss and the consistent loss:

$$\mathcal{L} = \mathcal{L}^{DUAL} + \mathcal{L}^{CON} + \lambda \|\theta\|^2, \tag{11}$$

where λ is the coefficient of L_2-norm regularization, and θ denotes all the parameters in this model.

3.5.4 Training Flow. To train our CD-MRC model, we need first convert the ECPE training set to MRC-style training set and then randomly sample mini-batches from the MRC-style training set to train model. The detailed training process of our model is illustrated in Algorithm 1.

As shown in the algorithm, for each document in the ECPE dataset, we can construct a total of six types of instances, four of which correspond to the aforementioned four types of queries, while the other two correspond to the consistent strategy. Each MRC-style instance consists of a query and a document, and its label is a sequence of numbers indicating whether the corresponding clause in the document is an answer to the query. For the first-turn query, we can construct the emotion query instance and cause query instance based on our designed target-free static queries. Similarly, for the second-turn query, we can construct the emotion-specific cause query instance and cause-specific emotion query instance based on our designed target-specific dynamic queries. It should be noted that the construction of pseudo second-turn instances is decided by a probability. Thus, they may not always be constructed for a document. Finally, after constructing instances for all documents, we shuffle the MRC-style training set and randomly sample instances for model training.

3.6 Inference

During inference, the ground-truth sets of emotion clauses and cause clauses are unavailable. Thus, we need to generate the queries of the second turn of MRC based on the results of the first turn. The complete inference process of our model is illustrated in Algorithm 2.

3.6.1 Set Combination. It shows that both directions follow a similar pipeline and can generate a candidate set of emotion-cause pairs. To combine these two candidate sets into a final set, here, we consider four combination strategies. An intuitive idea to fuse two sets is to take the *intersection* or *union* of them as the final results:

- *Intersection*: Only when the emotion-cause pair is predicted in both directions, it can be viewed as a valid pair.
- *Union*. When an emotion-cause pair is predicted in either direction, it can be viewed as a valid pair.

[1]Of course, we can set a new hyper-parameter to control this probability. But to not introduce additional hyper-parameter, we set the same probabilities for both directions.

ALGORITHM 1: The Training Flow of Consistent Dual-MRC Model

Require: The original ECPE training set $\mathcal{D} = \{\ldots, (d, P = \{\ldots, (c^e, c^c), \ldots\}), \ldots\}$
Ensure: A well-trained CD-MRC model
Initialize the MRC-style training set $\mathcal{D}_{tr} = \emptyset$
 # Convert the original ECPE training set $\mathcal{D}$ to MRC-style training set $\mathcal{D}_{tr}$:
for each $(d, P = \{\ldots, (c^e, c^c), \ldots\})$ in $\mathcal{D}$ **do**
 # Construct the first-turn MRC-style instances for both directions:
Construct an emotion query instance $x^e = (q^e, d)$ and its label sequence y^e
Construct a cause query instance $x^c = (q^c, d)$ and its label sequence y^c
Add (x^e, y^e) and (x^c, y^c) to $\mathcal{D}_{tr}$
 # Construct the second-turn MRC-style instances for both directions:
for each emotion clause c^e in P **do**
Construct an emotion-specific cause query instance $x^{ec} = (q^{ec} \oplus c^e, d)$ and its label sequence y^{ec}
Add (x^{ec}, y^{ec}) to $\mathcal{D}_{tr}$
end for
for each cause clause c^c in P **do**
Construct a cause-specific emotion query instance $x^{ce} = (q^{ce} \oplus c^c, d)$ and its label sequence y^{ce}
Add (x^{ce}, y^{ce}) to $\mathcal{D}_{tr}$
end for
 # Construct pseudo second-turn instances for consistent loss:
Randomly sample a non-emotion clause c^{ne} and a non-cause clause c^{nc} from d with probability α
Construct a pseudo emotion-specific cause query instance $z^{ec} = (q^{ec} \oplus c^{ne}, d)$
Construct a pseudo cause-specific emotion query instance $z^{ce} = (q^{ce} \oplus c^{nc}, d)$
Construct a zero-valued label sequence y^o
Add (z^{ec}, y^o) and (z^{ce}, y^o) to $\mathcal{D}_{tr}$
end for
 # Train the CD-MRC model:
for all iteration = 1, ..., MaxIter **do**
Randomly sample a mini-batch data from $\mathcal{D}_{tr}$ and calculate loss based on Equation (9)
Obtain derivative and update the CD-MRC model
end for
return The well-trained model

Among the above two strategies, the intersection strategy ignores pairs that only identified in one direction, even if they have high probabilities to be valid pairs. In contrast, union strategy usually introduce many wrong emotion-cause pairs, resulting in lower precision. Thus, we consider two more flexible strategies:

- *Harmonic*: We consider pairs to be reliable if they are extracted by both directions or by one direction with a high probability. We introduce a hyper-parameter β as the threshold of high probability. When β equals to 0.5 (0.5 is the minimum positive probability of binary classification), this strategy is equivalent to the *Union* strategy. When β equals to 1 (1 is the maximum probability of binary classification), this strategy is equivalent to the *Intersection* strategy.
- *Complementary*: We consider the pairs to be reliable if they are extracted by E-C direction, since the performance of E-C direction is often better than C-E direction. Besides, we

ALGORITHM 2: The Inference Algorithm of Consistent Dual-MRC Model

Require: A document d, a sentiment dictionary S
Ensure: A set of pairs $P = \{(e_1, c_1), (e_2, c_2), \ldots, (e_m, c_m)\}$
Initialize $P = \emptyset$, $P_{EC} = \emptyset$, $P_{CE} = \emptyset$
 # E-C Direction:
Construct emotion extraction query q_e
Take q_e and d as input to get the emotion clauses set L_e
if L_e is not empty **then**
  Construct emotion-specific cause extraction queries q_{ec}
  Take q_{ec} and d as input to get the emotion-cause pair set P_{EC}
end if
 # C-E Direction:
Construct cause extraction query q_c
Take q_c and d as input to get the cause clauses set L_c
if L_c is not empty **then**
  Construct cause-specific emotion extraction queries q_{ce}
  Take q_{ce} and d as input to get the emotion-cause pair set P_{CE}
end if
 # Set Combination:
Combine the two candidate pair sets P_{EC} and P_{CE} to get P
 # Emotion Filtering:
for each pair (e_i, c_i) in P **do**
  if all words of emotion clause e_i are not emotion words defined by S **then**
    Remove the pair from P
  end if
end for
return P

also use the high-probability pairs of the C-E direction as the supplement and introduce a hyper-parameter γ as the threshold of high probability.

3.6.2 Emotion Filtering. After obtaining the combined pair set, we further employ an emotion filtering strategy to filter pairs with obviously invalid emotion clause. To judge whether an emotion clause is valid, we consider a necessary condition, that is, the emotion clause should contain at least one emotion word. To implement the emotion filtering strategy, we use a sentiment dictionary to define the emotion words and consider a pair to be valid only if its emotion clause contains emotion words.

4 EXPERIMENTS

In this section, we first introduce the datasets and evaluation metric. Then, we illustrate the experimental settings, implementation details, performance comparison, ablation study, model analysis, case study, and discussion.

4.1 Datasets and Evaluation Metrics

We conduct experiments on two benchmark datasets: a Chinese dataset[2] [51] and an English dataset[3] [44]. The Chinese dataset is constructed based on the benchmark ECE corpus [24] and

[2]https://github.com/NUSTM/ECPE.
[3]https://github.com/Aaditya-Singh/E2E-ECPE.

Table 1. Statistics of Two Datasets

Statistics	Chinese	English
Number of documents	1,945	2,843
Number of pairs	2,167	3,215
Average number of clauses per document	14.77	7.69
Number of documents with one pair	1,746	2,537
Number of documents with two pairs	177	256
Number of documents with more than two pairs	22	50

For convenience, we use pair to represent emotion-cause pair.

has 1,945 documents collected from SINA city news. The English dataset has 2,843 documents collected from novels [21]. There is at least one emotion-cause pair in each document for two datasets. Detailed statistics are listed in Table 1.

We use the **pair-level precision (P), recall (R)**, and F1 score as evaluation metrics defined by Xia and Ding [51] for the ECPE task:

$$P = \frac{\sum correct_{pairs}}{\sum predicted_{pairs}}, \tag{12}$$

$$R = \frac{\sum correct_{pairs}}{\sum annotated_{pairs}}, \tag{13}$$

$$F1 = \frac{2 \cdot P \cdot R}{P + R}, \tag{14}$$

where $predicted_{pairs}$ denotes the pairs predicted by the model, $annotated_{pairs}$ denotes the pairs that are labeled in the dataset, and the $correct_{pairs}$ means the correct pairs predicted by the model. In addition, we evaluate the performance of two subtasks: emotion extraction and cause extraction.

4.2 Experimental Settings

There are two experimental settings adopted by previous studies on the Chinese dataset. To make a fair comparison with the previous method, we conduct experiments on both settings to ensure the comparisons are made under the same experimental settings. In the first setting, we use the same data split adopted by Xia and Ding [51], where the dataset is split into two parts: 90% for training and the remaining 10% for testing. In the second setting, we use the same data split[4] adopted by Fan et al. [18], where the dataset is divided into a training/development/test set in a ratio of 8:1:1. We test the model with best performance on the development set. For the English dataset, we use the same data split adopted by Singh et al. [44], where the dataset is divided into the training/development/test set in a ratio of 8:1:1. We also test the model with best performance on the development set. To obtain statistically credible results, we repeat the experiments 20 times and report the average results to get reliable results on all settings. For the following experiments, unless otherwise specified, the Chinese dataset with the first setting is used as the default experimental setting.

4.3 Implementation Details

We adopt BERT-Base[5] [12] as model backbone in our work. We use AdamW [38] optimizer and the batch size is set to be 8. The learning rate for BERT and other parameters (i.e., BiLSTM and

[4]https://github.com/HLT-HITSZ/TransECPE.
[5]https://github.com/huggingface/pytorch-pretrained-BERT.

fully connected layer) are set to be 2e-5 and 1e-4. The hidden state of BiLSTM is set to be 100 with 1 layer. The coefficient of L_2 term is 1e-5. α is set to be 0.1. We schedule the learning rate that the first 10% of all training steps is a linear warmup phrase and then the rest is a linear decay phrase. We train our model for 20 epochs with early stopping. We use ANTUSD[6] [48] as the sentiment dictionary for the Chinese dataset and do not employ emotion filtering strategy for the English dataset. The natural language query is used by default.

4.4 Comparison Methods

To evaluate the effectiveness of our method, we compare our method with two groups of baselines. The first group of baselines is the two-step model proposed by Xia and Ding [51]:

- **Indep** [51] extracts the emotions and the causes independently in the first step, then pairs the extracted emotions and causes and filters out the invalid emotion-cause pairs in the second step.
- **Inter-CE** [51] is similar to **Indep**. The difference lies in the first step, where the prediction of cause extraction is used to improve emotion extraction.
- **Inter-EC** [51] is similar to **Indep**. The difference lies in the first step, where the prediction of emotion extraction is used to improve cause extraction.

The second group of baselines employs end-to-end neural network to address the ECPE task:

- **PExt** [44] is an end-to-end version of two-step model, which directly uses Cartesian Product to construct pair representation for classification.
- **E2EECPE** [45] predicts the relation between emotions and causes via biaffine attention.
- **MTNECP** [50] is a multi-task network that learns emotion extraction, cause extraction, and relation classification jointly.
- **SLSN** [11] is a symmetric local search network that uses a local search model to extract pairs based on sliding windows.
- **IE-CNN+CRF** [6] is a unified sequence labeling model based on stacked CNN and CRF and uses the type of emotion to pair.
- **Sequence** [54] is another sequence labeling model based on BERT and uses distance to pair.
- **TRANS** [18] is a transition-based model that transforms the ECPE task into a procedure of parsing-like directed graph construction.
- **ECPE-2D** [14] is a joint model that integrates the emotion-cause pair representation, interaction, and prediction.
- **UTOS** [10] proposes unified target-oriented sequence-to-sequence model for sequence labeling and uses pair index to pair.
- **RANKCP** [49] models inter-clause relationship through graph attention network and ranks the candidate emotion-cause pair to get final result.
- **Pair-GCN** [8] models dependency relations among local neighborhood candidate pairs through graph convolutional network.
- **Refinement** [19] is a sequence labeling model that uses the output of emotion extraction and cause extraction to refine the labels of emotion-cause pair extraction and uses distance to pair.
- **ISML** [15] employs the multi-label learning to extract pairs based on sliding window and is the best baseline.

[6] https://academiasinicanlplab.github.io/.

Table 2. The Performance of Our Model and the Baselines on the ECPE Task

Method	Emotion Extraction			Cause Extraction			Emotion-Cause Pair Extraction		
	P	R	F1	P	R	F1	P	R	F1
Indep	0.8375	0.8071	0.8210	0.6902	0.5673	0.6205	0.6832	0.5082	0.5818
Inter-CE	0.8494	0.8122	0.8300	0.6809	0.5634	0.6151	0.6902	0.5135	0.5901
Inter-EC	0.8364	0.8107	0.8230	0.7041	0.6083	0.6507	0.6721	0.5705	0.6128
E2EECPE	0.8595	0.7915	0.8238	0.7062	0.6030	0.6503	0.6478	0.6105	0.6280
MTNECP	0.8662	0.8393	0.8520	0.7400	0.6378	0.6844	0.6828	0.5894	0.6321
SLSN	0.8406	0.7980	0.8181	0.6992	0.6588	0.6778	0.6836	0.6291	0.6545
IE-CNN+CRF	0.8614	0.7811	0.8188	0.7348	0.5841	0.6496	0.7149	0.6299	0.6686
ECPE-2D†	0.8627	0.9221	0.8910	0.7336	0.6934	0.7123	0.7292	0.6544	0.6889
UTOS†	0.8815	0.8321	0.8556	0.7674	0.7320	0.7471	0.7389	0.7062	0.7203
RANKCP†	0.9123	0.8999	0.9057	0.7461	0.7788	0.7615	0.7119	0.7630	0.7360
ISML†	0.8608	0.9191	0.8886	0.7382	0.7912	0.7630	0.7700	0.7235	0.7452
Refinement†	0.8711	0.8178	0.8436	0.7947	0.7404	0.7666	0.7746	0.7199	0.7463
CD-MRC†	**0.9692***	**0.9398***	**0.9537***	**0.8101***	**0.8068***	**0.8077***	**0.8249***	**0.7800***	**0.8013***

We use the data split adopted by Xia and Ding [51]. All results in this table use the same experimental settings. The best results are in bold. *indicates statistical significance ($p < 0.01$) by comparing with Refinement and ISML in paired t-tests. †denotes that BERT is used as the encoder of the model.

Table 3. The Performance of Our Model and the Baselines on the ECPE Task

Method	Emotion Extraction			Cause Extraction			Emotion-Cause Pair Extraction		
	P	R	F1	P	R	F1	P	R	F1
Sequence†	0.8196	0.7329	0.7739	0.7490	0.6602	0.7018	0.7243	0.6366	0.6776
TRANS†	0.8716	0.8244	0.8474	0.7562	0.6471	0.6974	0.7374	0.6307	0.6799
UTOS†	0.8649	0.8293	0.8491	0.7418	0.7084	0.7281	0.7104	0.6812	0.6907
RANKCP†	0.8936	0.8948	0.8942	0.6940	0.7471	0.7191	0.6575	0.7305	0.6915
Refinement†	0.8593	0.7993	0.8282	0.7614	0.7039	0.7315	0.7377	0.6802	0.7078
Pair-GCN†	0.8857	0.7958	0.8375	**0.7907**	0.6928	0.7375	0.7672	0.6791	0.7202
ISML†	0.8465	0.8990	0.8717	0.7051	0.7704	0.7358	0.7488	0.6976	0.7220
CD-MRC†	**0.9592***	**0.9183***	**0.9381***	0.7789	**0.7616***	**0.7694***	**0.7739***	**0.7478***	**0.7598***

We use the data split adopted by Fan et al. [18]. All results in this table use the same experimental settings. The best results are in bold. *indicates statistical significance ($p < 0.01$) by comparing with ISML in paired t-tests. †denotes that BERT is used as the encoder of the model.

4.5 Performance Comparison

RQ1: To validate the effectiveness of our method, we conduct experiments on two datasets and compare our method with the existing ECPE methods. Tables 2 and 3 present the performance of our proposed method and baseline models on the Chinese dataset under two experimental settings, respectively. Table 4 presents the performance of our proposed method and baseline models on the English dataset.

First, we observe the performance of the baseline models. We can find that the end-to-end neural network model outperforms the two-step model. This shows that end-to-end training can improve performance and avoid error accumulation problem. In addition, we can see that models that use BERT as encoder can achieve better performance. This is because models that use BERT as encoder can get better clause representation.

Second, we can find that our method achieves the best performance over all metrics for the emotion-cause pair extraction. Our method outperforms the best baseline method **ISML** by 5.61% and 3.78% in F1 under both experimental settings, as shown in Tables 2 and 3, respectively. Our method also outperforms **ISML** by 2.32% in F1 under the same experimental setting on the English dataset, as shown in Table 4. Our method also outperforms method **Refinement** by 5.50% and 5.20% in F1 under both experimental settings, as shown in Tables 2 and 3, respectively. The

Table 4. The Performance of Our Model and the Baselines on the ECPE Task

Method	Emotion Extraction			Cause Extraction			Emotion-Cause Pair Extraction		
	P	R	F1	P	R	F1	P	R	F1
Inter-EC	0.6741	0.7160	0.6940	0.6039	0.4734	0.5301	0.4694	0.4102	0.4367
PExt	0.7163	0.6749	0.6943	**0.6636**	0.4375	0.5226	0.5134	0.4929	0.5017
ECPE-2D	0.7435	0.6968	0.7189	0.6491	0.5353	0.5855	0.6049	0.4384	0.5073
ISML	**0.7546**	0.6996	0.7225	0.6350	0.5919	0.6110	0.5926	0.4530	0.5121
CD-MRC	0.7155	**0.7585***	**0.7347***	0.6282	**0.6509***	**0.6388***	**0.6065***	0.4621*	**0.5243***

We use the data split adopted by Singh et al. [44]. All results in this table use the same experimental settings. The best results are in bold. *indicates that the performance improvement over ISML is statistically significant ($p < 0.05$) by comparing with ISML in paired t-tests.

improvements are significant with $p < 0.01$ by comparing with **Refinement** and **ISML** in paired t-tests. This demonstrates that our adopted target-specific question-answering fashion can effectively capture the pairing relationship between emotions and causes.

Finally, for the two subtasks (i.e., emotion extraction and cause extraction), we can find that our method also achieves the best performance under both experimental settings. Our method outperforms the best baseline method **ISML** by 4.80% on emotion extraction and 4.11% on cause extraction in F1, as shown in Table 2. Our method also outperforms **ISML** by 1.22% on emotion extraction and 2.78% on cause extraction in F1, as shown in Table 4. This shows the effectiveness of our proposed method. Considering that our adopted emotion filtering operation may contribute a lot to the performance on emotion extraction task, the high performance on cause extraction task can be attributed to the query of cause extraction. This indicates that the question-answering fashion adopted in our consistent dual-MRC framework provides a better way of the identifying of cause than baseline methods.

4.6 Ablation Study

RQ2: To validate the contribution of each component to our method, we report the performance of removing each of them from our method individually. These components include: dual-direction training, two-turn querying, emotion filtering, and consistent loss. To reduce the randomness of the reported performance, we run the ablation variants under 10 different random seeds and report the average performance. Besides, we further conduct paired t-tests between CD-MRC and each of its ablation variants to validate whether the performance drop brought by ablating each design from CD-MRC is statistically significant.

First, we remove the dual framework and only train the model in one direction (i.e., we only use data in one direction). As shown in Table 5, we can find that after removing one direction (i.e., w/o E-C Direction and w/o C-E Direction), the F1 score decreases from 0.8013 to 0.7698 and 0.7767, respectively, which are still better than that of the baseline methods. This indicates that a simple one-direction MRC framework is also effective, and combining the two directions will further improve the performance. Second, we ablate the second turn of the consistent dual-MRC framework (i.e., w/o Second Turn). Here, we use emotion extraction query and cause extraction query to extract emotion clauses and cause clauses, and we use Cartesian product strategy [51] as the alternative of the second turn for the pairing of emotions and causes extracted by the first turn. We can observe a significant drop of performance (i.e., from 0.8013 to 0.7327 in F1) brought by removing the second turn. This indicates that the target-specific queries in the second turn are effective for the pairing of emotions and causes. Third, we further ablate the emotion filtering strategy from the framework (i.e., w/o Emotion Filtering). We can find that the emotion filtering strategy is effective and it can greatly improve the precision. This is because the filtering strategy

Table 5. Ablation Study of the CD-MRC Framework on the ECPE Task

Model Setting	P	R	F1
CD-MRC	**0.8249**	0.7800	**0.8013**
w/o E-C Direction	0.8003	0.7450	0.7698*
w/o C-E Direction	0.8014	0.7562	0.7767*
w/o Second Turn	0.7426	0.7260	0.7327*
w/o Emotion Filtering	0.7593	0.7540	0.7588*
w/o Consistent Loss	0.8005	**0.7899**	0.7934*

The ablation study is repeated 10 times with 10 different random seeds and the average results are reported. *indicates that the performance drop of F1-score brought by ablating the corresponding design is statistically significant ($p < 0.05$) in paired t-tests.

filters many wrong pairs. It is also worth noting that even if the emotion filtering strategy is not used, the performance of consistent dual-MRC framework is still better than that of the baseline methods in Table 2. Finally, we ablate the consistent loss from the framework (i.e., w/o Consistent Loss). We can see that the F1 score decreases from 0.8013 to 0.7934, and the performance drop is statistically significant. This shows that the consistent loss makes the training and inference phases more consistent. It is worth noting that the precision drops significantly and the recall rises. This may be because when there is no consistent loss, all emotion-specific cause extraction queries on the dataset have at least one corresponding cause clause, and the model tends to predict at least one cause clause for each emotion-specific cause extraction query. When there is a consistent loss, some emotion-specific cause extraction queries on the dataset do not have corresponding cause clauses, and the model will more consider the relationship between the query clause and the candidate clauses.

4.7 Model Analysis

In this part, we analyze the effect of query designment, combination strategy, probability thresholds, consistent training strategy, and pair number on the performance of our model and show the model size.

4.7.1 Effect of Query Designment. **RQ3:** Considering that different query templates may bring different effects to the MRC model, we first explore the effect of query designment on the performance of our model. To this end, we consider six styles of query templates: *natural language template text (Find)*, *natural language template text (Which)*, *natural language template text (Where)*, *ungrammatical template text (phrase)*, *without target content*, and *only target content*. Among these styles, while *natural language template text (Find)* and *ungrammatical template text (phrase)* are introduced in the method, here, we introduce other four styles of query templates. For the *natural language template*, we consider extra two kinds of template texts with different keywords: which and where. Take the emotion extraction query as an example: The three kinds of natural language template texts are "Find the emotion clauses," "Which are the emotion clauses," and "Where are the emotion clauses," respectively. *without target content* means that we do not encode the target content into query, but as a result, the second-turn query cannot be constructed. Therefore, *without target content* is equivalent to "w/o Second Turn" in Table 5 (i.e., we ablate the second turn in our framework and use Cartesian product to pair emotions and causes extracted by the first turn). *only content query* can be constructed by setting template text as blank and only keeping the content of target clause in the second-turn query (e.g., in this setting, the emotion-specific cause extraction query in Figure 2 is *"but I was tired of going to the same restaurant always"*).

Table 6. Effect of Using Different Query Templates on the Performance of ECPE Task

Query Template	P	R	F1
Natural Language Template Text (Find)	0.8249	**0.7800**	**0.8013**
Natural Language Template Text (Which)	0.8256	0.7756	0.7997
Natural Language Template Text (Where)	0.8311	0.7705	0.8007
Ungrammatical Template Text (phrase)	**0.8395**	0.7604	0.7977
Without Target Content	0.7426	0.7260	0.7327
Only Target Content	0.8252	0.7697	0.7953

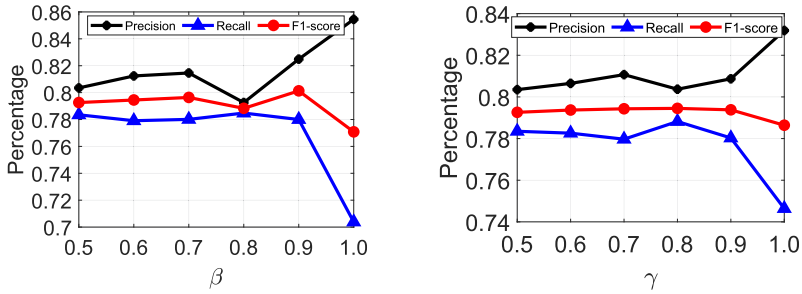
Table 7. Effect of Using Different Combination Strategies on the Performance of ECPE Task

Strategy	P	R	F1
Intersection	**0.8545**	0.7038	0.7708
Union	0.8035	0.7835	0.7926
Harmonic	0.8249	0.7800	**0.8013**
Complementary	0.8037	**0.7882**	0.7945
Only E-C Results	0.8319	0.7464	0.7864
Only C-E Results	0.8120	0.7421	0.7739

First, as shown in Table 6, we can find that compared to the default template *natural language template text (Find)*, the performance of *without target content* drops a lot (e.g., from 0.8013 to 0.7327). This indicates that it is necessary to use clause content in second turn for pairing. Second, we can find that the performance of *natural language template text (Find)* and *ungrammatical template text (phrase)* is better than that of *only target content*. This indicates that the information contained in template text can be perceived by the CD-MRC model and the task-related template text is very helpful to improve model performance. The reason behind may be that when the template text contains task-related information, it can be used as an indicator of subtasks, thus the encoder can be aware of the current subtask and encode the document specific to it. It is worth noting that the encoder is shared by all subtasks but task-related template text enables the same document to be encoded into different representations. Finally, among the task-related template texts (i.e., *natural language template text* and *ungrammatical template text*), we can find that the three styles of *natural language template text* achieve similar performance and they slightly outperform *ungrammatical template text*. This suggests that more natural text leads to better performance.

4.7.2 Effect of Combination Strategy. **RQ4:** Considering that the predictions from both directions may be helpful to get better final predictions, we then explore the effect of using different combination strategies (i.e., intersection, union, harmonic, complementary) and not using combination strategy (i.e., just results in single direction) on our method.

As shown in Table 7, we can find that even without integrated prediction, our model performs well in E-C direction (i.e., F1 of Only E-C Results is 0.7864) and C-E direction (i.e., F1 of Only C-E Results is 0.7739). The performance in E-C direction is better than that in C-E direction. This is because the cause extraction is more difficult than emotion extraction (i.e., with lower F1 score), and when cause extraction is used as the first-turn task, more errors would propagate to the second-turn task. It is worth noting that the result of E-C direction in Table 7 (i.e., 0.7864) is higher than that of w/o C-E direction in Table 5 (i.e., 0.7767). This shows that the data in the other direction is helpful to improve the generalization of the model. Besides, we can find that the performance of using the union strategy exceeds the performance of the E-C direction. This indicates that simply combining the results of two directions by union can improve the performance. We also find



(a) The effect of β (b) The effect of γ

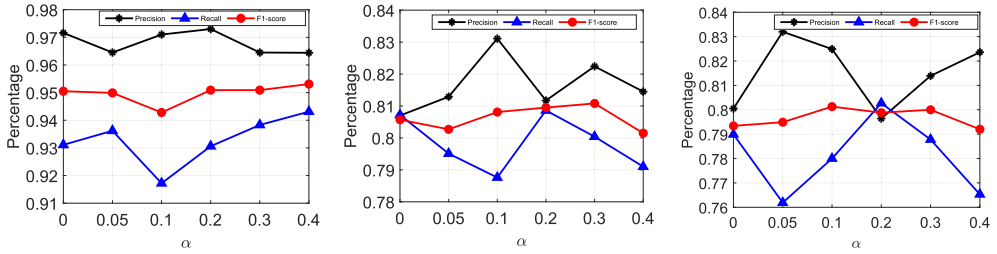
Fig. 4. Effect of β and γ in harmonic and complementary strategies on the performance of ECPE task.

that complementary strategy and harmonic strategy improves performance and harmonic strategy achieves the best performance. This shows that the two more flexible strategies are more effective and setting a threshold to filter pairs in both directions is effective. It is worth noting that the complementary strategy achieves the best recall, which even exceeds the union strategy. This is because the best model of each strategy is selected separately to make a fair comparison, and these two strategies may achieve the best F1-score at different epochs and thus select different models. Since different models will produce different sets of predicted pairs, it is possible that complementary strategy achieves a higher recall than union strategy. We further compare the performance of harmonic strategy and complementary strategy, and we find that the precision of harmonic is better than complementary and the recall of complementary is better than harmonic. This shows that the pairs in E-C direction are not completely reliable and setting a threshold to filter pairs in E-C direction is effective.

4.7.3 Effect of Probability Thresholds. **RQ4:** While the harmonic and complementary strategies exhibit to be effective, we further explore the effect of hyper-parameters β and γ of these two strategies on our method.

As shown in Figure 4(a), we can find that the performance fluctuates significantly when β equals to 0.5 or 1. This is because when β equals to 0.5 or 1, the harmonic strategy degenerates to the union strategy or intersection strategy, respectively. We can observe that our model achieves the relatively stable performance with varying β in [0.6,0.9], which indicates the robustness of this strategy. As β increases, the overall performance increases first and then decreases. When β is equal to 0.9, the performance is best (i.e., 0.8013). This indicates that the pairs extracted by only one direction but with high probability are also reliable and should be considered to be added to the final pair set. It should be noted that as β increases, the precision and recall fluctuate. This is because for each threshold value of β, we select its best model according to F1-score, thus the precision and recall are unconstrained and may fluctuate.

As shown in Figure 4(b), we can find that the performance also fluctuates significantly when γ equals to 0.5 or 1. This is because when γ equals to 0.5 or 1, the complementary strategy degenerates to the union strategy or only results of E-C direction, respectively. We can observe that our model achieves the relatively stable performance with varying γ in [0.6,0.9], which indicates the robustness of this strategy. When γ is equal to 0.8, the performance is best (i.e., 0.7945). This indicates that the complementary strategy can improve the performance and introducing pairs in C-E direction to strengthen results in E-C direction will bring improvement. When γ is equal to 1.0, the performance is worst. This shows that combining the results in two directions can improve performance. Finally, we can find when γ increases, the precision and recall do not consistently



(a) Emotion Extraction (b) Cause Extraction (c) Emotion-Cause Pair Extraction

Fig. 5. Effect of α in consistent training strategy.

improve or decline. This is because the threshold values of γ separately select their best models, thus may finally select different models. Since different models will produce different sets of predicted pairs, it is possible that as γ increases, the precision and recall do not consistently improve or decline.

4.7.4 Effect of Consistent Training Strategy. **RQ5:** Considering that the consistent training strategy can bring improvements to our method, we further explore the effect of hyper-parameter α in strategy on emotion extraction, cause extraction, and emotion-cause pair extraction tasks.

When α is equal to 0, the consistent training strategy is not used. First, as shown in Figure 5(c), the performance of our model first has a rising trend with an increase of α and then decreases. When α is equal to 0.1, the performance is best. When α is equal to 0.4, the performance is worst. This shows that setting a suitable consistent training probability can improve performance. When the probability is large, there will be a negative effect. It is worth noting the precision of using consistent training strategy is mostly higher than that of not using it. This may be because after using consistent training strategy, some pseudo queries on the dataset do not have corresponding emotion or cause clauses and the model will more consider the relationship between the query clause and the candidate clauses. Second, as shown in Figure 5(a), we can see that the improvement on emotion extraction is relatively small. This may be because the performance of emotion extraction is relatively high, and it is difficult to improve. Finally, as shown in Figure 5(b), we can see that a suitable α also can improve performance of cause extraction. This shows that these pseudo queries can also improve the performance of cause extraction. It is worth noting that precision and recall fluctuate significantly. This may be attributed to two reasons. First, the process of negative sampling in consistent training strategy is random (i.e., randomly sample a non-emotion clause as a pseudo emotion clause and randomly sample a non-cause clause as a pseudo clause clause), which changes the original training set. Therefore, with the change of α in consistent training strategy, the training set will also change, which may make the model fluctuate. Second, for each value of α, we select the model according to F1-score rather than precision or recall. Therefore, the selected model usually has a high F1-score (the fluctuation of F1 is very small, about 1%), but the precision and recall are not always high and may fluctuate.

4.7.5 Effect of Pair Number. Considering that documents contain multiple emotion-cause pairs seem to be more difficult to process than those contain only one pair, we attempt to further verify the effect of pair number on the performance of our model. Same as Wei et al. [49], we divide the test set into two subsets: One subset contains documents with one emotion-cause pair, and the other subset contains documents with two or more emotion-cause pairs.

As shown in Table 8, we compare our model with **RANKCP**, **Refinement**, and **ISML**. First, it can be seen that our method achieves the best performance on document with one emotion-cause

Table 8. Effect of the Number of Emotion-cause Pairs in Documents

# Pairs	Model	P	R	F1
One per doc.	**RANKCP**†	0.7203	**0.8123**	0.7633
	Refinement†	0.7653	0.7561	0.7607
	ISML†	0.7511	0.7699	0.7678
	CD-MRC†	**0.8041**	0.7882	**0.7961**
Two or more per doc.	**RANKCP**†	0.6772	0.5146	0.5802
	Refinement†	**0.8175**	0.5189	**0.6349**
	ISML†	0.7201	0.5341	0.6104
	CD-MRC†	0.7270	**0.5512**	0.6244

†denotes that the model uses BERT as encoder.

Table 9. Model Size and Performance of Several Strong Baselines and Our Model under Two Experimental Settings

Method	F1	# Param
TRANS†	- /0.6799	2.44M
RANKCP†	0.7360/0.6915	3.70M
Pair-GCN†	- /0.7202	7.11M
Refinement†	0.7463/0.7078	3.22M
ISML†	0.7452/0.7220	0.64M
CD-MRC†	**0.8013/0.7598**	**0.40M**

The best results and the smallest number of parameters are in bold. "-"denotes there is no reported result under this setting.
†denotes that the model uses BERT as encoder.

pair and the second performance on document with two or more emotion-cause pairs. This shows that our method can achieve good performance in different kinds of documents and the main advantage of our method is concentrated on document with one emotion-cause pair. Second, we can see that our method outperforms **Refinement** and **ISML** in all metrics (e.g., P, R, and F1) on the test documents with one pair. This shows that compared with these methods, our method can not only extract more pairs, but also ensure the quality of extracted pairs on the test documents with one pair. Note that the recall of **RANKCP** outperforms our model in documents with one pair. This is because **RANKCP** extracts at least one pair in each document, which makes the recall high and the precision poor. Third, we can see that **Refinement** achieves the best performance in P and F1, and our method achieves the best performance in R and the second performance in P and F1 on the test documents with two or more pairs. This may be because **Refinement** extracts few pairs for documents with two or more pairs. Besides, it is worth noting that the performance of all models on test documents with two or more pairs is worse than those with one pair. This means that processing documents with two or more pairs is the bottleneck of the ECPE task.

4.7.6 Analysis on Model Size. In this part, we further conduct some analysis on the model size of our model and several strong baselines.

As shown in Table 9, for each model, both its performance on the ECPE task under two settings and its model size are listed. Considering that BERT is used as the clause encoder of all strong baselines and our method, the parameters of BERT are not taken into consideration and only parameters outside BERT were counted. From Table 9, we can find that our model not only achieves the best performance but also has the least number of parameters (i.e., 0.40M). The number of our

Table 10. Two Examples for the Case Study of Our Method and the Baseline Methods

Document	Ground-truth	RANKCP	Refinement	ISML	Our Model
18 years ago (c_1), Chengwei Sun killed his uncle (c_2), and then began to flee (c_3). He changed his name and got married (c_4), but he could not escape the law (c_5). He said that for 18 years (c_6), he has been living in fear (c_7).	(c_7, c_2)	(c_7, c_6)	None	None	(c_7, c_2)
25 years ago (c_1), my mother went missing (c_2). In April (c_3), I got news from my friends (c_4), and finally found my mother in Henan Province (c_5). In addition to happiness (c_6), I also worry about my mother's difficulty in settling down (c_7).	(c_6, c_5), (c_7, c_7)	(c_6, c_7), (c_7, c_7)	(c_7, c_7)	(c_6, c_5)	(c_6, c_5), (c_7, c_7)

extra parameters are only equivalent to about 6% of **Pair-GCN** and 63% of **ISML**. This indicates that our model is lightweight yet effective.

4.8 Case Study

RQ6: Since previous experimental results only demonstrate the effectiveness of our method with quantitative analysis, here, we further analyze the superiority of our method over the baseline methods and ablation methods based on some concrete examples in dataset. Specifically, we use four examples in the test set to show the effectiveness of our method. To make a better comparison, we list three strong baseline methods, i.e., **RANKCP**, **Refinement**, and **ISML**, in Table 10, and list three ablation methods, i.e., **w/o Second Turn**, **w/o C-E Direction**, and **w/o Consistent Loss**, in Table 11.

We first compare our method with the baseline methods in Table 10. *For the first example*, this document has a ground-truth emotion-cause pair (c_7, c_2). We can find that when the distance between emotion and cause is far, these baseline methods fail to extract this pair. **RANKCP** predicts wrong pair (c_7, c_6). This is because **RANKCP** predicts at least one pair for each document and prefers the pairs in which cause clause appears before emotion clause. **Refinement** and **ISML** predict None. This may be because it is difficult for sequence labeling method **Refinement** to model the long-term relationship between clauses. When the emotion and its corresponding cause clauses are not in the sliding window (i.e., the distance exceeds the window size), **ISML** cannot extract the correct emotion-cause pair. Different from these methods, **Our Model** can predict the correct pair (c_7, c_2). This may be because our method can utilize the whole document as context for pair extraction and can extract pair even if the distance between the emotion and cause is far. This also explains why our method achieves a higher recall rate. *For the second example*, this document has two emotion-cause pairs (c_6, c_5), (c_7, c_7). We can find that only our model predicts all correct emotion-cause pairs. **RANKCP** predicts correct pair (c_7, c_7) and wrong pair (c_6, c_7). This shows that when a document has multiple pairs, the pair representation is difficult to effectively encode all context information for prediction. **Refinement** and **ISML** only predict one correct pair (i.e., (c_7, c_7) or (c_6, c_5)). This shows that it is difficult for these methods to extract all pairs. Compared to

Table 11. Two Examples for the Case Study of Our Method and the Ablation Methods

Document	Ground-truth	w/o Second Turn	w/o C-E Direction	w/o Consistent Loss	Our Model
It was supposed to be a blind date meeting where young people made their own decisions (c_1), but instead (c_2), parents played the leading role (c_3). Either the child refuses the blind date (c_4), so they are unwilling to come to the date (c_5). Parents are eager to take the initiative to participate (c_6).	(c_6, c_4), (c_6, c_5)	(c_6, c_4)	(c_6, c_5)	(c_6, c_4), (c_6, c_5)	(c_6, c_4), (c_6, c_5)
My parents taught me not to be pleasure-seeking (c_1), but to work hard (c_2). My parents have retired for several (c_3), but they have been working (c_4). I admire them very much (c_5), and want to be like them (c_6).	(c_5, c_4)	None	(c_5, c_4)	(c_5, c_4), (c_5, c_6)	(c_5, c_4)

these methods, our method has a better coverage of potential pairs and can correctly extract pairs in the presence of multiple pairs.

We then compare our full method with the ablation methods in Table 11. *For the first example*, we can find that both **w/o Second Turn** and **w/o C-E Direction** only predict one correct pair while **w/o Consistent Loss** predicts all correct pairs. This indicates that the two-turn framework is effective and combining the results in two directions is effective to extract all potential pairs. *For the second example*, **w/o C-E Direction** and **Our Model** correctly predict the pair, while **w/o Consistent Loss** predicts an extra wrong pair (c_5, c_6). This indicates that the consistent loss is effective and can filter the errors produced by C-E direction. Besides, **w/o Second Turn** does not extract any pair. This suggests that jointly training with the second-turn query can benefit the first-turn query on the ability of identifying emotion and cause.

4.9 Discussion

Since our CD-MRC method is a pipeline system suffering from the error propagation issue, we further explore how it outperforms previous end-to-end method based on empirical analysis. To this end, we conduct experiments by starting with a simple single-direction MRC framework and add our specially designed modules one-by-one. These modules include (1) jointly training the E-C direction and the C-E direction, (2) combining results of two directions, (3) consistent training, and (4) emotion filtering. The experimental results are listed in Table 12.

From Table 12, we can first find that the performance of *only E-C Direction* (i.e., we only use the MRC-style data of E-C direction to train an MRC model) is 0.7335. This performance illustrates the effectiveness of the MRC framework to some extent, but does not exceed that of some strong baselines (i.e., **RANKCP**, **ISML**, **Refinement**). Then, we introduce the MRC-style data in C-E

Table 12. Performance of Adding Modules One-by-one

Model Setting	P	R	F1
Only E-C direction	0.7379	0.7301	0.7335
+ Jointly training with C-E direction	0.7462	0.7583	0.7500
+ Combining results of C-E direction	0.7532	0.7532	0.7512
+ Consistent Training	0.7593	0.7540	0.7588
+ Emotion Filering	**0.8249**	**0.7800**	**0.8013**

direction to jointly train the model, the F1-score of which increases to 0.7500. We can find that this performance is better than that of the best baselines method in Table 2. This indicates that optimizing the four subtasks (i.e., emotion extraction, cause extraction, emotion-specific cause extraction, and cause-specific emotion extraction) jointly can enable them to benefit from each other. After that, we further add the combination strategy, consistent training strategy, and emotion filtering strategy and find that the performance of our model can be further improved. This implies that all these strategies are effective and can help the model to further reduce the error. Overall, the reasons that our CD-MRC method outperforms previous end-to-end methods can be summarized as: (1) jointly optimizing four subtasks, (2) combining results of both directions, (3) using consistent loss to make the training and inference phases more consistent, and (4) filtering easily detected invalid emotion clauses.

5 CONCLUSION

In this article, we reformalize the ECPE task as a two-turn MRC task and propose a consistent dual-MRC framework to solve the task. In the consistent dual-MRC framework, the emotion-cause pairs are extracted in two directions, i.e., E-C direction and C-E direction. In the E-C direction, we first identify all emotion clauses based on the emotion extraction query and then identify the corresponding cause clauses for each identified emotion clause based on the emotion-specific cause extraction query. The C-E direction is similar but conducted in opposite direction. For candidate pairs extracted from both directions, we explore four strategies to combine them into the final set of pairs. Furthermore, we propose a consistent training strategy for model training, which enables the model to filter the errors produced by the first turn at inference. The experimental results demonstrate that: (1) our method outperforms the existing methods and achieves state-of-the-art performance; (2) combining the results from two directions can further improve performance; (3) our proposed consistent training strategy can alleviate *exposure bias*; (4) we found that natural language queries contain more semantic information and can achieve better results.

REFERENCES

[1] Faizan Ahmad, Ahmed Abbasi, Jingjing Li, David G. Dobolyi, Richard G. Netemeyer, Gari D. Clifford, and Hsinchun Chen. 2020. A deep learning architecture for psychometric natural language processing. *ACM Trans. Inf. Syst.* 38, 1 (2020), 6:1–6:29.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations*.

[3] Danqi Chen. 2018. *Neural Reading Comprehension and Beyond*. Ph.D. Dissertation. Stanford University.

[4] Shaowei Chen, Yu Wang, Jie Liu, and Yuelin Wang. 2021. Bidirectional machine reading comprehension for aspect sentiment triplet extraction. In *35th AAAI Conference on Artificial Intelligence*. 12666–12674.

[5] Xinhong Chen, Qing Li, and Jianping Wang. 2020. Conditional causal relationships between emotions and causes in texts. In *Conference on Empirical Methods in Natural Language Processing*. 3111–3121.

[6] Xinhong Chen, Qing Li, and Jianping Wang. 2020. A unified sequence labeling model for emotion cause pair extraction. In *28th International Conference on Computational Linguistics, COLING 2020*. 208–218.

[7] Ying Chen, Wenjun Hou, Xiyao Cheng, and Shoushan Li. 2018. Joint learning for emotion classification and emotion cause detection. In *Conference on Empirical Methods in Natural Language Processing*. 646–651.

[8] Ying Chen, Wenjun Hou, Shoushan Li, Caicong Wu, and Xiaoqiang Zhang. 2020. End-to-end emotion-cause pair extraction with graph convolutional network. In *28th International Conference on Computational Linguistics*. 198–207.
[9] Ying Chen, Sophia Yat Mei Lee, Shoushan Li, and Chu-Ren Huang. 2010. Emotion cause detection with linguistic constructions. In *23rd International Conference on Computational Linguistics*. 179–187.
[10] 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 Trans. Audio Speech Lang. Process.* 29 (2021), 2779–2791.
[11] Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Hua Yu, and Qing Gu. 2020. A symmetric local search network for emotion-cause pair extraction. In *28th International Conference on Computational Linguistics*. 139–149.
[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 4171–4186.
[13] Zixiang Ding, Huihui He, Mengran Zhang, and Rui Xia. 2019. From independent prediction to reordered prediction: Integrating relative position and global label information to emotion cause identification. In *33rd AAAI Conference on Artificial Intelligence*. 6343–6350.
[14] Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction. In *58th Annual Meeting of the Association for Computational Linguistics*. 3161–3170.
[15] Zixiang Ding, Rui Xia, and Jianfei Yu. 2020. End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Conference on Empirical Methods in Natural Language Processing*. 3574–3583.
[16] Xinya Du and Claire Cardie. 2020. Event extraction by answering (almost) natural questions. In *Conference on Empirical Methods in Natural Language Processing*. 671–683.
[17] Chuang Fan, Hongyu Yan, Jiachen Du, Lin Gui, Lidong Bing, Min Yang, Ruifeng Xu, and Ruibin Mao. 2019. A knowledge regularized hierarchical approach for emotion cause analysis. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 5613–5623.
[18] Chuang Fan, Chaofa Yuan, Jiachen Du, Lin Gui, Min Yang, and Ruifeng Xu. 2020. Transition-based directed graph construction for emotion-cause pair extraction. In *58th Annual Meeting of the Association for Computational Linguistics*. 3707–3717.
[19] Chuang Fan, Chaofa Yuan, Lin Gui, Yue Zhang, and Ruifeng Xu. 2021. Multi-task sequence tagging for emotion-cause pair extraction via tag distribution refinement. *IEEE ACM Trans. Audio Speech Lang. Process.* 29 (2021), 2339–2350.
[20] Kai Gao, Hua Xu, and Jiushuo Wang. 2015. Emotion cause detection for chinese micro-blogs based on ECOCC model. In *19th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining*. 3–14.
[21] Qinghong Gao, Jiannan Hu, Ruifeng Xu, Gui Lin, Yulan He, Qin Lu, and Kam-Fai Wong. 2017. Overview of NTCIR-13 ECA task. In *NTCIR-13 Conference*.
[22] Diman Ghazi, Diana Inkpen, and Stan Szpakowicz. 2015. Detecting emotion stimuli in emotion-bearing sentences. In *16th International Conference on Computational Linguistics and Intelligent Text Processing*. 152–165.
[23] Lin Gui, Jiannan Hu, Yulan He, Ruifeng Xu, Qin Lu, and Jiachen Du. 2017. A question answering approach for emotion cause extraction. In *Conference on Empirical Methods in Natural Language Processing*. 1593–1602.
[24] Lin Gui, Dongyin Wu, Ruifeng Xu, Qin Lu, and Yu Zhou. 2016. Event-driven emotion cause extraction with corpus construction. In *Conference on Empirical Methods in Natural Language Processing*. 1639–1649.
[25] Lin Gui, Li Yuan, Ruifeng Xu, Bin Liu, Qin Lu, and Yu Zhou. 2014. Emotion cause detection with linguistic construction in Chinese Weibo text. In *3rd CCF Conference on Natural Language Processing and Chinese Computing*. 457–464.
[26] Sophia Yat Mei Lee, Ying Chen, and Chu-Ren Huang. 2010. A text-driven rule-based system for emotion cause detection. In *NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*. 45–53.
[27] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. 2017. Zero-shot relation extraction via reading comprehension. In *21st Conference on Computational Natural Language Learning (CoNLL'17)*. 333–342.
[28] Fayuan Li, Weihua Peng, Yuguang Chen, Quan Wang, Lu Pan, Yajuan Lyu, and Yong Zhu. 2020. Event extraction as multi-turn question answering. In *Findings of the Association for Computational Linguistics:'20*. 829–838.
[29] Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *58th Annual Meeting of the Association for Computational Linguistics*. 5849–5859.
[30] Xiangju Li, Wei Gao, Shi Feng, Daling Wang, and Shafiq R. Joty. 2021. Span-level emotion cause analysis by BERT-based graph attention network. In *30th ACM International Conference on Information and Knowledge Management*. 3221–3226.
[31] Xiangju Li, Wei Gao, Shi Feng, Daling Wang, and Shafiq R. Joty. 2021. Span-level emotion cause analysis with neural sequence tagging. In *30th ACM International Conference on Information and Knowledge Management*. 3227–3231.
[32] Xiangju Li, Wei Gao, Shi Feng, Yifei Zhang, and Daling Wang. 2021. Boundary detection with BERT for span-level emotion cause analysis. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP'21*. 676–682.

[33] Xiangju Li, Kaisong Song, Shi Feng, Daling Wang, and Yifei Zhang. 2018. A co-attention neural network model for emotion cause analysis with emotional context awareness. In *Conference on Empirical Methods in Natural Language Processing*. 4752–4757.

[34] Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019. Entity-relation extraction as multi-turn question answering. In *57th Conference of the Association for Computational Linguistics*. 1340–1350.

[35] Yanran Li, Ke Li, Hongke Ning, Xiaoqiang Xia, Yalong Guo, Chen Wei, Jianwei Cui, and Bin Wang. 2021. Towards an online empathetic chatbot with emotion causes. In *44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2041–2045.

[36] Jian Liu, Yubo Chen, Kang Liu, Wei Bi, and Xiaojiang Liu. 2020. Event extraction as machine reading comprehension. In *Conference on Empirical Methods in Natural Language Processing*. 1641–1651.

[37] Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. In *Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*. 3728–3738.

[38] Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *7th International Conference on Learning Representations*.

[39] Bing Ma, Cao Liu, Jingyu Wang, Shujie Hu, Fan Yang, Xunliang Cai, Guanglu Wan, Jiansong Chen, and Jianxin Liao. 2021. Distant supervision based machine reading comprehension for extractive summarization in customer service. In *44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1895–1899.

[40] Yue Mao, Yi Shen, Chao Yu, and Longjun Cai. 2021. A joint training dual-MRC framework for aspect based sentiment analysis. In *35th AAAI Conference on Artificial Intelligence*. 13543–13551.

[41] Bryan McCann, Nitish Shirish Keskar, Caiming Xiong, and Richard Socher. 2018. The natural language decathlon: Multitask learning as question answering. *CoRR* abs/1806.08730 (2018).

[42] Soujanya Poria, Navonil Majumder, Devamanyu Hazarika, Deepanway Ghosal, Rishabh Bhardwaj, Samson Yu Bai Jian, Romila Ghosh, Niyati Chhaya, Alexander F. Gelbukh, and Rada Mihalcea. 2020. Recognizing emotion cause in conversations. *CoRR* abs/2012.11820 (2020).

[43] Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *4th International Conference on Learning Representations*.

[44] Aaditya Singh, Shreeshail Hingane, Saim Wani, and Ashutosh Modi. 2021. An end-to-end network for emotion-cause pair extraction. In *11th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. 84–91.

[45] Haolin Song, Chen Zhang, Qiuchi Li, and Dawei Song. 2020. End-to-end emotion-cause pair extraction via learning to link. *CoRR* abs/2002.10710 (2020).

[46] Elsbeth Turcan, Shuai Wang, Rishita Anubhai, Kasturi Bhattacharjee, Yaser Al-Onaizan, and Smaranda Muresan. 2021. Multi-task learning and adapted knowledge models for emotion-cause extraction. In *Findings of the Association for Computational Linguistics: ACL/IJCNLP'21*. 3975–3989.

[47] Fanfan Wang, Zixiang Ding, Rui Xia, Zhaoyu Li, and Jianfei Yu. 2021. Multimodal emotion-cause pair extraction in conversations. *CoRR* abs/2110.08020 (2021).

[48] Shih-Ming Wang and Lun-Wei Ku. 2016. ANTUSD: A large Chinese sentiment dictionary. In *10th International Conference on Language Resources and Evaluation*.

[49] Penghui Wei, Jiahao Zhao, and Wenji Mao. 2020. Effective inter-clause modeling for end-to-end emotion-cause pair extraction. In *58th Annual Meeting of the Association for Computational Linguistics*. 3171–3181.

[50] Sixing Wu, Fang Chen, Fangzhao Wu, Yongfeng Huang, and Xing Li. 2020. A multi-task learning neural network for emotion-cause pair extraction. In *24th European Conference on Artificial Intelligence*. 2212–2219.

[51] Rui Xia and Zixiang Ding. 2019. Emotion-cause pair extraction: A new task to emotion analysis in texts. In *57th Conference of the Association for Computational Linguistics*. 1003–1012.

[52] Rui Xia, Mengran Zhang, and Zixiang Ding. 2019. RTHN: A RNN-transformer hierarchical network for emotion cause extraction. In *28th International Joint Conference on Artificial Intelligence*. 5285–5291.

[53] Hanqi Yan, Lin Gui, Gabriele Pergola, and Yulan He. 2021. Position bias mitigation: A knowledge-aware graph model for emotion cause extraction. In *59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*. 3364–3375.

[54] Chaofa Yuan, Chuang Fan, Jianzhu Bao, and Ruifeng Xu. 2020. Emotion-cause pair extraction as sequence labeling based on a novel tagging scheme. In *Conference on Empirical Methods in Natural Language Processing*. 3568–3573.

[55] Lei Zhang and Bing Liu. 2017. Sentiment analysis and opinion mining. In *Encyclopedia of Machine Learning and Data Mining*. Springer, 1152–1161.

[56] Guangyou Zhou and Jimmy Xiangji Huang. 2017. Modeling and mining domain shared knowledge for sentiment analysis. *ACM Trans. Inf. Syst.* 36, 2 (2017), 18:1–18:36.

Received 19 December 2021; revised 28 June 2022; accepted 1 August 2022