A Unified Target-Oriented Sequence-to-Sequence Model for Emotion-Cause Pair Extraction

Zifeng Cheng, Zhiwei Jiang, Yafeng Yin, Na Li, and Qing Gu

***Abstract*—Emotion-cause pair extraction is a recently proposed task that aims at extracting all potential clause-level pairs of emotion and cause in text. To solve this task, researchers first proposed a two-step pipeline method. This method extracts the emotions and causes individually in the first step, then pairs the extracted emotions and causes and filters the invalid emotion-cause pairs in the second step. Due to that the two-step method has the error accumulation problem and is hard to be optimized jointly, several one-step end-to-end models have been proposed. These models share a similar underlying idea, that is, reframing the emotion-cause pair extraction task as a classification problem of candidate clause pairs. Unlike these models, in this paper, we reframe the emotion-cause pair extraction task as a unified sequence labeling problem, which allows to extract emotion-cause pairs through one pass of sequence labeling. This is realized by designing a special set of unified labels. In the unified label, we design a content part for emotion/cause identification and a pairing part for clause pairing. Then the emotion-cause pairs can be implicitly derived from the unified labels. To address this unified sequence labeling problem, we propose a unified target-oriented sequence-to-sequence model, which comprehensively utilizes the information of target clause, global context, and former decoded label, to perform end-to-end unified sequence labeling. The experimental results demonstrate the effectiveness of both our proposed unified sequence labeling scheme and unified target-oriented sequence-to-sequence model. All the code and data of this work can be obtained at https://github.com/zifengcheng/UTOS.**

***Index Terms*—Emotion-cause pair extraction, sequence-to-sequence learning, sequence labeling.**

I. Introduction

EMOTION-CAUSE pair extraction (ECPE) [1] is a new task recently proposed in the field of sentiment analysis and has received extensive attention [2]–[7]. The objective of ECPE is to simultaneously identify the emotions and their corresponding causes in text, that is, to extract all potential pairs of emotion and cause in text. It is commercially valuable to know what emotions are expressed in text and understand why these emotions occur for applications such as product reviews mining and user feedback analysis.

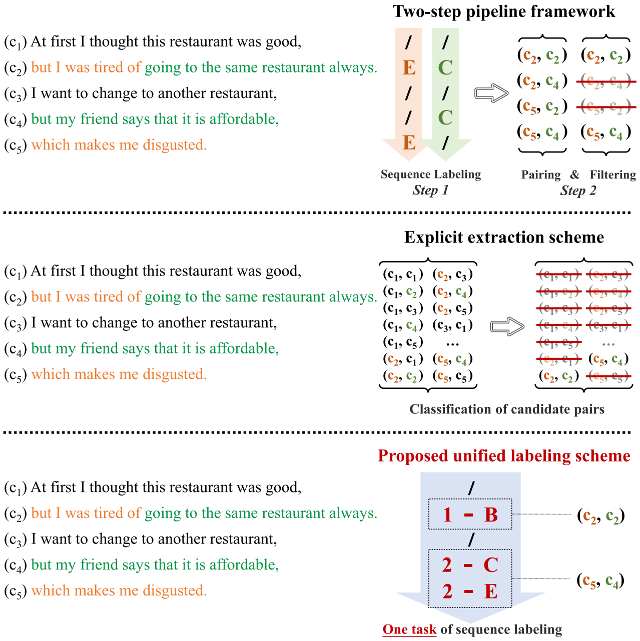

Two-step pipeline framework
(c_1) At first I thought this restaurant was good,
(c_2) but I was tired of going to the same restaurant always.
(c_3) I want to change to another restaurant,
(c_4) but my friend says that it is affordable,
(c_5) which makes me disgusted.
/ E / / E
/ C / C /
(c_2, c_2) (c_2, c_4) (c_5, c_2) (c_5, c_4)
(c_2, c_2) ~~(c_2, c_4)~~ ~~(c_5, c_2)~~ (c_5, c_4)
Sequence Labeling
Step 1
Pairing & Filtering
Step 2
Explicit extraction scheme
(c_1, c_1) (c_1, c_2) (c_1, c_3) (c_1, c_4) (c_1, c_5) (c_2, c_1) (c_2, c_2)
(c_2, c_3) (c_2, c_4) (c_2, c_5) (c_3, c_1) … (c_5, c_4) (c_5, c_5)
Classification of candidate pairs
Proposed unified labeling scheme
/
1 - B
/
2 - C
2 - E
(c_2, c_2)
(c_5, c_4)
One task of sequence labeling

Fig. 1. An example of three schemes for the ECPE task. Clause c_2 itself can form an emotion-cause pair (c_2, c_2). Emotion clause c_5 and its corresponding cause clause c_4 can form another emotion-cause pair (c_5, c_4).

The ECPE task is originated from the previous emotion cause extraction (ECE) task [8], [9], which aims to identify the cause of a given emotion (i.e., the emotion should be annotated) in text. Considering that the emotions are not naturally annotated in text, Ding and Xia [1] proposed the ECPE task, which needs to identify both emotion and cause, as well as the causal relationship between them. As shown in Figure 1, given an unannotated document as the input, ECPE aims to extract a set of valid emotion-cause pairs at clause level: (c_2, c_2), (c_5, c_4).

To address the ECPE task, many methods have been proposed. Ding and Xia [1] first proposed a two-step pipeline framework to address the task. As shown in the top part of Figure 1, in the first step, this pipeline framework utilizes two sub-tasks to extract the emotions and causes individually. In the second step, it pairs the extracted emotions and causes, and then filters out the invalid emotion-cause pairs. The two-step pipeline method has shown its effectiveness, but it has two shortcomings: the errors

Manuscript received November 27, 2020; revised April 23, 2021 and July 16, 2021; accepted July 19, 2021. Date of publication August 11, 2021; date of current version August 31, 2021. This work was supported in part by the National Science Foundation of China under Grants 61906085, 61802169, 61972192, and 41972111, by JiangSu Natural Science Foundation under Grant BK20180325, by the Second Tibetan Plateau Scientific Expedition and Research Program under Grant 2019QZKK0204, and by the Collaborative Innovation Center of Novel Software Technology and Industrialization. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianfeng Gao. *(Corresponding author: Zhiwei Jiang.)*

The authors are with the State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China (e-mail: chengzf@smail.nju.edu.cn; jzw@nju.edu.cn; yafeng@nju.edu.cn; li.na@smail.nju.edu.cn; guq@nju.edu.cn).

Digital Object Identifier 10.1109/TASLP.2021.3102194

from the first step will affect the performance of the second step, and it is hard to optimize the overall performance of two steps jointly.

Considering the above problems, researchers therefore seek to perform the ECPE task in one step and have proposed several effective end-to-end neural network models [2]–[6]. While these models have different model structures, they share a similar underlying idea, that is, reframing the ECPE task as a classification problem of candidate emotion-cause pairs. From the perspective of candidate pair generation, these models can be categorized into two types. One type of them enumerates the clause pair candidates by Cartesian product (i.e., each two clauses can form a pair), and constructs the representation of each clause pair for classification [2], [4]–[6]. The other type of them incrementally decodes the clause pair candidates along a process of transition-based parsing and uses a neural transition-based model for classification [3]. Both types of these models can be viewed as a kind of explicit extraction scheme, as shown in the middle part of Fig. 1, where the emotion-cause pairs are extracted by explicitly applying classification on potential clause pairs.

Unlike viewing the ECPE task as a clause pair classification problem, in this paper, we reframe the ECPE task as a unified sequence labeling problem, which allows to extract emotion-cause pairs through one pass of sequence labeling. This is mainly realized by designing a special set of unified labels, each of which consists of the content part and the pairing part. The content part indicates whether the clause contains emotion or cause, while the pairing part indicates which clauses should be paired. Taking the clause sequence in the bottom part of Fig. 1 as an example, for clause c_2, the content part B indicates that c_2 contains both the emotion and the corresponding cause, while the pairing part 1 indicates that other clauses with the same pairing part 1 should be paired with this clause (in this example, there is no other label with pairing part 1). Similarly, for clause c_4 and c_5, their labels indicate that c_4 only contains the cause, c_5 only contains the emotion, and these two clauses should be paired since their labels are of the same pairing part 2. Thus, two emotion-cause pairs (i.e., (c_2, c_2) and (c_5, c_4)) can be implicitly derived from the predicted unified labels.

Under our proposed unified labeling scheme, the content part is labeled with a method similar to the first step of the two-step pipeline framework and can be predicted based on the contextualized representation of target clause. The difficulty mainly lies in that the labeling of pairing part depends not only on the contextualized representation of target clause, but also on the labels of other non-target clauses (e.g., the pairing part of a cause clause should be the same as that of its corresponding emotion clause, and different from that of other emotion clauses). To address this challenge, we introduce the sequence-to-sequence learning framework, which can model the dependencies among pairing parts through a way of predicting the next label conditioned on the former predicted labels.

By adapting the sequence-to-sequence model to the unified labeling scheme, we propose a Unified Target-Oriented Sequence-to-sequence (UTOS) model for the ECPE task. Different from general sequence-to-sequence model that the input and output sequence can be of different lengths, our UTOS model constrains the input and output sequence to have the same length and guarantees the i-th decoded label is corresponding to the i-th clause in the input sequence. Specifically, the UTOS model is designed based on the encoder-decoder framework. The encoder accepts the document as input and performs hierarchical encoding to generate the representation of sequence and each clause. The decoder takes the sequence representation and clause representations as inputs and decodes the unified label of each clause. In particular, to ensure the one-to-one mapping between decoded labels and input clauses, and make full use of the information about current clause, previous decoded labels, and global context, we design two components in the decoder: target-oriented sequence decoder and unified labeling. The target-oriented sequence decoder can decode a state for each clause by taking the global sequence representation, the representation of target clause, and the predicted label of the former clause as input. After obtaining the decoded state of clause, the multi-class classification is performed to fulfill the unified labeling. Finally, the results of unified labeling can be straightly mapped to the emotion-cause pairs.

The contributions of this paper are as follows:

- We reframe the ECPE task as a unified sequence labeling problem by designing a unified labeling scheme, which allows to predict where emotions and causes are and how they pair through one pass of sequence labeling.
- We propose a unified target-oriented sequence-to-sequence model to address the unified sequence labeling problem. The model can comprehensively utilize the information of target clause, global context, and former decoded label, to perform end-to-end unified sequence labeling.
- The experimental results demonstrate the effectiveness of both the proposed unified sequence labeling scheme and UTOS model.

The rest of this paper is organized as follows. Section II introduces the related work on emotion-cause pair extraction and sequence-to-sequence learning. Section III gives the definition of the ECPE task from the perspective of unified sequence labeling and describes the details of the proposed UTOS model. Section IV reports the evaluation results of the proposed method against three groups of baseline methods and conducts comprehensive experiments for analysis. Conclusions are finally drawn in Section V.

II. Related Work

A. Emotion-Cause Pair Extraction

The emotion-cause pair extraction (ECPE) task is a new task originated from the emotion cause extraction (ECE) task. Lee *source* [8] first proposed the ECE task and formulated it as a word-level sequence labeling problem. But Chen *source* [10] suggested that the ECE task may be more suitable to be addressed at the clause level than word level. Afterwards, Gui *source* [9] released a Chinese ECE corpus which defined the task as a clause-level sequence labeling problem and became a benchmark corpus for latter studies on the ECE task. While early studies mainly adopted the rule-based methods [10]–[12]

and traditional machine learning methods [13] to deal with the ECE task, recent studies has begun to apply the deep learning methods to solve this task [14]–[20]. More recently, considering that the emotions are often not given in practice, Xia and Ding [1] proposed the ECPE task.

To address the ECPE task, Xia and Ding [1] proposed a two-step pipeline method. This method first extracts the emotion and cause individually, then uses Cartesian product and logistic regression to pair the extracted emotions and causes and filter out invalid pairs. Due to that the two-step method has the error accumulation problem and is hard to be optimized jointly, several one-step end-to-end models have been proposed [2]–[6]. These models reframe the ECPE task as a clause pair classification problem and extract emotion-cause pairs by explicitly performing classification on candidate clause pairs. Considering the difference in candidate pair generation, one type of these models enumerate the candidate clause pairs by Cartesian product [2], [4]–[6], while the other type incrementally decodes the candidate clause pairs along a process of transition-based parsing [3]. Among the models of the former type, Ding *source* [2] represented the emotion-cause pairs by a 2D representation and designed a 2D transformer module for pair extraction. Wei *source* [6] learned the clause pair representations with graph attention and kernel-based relative position embedding, and extracted the emotion-cause pairs from a ranking perspective. Wu *source* [6] and Song *source* [5] represented the clause pair representation based on the shared features across three tasks which include ECPE, emotion extraction, and cause extraction, and proposed a multi-task neural network for pair extraction. For the latter type, Fan *source* [3] transformed the ECPE problem into a procedure of transition-based directed graph construction and proposed a neural transition-based model for pair extraction.

Recently, the methods based on unified sequence labeling have been demonstrated to be effective in joint extraction tasks, such as joint extraction of entities and relations [21]. For the ECPE task, in the same period of our work, researchers have also proposed some sequence labeling methods based on other tagging schemes to jointly extract the emotions and causes [22], [23]. Yuan *source* [22] designed a novel tagging scheme, in which each clause is labeled by two kinds of labels. The first one is used to indicate whether the clause is a cause clause and the second one further indicates the distance between the cause and the corresponding triggered emotion. Chen *source* [23] also designed a set of unified labels to pair up emotions and causes, where each unified label contains a causal identity part to indicate the type of clauses and an emotion type part to pair the clauses. Among these two methods and our method, the difference mainly lies on the designment of the paring part. Unlike these two methods, we use an incremental number as pairing part for pairing.

B. Sequence-to-Sequence Learning

Sequence-to-sequence learning is first proposed to provide a way of applying deep neural networks to solve general sequence-to-sequence problems in machine translation [24], [25]. These models usually encode a source sentence into a fixed-length vector from which a decoder is then used to generate a translation sentence. Bahdanau *source* [26] conjectured that the use of a fixed-length vector is a bottleneck in improving the performance of this basic encoder-decoder architecture. To solve the problem, they introduced the attention mechanism which can automatically find the part of a source sentence that is relevant to predicting a target word. Then Luong *source* [27] examined two simple and effective classes of attention: a global approach which always attends to all source words and a local one that only looks at a subset of source words at a time.

In recent years, due to the ability of generating sequence, sequence-to-sequence learning has been widely applied in other sequence generation tasks, such as abstractive summarization [28], voice conversion [29], and emphasis speech translation [30]. Besides, considering that the sequence-to-sequence model can capture the complex global dependencies among output labels and the overall meaning of the input sequence, it has also been applied to some tasks such as multi-label classification [31]–[33], dialogue act prediction [34], dependency parsing [35], [36], and aspect term extraction [37].

Motivated by these properties, we propose the UTOS model based on sequence-to-sequence learning. Unlike the classic sequence-to-sequence model that generates sequence of unrestricted length, UTOS decodes the sequence in a target-oriented way, so as to ensure a one-to-one correspondence between the output sequence and the input sequence.

C. Neural Sequence Labling

Sequence labeling is an important task in the field of NLP and its typical applications include named entity recognition [38], [39], part-of-speech tagging [40], aspect term extraction [37], [41], and so on. In recent years, with the development of deep learning technology, the neural sequence labeling model has shown its superiority and gradually become the mainstream method for the sequence labeling task.

Neural sequence labeling model generally consists of two main components: feature extractor and decoder. Feature extractor is used to extract the features for each unit of sequence (e.g., word is the unit of sentence). The commonly used feature extractors include CNN [42], BiLSTM [38], [43]–[45], and Transformer [46]. Decoder is used to assign a label to each unit of sequence. In general, a decoder that can capture dependencies among labels could decode better label sequence. Since CRF (Conditional Random Field) can use transition matrix to model the local label dependency [42], previous studies usually use CRF as the decoder [38], [42], [43], [45], [46]. The classic neural sequence labeling models include BiLSTM-Softmax [44], BiLSTM-CRF [45], BERT-Softmax [47], and so on.

Recently, some other methods have also appeared. Cui and Zhang [48] pointed that BiLSTM-CRF does not always lead to better results compared with BiLSTM-Softmax, and proposed a label attention network for sequence labeling. Considering that the sequence-to-sequence learning has a good ability to learn complex global dependencies, researchers have applied it to solve the sequence labeling tasks such as chunking [49], aspect term extraction [37], and dialogue act prediction problem [34].

Besides, Liu *source* [50] first introduced the deep transition architecture to sequence labeling task and further enhance it with the global contextual representation.

III. Method

In this section, we first present the task definition of ECPE from the perspective of unified labeling. Then, we introduce the overview and the technical details of the proposed Unified Target-Oriented Sequence-to-sequence model (UTOS).

A. Task Definition

We formulate the emotion-cause pair extraction task as a unified sequence labeling problem. Given a sequence of clauses (i.e., a document) $X = \{c_1, c_2, \ldots, c_T\}$ with length T as input, the goal of ECPE can be reframed as predicting a sequence of unified labels $Y = \{y_1, y_2, \ldots, y_T\}$, where $y_i \in \mathcal{Y}$ is the label of the clause c_i and $\mathcal{Y}$ is a set of unified labels.

To establish a mapping between the predicted sequence labels and the emotion-cause pairs, we design a special set of unified labels $\mathcal{Y} = \{$1-E, 1-C, 1-B, 2-E, 2-C, 2-B, $\cdots$, k-E, k-C, k-B$\} \bigcup \{$N$\}$. Except N which indicates neither emotion nor cause is identified in the clause, each unified label consists of two parts: the content part and the pairing part. The content part can be labeled as E, C, or B to indicate which type of content (i.e., emotion, cause, or both of them) is identified in the clause. The pairing part can be labeled as $i \in \{1, 2, \ldots, k\}$ and indicates how to pair clauses. Generally, an emotion clause (labeled as E or B) and a cause clause (labeled as C or B) with the same pairing part can constitute an emotion-cause pair.

To ensure the ground-truth emotion-cause pairs can be mapped to a deterministic sequence of unified labels, we define a deterministic allocation scheme for the pairing part of unified label. Specifically, given a sequence of clauses, the pairing parts of clauses are assigned one-by-one from the first clause to the final clause, and the pairing index is allocated in ascending order from 1 to k. In this way, if the i-th clause can be paired with the previous j-th clause (i.e., j $<$ i) to form an emotion-cause pair, then the i-th clause is assigned the same pairing part as the j-th clause; otherwise, the the i-th clause is assigned a new pairing index. For example, as shown in the bottom part of Fig. 1, c_2 is a clause carrying both emotion and cause, and there is no previous clause can be paired with it, then c_2 receives the pairing index 1 and is labeled as 1-B. The cause clause c_4 also can not be paired with its previous clause, thus receives a new pairing index 2 and is labeled as 2-C. The emotion clause c_5 can be paired with clause c_4, thus receives c_4's pairing index 2 and is labeled as 2-E.

It is worth noting that an emotion may correspond to multiple causes and a cause may correspond to multiple emotions. For these particular cases, these clauses (i.e., the emotion clause and its multiple cause clauses or the cause clause and its multiple emotion clauses) should be treated as a pairing group, and the clauses in the same pairing group should be assigned the same pairing part to avoid conflicts. In addition, the maximum value k of pairing part is determined by the dataset to ensure the coverage of all ground-truth unified labels.

B. An Overview of UTOS

To address the ECPE task under the unified labeling scheme, we propose a Unified Target-Oriented Sequence-to-sequence model (UTOS). As shown in Fig. 2, UTOS receives a sequence of clauses as input and predicts the unified label of each clause one by one, all of which can be finally converted into a set of emotion-cause pairs.

The proposed UTOS model is based on sequence-to-sequence learning but decodes the sequence of unified labels in a target-oriented way. Specifically, UTOS consists of two main components: the hierarchical sequence encoder (HSE) and the target-oriented sequence decoder (TOSD). As shown in Figure 2, HSE refers to the bottom four layers, and is used to encode the input clause sequence in a hierarchical way and output the representation of sequence and target clause. TOSD refers to the upper two layers, and is used to decode the unified labels of sequence from the sequence representation one-by-one meanwhile taking the information of target clause and the former decoded labels into consideration.

To effectively assign the content part and pairing part under the unified labeling scheme, UTOS is designed to integrate the characteristics of both sequence labeling model and sequence-to-sequence model. General sequence labeling model usually predicts the label of each target clause only based on its contextualized representation, while general sequence-to-sequence model usually decodes the output sequence from the input sequence representation and the length of output sequence is unrestricted. By integrating both of them, UTOS encodes the information of input sequence into both the sequence representation and the target clause representation, and decodes unified labels from both representations meanwhile restricts that the decoded unified labels should correspond to the input clauses one to one. Under our unified labeling scheme, this target-oriented sequence-to-sequence way have two advantages over the general sequence labeling model and sequence-to-sequence model. First, the content part of input clauses can be better inferred when both the sequence representation and the target clause representation are used for decoding. Second, the pairing part of output unified labels can be better assigned when the former decoded unified labels are given for latter decoding.

C. Model Description

In this section, we present the detailed description of the hierarchical sequence encoder and target-oriented sequence decoder.

1) Hierarchical Sequence Encoder: HSE encodes the input clause sequence in a hierarchical way with four levels of encoders: word embedding, clause encoder, sequence encoder, and target encoder.

Word Embedding The input of our model is a sequence of clauses $X = \{c_1, c_2, \ldots, c_T\}$, where each clause $c_i \in X$ is a list of words $c_i = \{w_1^i, w_2^i, \ldots, w_{l_i}^i\}$ with the length l_i. For each clause c_i, the word embedding layer maps the words into their word embeddings $e_i = \{e_1^i, e_2^i, \ldots, e_{l_i}^i\}$.

Clause Encoder After word embedding, a bi-directional Long Short-Term Memory (LSTM) layer [51] takes the word

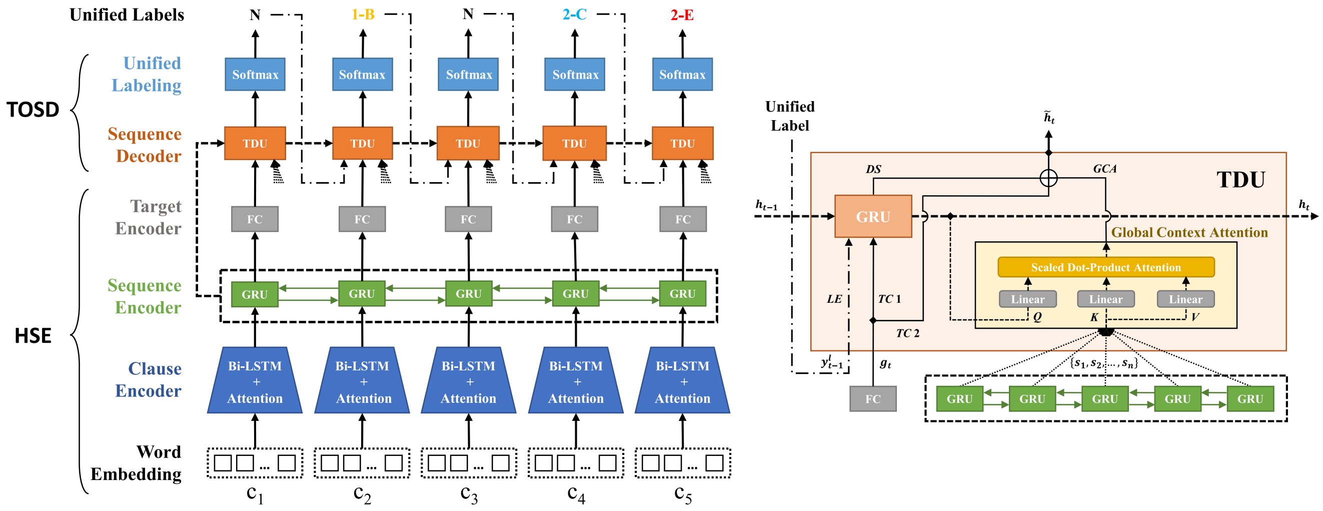


Fig. 2. The framework of the proposed UTOS model. The left part is the framework of the proposed UTOS model and the right part is the details of TDU.

embeddings e_i of clause c_i as input and outputs the context-aware representation $r_i = \{r^i_1, r^i_2, \ldots, r^i_{l_i}\}$. Then an attention layer is used to get the clause representation v_i for clause c_i:

$$v_i = \sum_j a^i_j r^i_j \tag{1}$$

$$a^i_j = \frac{exp((u^i_j)^T u_w)}{\sum_p exp((u^i_p)^T u_w)} \tag{2}$$

$$u^i_j = tanh(W_1 r^i_j + b_1) \tag{3}$$

where a^i_j is the attention weight of word representation r^i_j, and W_1, b_1 and u_w are weight matrix, bias vector and context vector respectively. Note that we also use BERT [47] as an alternative clause encoder to produce clause representation.

Sequence Encoder After clause encoding, a bi-directional Gated Recurrent Unit (GRU) layer takes a sequence of clause representations $\{v_1, v_2, \ldots, v_T\}$ as input and outputs their contextualized clause representation $\{s_1, s_2, \ldots, s_T\}$ where $s_i = [\overrightarrow{s_i}, \overleftarrow{s_i}]$. The sequence representation is the final state of the forward direction $h_0 = \overrightarrow{s_T}$.

Target Encoder For the target clause c_i, a fully connected layer takes its contextualized clause representation s_i as input and outputs its target clause representation:

$$g_i = ReLU(W_2 s_i + b_2) \tag{4}$$

where W_2 and b_2 are the weight matrix and bias vector respectively.

2) Target-Oriented Sequence Decoder: TOSD takes the sequence representation along with the contextualized clause representation and target clause representation as inputs and decodes the labels of sequence in a target-oriented way. For the label decoding of a specific target clause, the representation of the target clause, together with the former decoded hidden state and predicted label, are used as the input of the decoder.

Sequence Decoder The sequence decoder employs a Target Decoder Unit (TDU) to get the final representation of each clause. As shown in the right part of Figure 2, at the t-th step of sequence decoding, TDU takes four kinds of vectors as inputs, i.e., the target clause representation g_t, the hidden state h_{t-1} decoded by former TDU, the label embedding y^l_{t-1} of predicted label for former clause, and the contextualized clause representation of each clause $\{s_1, s_2, \ldots, s_n\}$, and outputs a final representation of target clause for latter unified labeling. Specifically, TDU mainly consists of two components: a GRU and a Global Context Attention (GCA). It is worth noting that the GRU used here can provide a way to model the long-term dependencies among unified labels.

At the t-th step, the GRU takes the hidden state h_{t-1}, the label embedding y^l_{t-1}, and the target clause representation g_t as inputs, and outputs hidden state h_t:

$$h_t = GRU(h_{t-1}, y^l_{t-1} \oplus g_t) \tag{5}$$

where label embedding y^l_{t-1} is a fixed-dimensional dense vector corresponding to the decoded unified label y_{t-1} (each unified label has a corresponding label embedding), $\oplus$ denotes concatenation operator, and the sequence representation h_0 is the initialization of the hidden state at the beginning.

After obtaining the hidden state h_t from GRU at the t-th step, GCA calculates the scaled dot-product attention [52] between the hidden state h_t and the contextualized clause representation of each clause $\{s_1, s_2, \ldots, s_n\}$, and outputs the global context vector ct_t:

$$ct_t = softmax\left(\frac{q_t K^T}{\sqrt{d_k}}\right) V \tag{6}$$

where q_t is the query vector non-linear transformed from h_t, K and V are the key and value matrices packed and mapped from $[s_1, s_2, \ldots, s_n]$, d_k is the dimension of key vector in K.

After the operations of both GRU and GCA at the t-th step, TDU outputs the final representation $\tilde{h}_t$ for the target clause c_t:

$$\tilde{h}_t = ReLU(W_3(h_t \oplus ct_t \oplus g_t) + b_3) \tag{7}$$

where W_3 and b_3 are weight matrix and bias vector respectively.

Unified Labeling To perform unified labeling on the target clause c_t, a softmax layer takes the final representation $\tilde{h}_t$

TABLE I
THE STATISTICS OF THE DATASET

	Documents			**Unified labels**									
	1 pair	**2 pairs**	**>2 pairs**	**N**	**1-E**	**1-C**	**1-B**	**2-E**	**2-C**	**2-B**	**3-E**	**3-C**	**3-B**
Number	1746	177	22	25008	1483	1535	467	88	93	37	7	7	2
Proportion	89.77%	9.10%	1.13%	87.05%	5.16%	5.34%	1.63%	0.31%	0.32%	0.13%	0.02%	0.02%	0.01%

as input, and produces the predictive probability distribution $p(y_t|y_{<t}, X)$ on the set of unified labels $\mathcal{Y}$:

$$P(y_t|y_{<t}, X) = softmax(W_4\tilde{h}_t + b_4) \tag{8}$$

where X is the input clause sequence, $y_{<t}$ denotes the label sequence $\{y_1, y_2, \ldots, y_{t-1}\}$, W_4 and b_4 are weight matrix and bias vector respectively.

D. Training

Given a clause sequence X and the predicted label sequence Y, the goal of model training is to maximize the log probability of label sequence $P(Y|X)$, which can be computed as:

$$logP(Y|X) = \sum_{t=1}^{T} logP(y_t|y_{<t}, X) \tag{9}$$

where T is the number of clauses in input document.

It should be noticed that the labels $y_{<t}$ are used to guide the decoding of next label y_t. For the labels $y_{<t}$, during training, they are the ground-truth labels of clauses $c_{<t}$, while during inference, they are the predicted labels of clauses $c_{<t}$. This discrepancy, called exposure bias [53], leads to a gap between training and inference. To address this problem, we use the scheduled sampling [54] during training, which sets a probability ϵ_i to decide which kinds of labels (i.e., ground-truth or predicted) should be sampled. Formally,

$$\epsilon_i = max(0, 1 - ui) \tag{10}$$

where u refers to the expected convergence speed and i is the number of training epochs.

IV. EXPERIMENTS

A. Dataset

We conduct our experiments on the benchmark ECPE corpus [1], which is first proposed by Gui *source* [9] and reconstructed for ECPE task by Xia and Ding [1]. The ECPE corpus contains 1945 documents. Every document has at least one emotion-cause pair, and each emotion clause has at least one corresponding cause clause. Table I lists the detailed statistics of the corpus, which includes the proportion of documents with different number of emotion-cause pairs and the label distribution of the unified labels.

B. Experimental Setting

In previous studies on ECPE, there exist two experimental settings of data split. One is designed by Xia and Ding [1] and adopted by the majority of studies [2], [4]–[6]. In this setting, the dataset is stochastically split into two parts, 90% for training, and the remaining 10% for testing. The other is designed by Fan *source* [3], where the dataset is stochastically split into a training/development/test set in a ratio of 8:1:1. The development set is used for model selection. To make a fair comparison with the previous ECPE methods, we conduct experiments on both settings to make a comprehensive comparison. To obtain statistically credible results, in both settings, the experiments are repeated 20 times and the average results are reported. We use Precision (P), Recall (R), and F1-score to measure the emotion-cause pair extraction performance based on the predicted emotion-cause pair and the ground-truth emotion-cause pair. In addition, we also evaluate the performance of two sub-tasks: emotion extraction and cause extraction.

The word embedding is pre-trained on Chinese Weibo corpus with the word2vec toolkit [55]. The dimension of word embedding is 200. We fix the word embedding during training. The dimension of hidden state of both LSTM and GRU in our model is set to 100. The dimension of query vector, key vector, and value vector are all set to 50. The dimension of label embedding is 50. The expected convergence speed u is set to 0.03. Our encoder and decoder are trained together based on the Adam optimizer [56], where the batch size and the learning rate are set to 32 and 0.005. The pretrained BERT$_{Chinese}$[1] is used as the candidate clause encoder. Specifically, each clause in the document is feed into the BERT model independently, and the representation of [CLS] is used as the clause representation.

C. Baselines

To evaluate the effectiveness of our approach, we compare our model with three groups of baselines. Among these methods, if the method name is marked with BERT, that means BERT is adopted as the clause encoder. The first group includes three two-step pipeline models proposed by Xia and Ding [1]:

- **Indep** 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** 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** 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 includes five existing one-step end-to-end ECPE models:

[1] [Online]. Available: https://github.com/huggingface/pytorch-pretrained-BERT

TABLE II

THE PERFORMANCE OF OUR MODEL AND THE BASELINES UNDER THE EXPERIMENTAL SETTING ADOPTED BY XIA AND DING [1] (I.E., DATA SPLIT IN A RATIO OF 9:1). THE TOP TWO MAXIMUM VALUE IN EACH COLUMN IS MARKED IN BOLD WHILE THE BEST RESULTS ARE UNDERLINED. AVERAGE RESULTS OVER 20 TIMES STOCHASTIC DATA SPLIT ARE REPORTED

Method	Emotion Extraction			Cause Extraction			Emotion-Cause Pair Extraction		
	P	R	F1	P	R	F1	P	R	F1
Indep [1]	0.8375	0.8071	0.8210	0.6902	0.5673	0.6205	0.6832	0.5082	0.5818
Inter-CE [1]	0.8494	0.8122	0.8300	0.6809	0.5634	0.6151	0.6902	0.5135	0.5901
Inter-EC [1]	0.8364	0.8107	0.8230	0.7041	0.6083	0.6507	0.6721	0.5705	0.6128
E2EECPE [5]	0.8595	0.7915	0.8238	0.7062	0.6030	0.6503	0.6478	0.6105	0.6280
MTNECP [6]	0.8662	0.8393	0.8520	0.7400	0.6378	0.6844	0.6828	0.5894	0.6321
IE-CNN+CRF [23]	0.8614	0.7811	0.8188	0.7348	0.5841	0.6496	0.7149	0.6279	0.6686
ECPE-2D+BERT [2]	0.8627	**0.9221**	**0.8910**	0.7336	0.6934	0.7123	0.7292	0.6544	0.6889
RANKCP+BERT [4]	0.8679	**0.8926**	**0.8797**	0.7262	**0.7646**	**0.7437**	0.6821	**0.7483**	**0.7121**
BiLSTM-Softmax	0.8672	0.7945	0.8289	0.7324	0.6088	0.6640	0.6836	0.5713	0.6217
BiLSTM-CRF	**0.8720**	0.7870	0.8267	0.7135	0.6517	0.6805	0.6749	0.6084	0.6389
BERT-Softmax	0.8399	0.8501	0.8441	**0.7693**	0.7071	0.7355	0.7227	0.6669	0.6927
BERT-CRF	0.8400	0.8510	0.8471	0.7657	0.7195	0.7409	0.7307	0.6682	0.6968
UTOS	0.8610	0.7925	0.8250	0.7189	0.6496	0.6802	0.6911	0.6193	0.6524
UTOS+BERT	**0.8815**	0.8321	0.8556	0.7671	**0.7320**	**0.7471**	**0.7389**	**0.7062**	**0.7203**

- **E2EECPE** is a multi-task learning model that can extract emotions, causes and emotion-cause pairs simultaneously in an end-to-end manner [5].
- **MTNECP** is a multi-task neural network to perform emotion-cause pair extraction in a unified model, where the representation of clause is shared across tasks [6].
- **TRANS+BERT** is a transition-based model which tackles the ECPE task through a procedure of parsing-like directed graph construction [3].
- **IE-CNN+CRF** is a unified sequence labeling model based on stacked CNN and CRF [23].
- **SLNT+BERT** is another unified sequence labeling model based on BERT and Softmax [22].
- **ECPE-2D+BERT** is an end-to-end neural model which represents emotion-cause pairs by a 2D representation scheme and uses a 2D transformer module to model the interactions of emotion-cause pairs [2].
- **RANKCP+BERT** is a one-step neural model which learns the clause representation based on graph attention and tackles emotion-cause pair extraction from a ranking perspective [4]. To make a fair comparison, the version of RANKCP without using sentiment lexicon is adopted as a baseline in this paper, considering that the additional knowledge base is not used in all other baseline methods and our model.

The third group of baselines includes four classic sequence labeling models, which are implemented to demonstrate the effectiveness of the proposed unified labeling scheme:

- **BiLSTM-Softmax** learns the clause representation by two Bi-LSTM layers of hierarchical structure and uses a softmax layer for sequence labeling.
- **BiLSTM-CRF** learns the clause representation by two Bi-LSTM layers of hierarchical structure and uses CRF for sequence labeling [45].
- **BERT-Softmax** adopts BERT as the clause encoder and uses a softmax layer for sequence labeling.
- **BERT-CRF** adopts BERT as the clause encoder and uses CRF for sequence labeling.

D. *Results and Analysis*

We report the performance of our model and the baselines under two data splitting settings (i.e., 9:1 and 8:1:1) in Table II and Table III, respectively. The results of different models are shown in Table II. From the results we can see that:

For the target emotion-cause pair extraction task, we can find that our proposed UTOS outperforms the three two-step models (i.e., Indep, Inter-CE, and Inter-EC), E2EECPE, and MTNECP on precision, recall, and F1-score (e.g., UTOS outperforms MTNECP by 2.03% on F1-score) in Table II. By further using BERT as the clause encoder in our model, UTOS+BERT can achieve the overall best performance on F1-score (e.g., UTOS+BERT outperforms RANKCP+BERT by 0.82% and 0.64% on F1-score in Table II and Table III, respectively). When compared with the other two unified sequence labeling models based on different tagging schemes and model structures (i.e., IE-CNN+CRF and SLSN+BERT), we can find that UTOS+BERT outperforms IE-CNN+CRF and SLNT+BERT by about 5.2% and 1.3% on F1-score, respectively. These results demonstrate that our proposed model UTOS is effective for the extraction of emotion-cause pairs. In addition, from Table II, we can find that the precision of UTOS+BERT is 5.67% higher than that of RANKCP+BERT on the ECPE task, while the recall of RANKCP+BERT is 4.21% higher than that of UTOS+BERT. This is mainly because that our unified sequence labeling scheme extracts the pairs in the sequence jointly, which would implicitly take the consistency among extracted pairs into consideration, and thus tend to extract fewer but more consistent pairs than the the explicit extraction

TABLE III
THE PERFORMANCE OF OUR MODEL AND THE BASELINES UNDER THE EXPERIMENTAL SETTING ADOPTED BY FAN *SOURCE* [3] (I.E., DATA SPLIT IN A RATIO OF 8:1:1). THE TOP TWO MAXIMUM VALUE IN EACH COLUMN IS MARKED IN BOLD WHILE THE BEST RESULTS ARE UNDERLINED. AVERAGE RESULTS OVER 20 TIMES STOCHASTIC DATA SPLIT ARE REPORTED

Method	Emotion Extraction			Cause Extraction			Emotion-Cause Pair Extraction		
	P	R	F1	P	R	F1	P	R	F1
SLNT+BERT [22]	0.8196	0.7329	0.7739	0.7490	0.6602	0.7018	**0.7243**	0.6366	0.6776
TRANS+BERT [3]	**0.8716**	0.8244	0.8474	**0.7562**	0.6471	0.6974	**0.7374**	0.6307	0.6799
RANKCP+BERT [4]	**0.8708**	**0.8784**	**0.8744**	0.7127	**0.7322**	**0.7221**	0.6603	**0.7107**	**0.6843**
BiLSTM-Softmax	0.8373	0.7707	0.8023	0.6870	0.5722	0.6235	0.6358	0.5516	0.5898
BiLSTM-CRF	0.8349	0.7612	0.7959	0.6909	0.6051	0.6447	0.6291	0.5738	0.5994
BERT-Softmax	0.8132	**0.8512**	0.8316	**0.7533**	0.7029	0.7153	0.6416	0.6678	0.6539
BERT-CRF	0.8191	0.8476	0.8330	0.7425	0.7057	0.7199	0.6445	0.6764	0.6594
UTOS	0.8207	0.7707	0.7947	0.6781	0.6064	0.6398	0.6531	0.5835	0.6159
UTOS+BERT	0.8649	0.8293	**0.8491**	0.7418	**0.7084**	**0.7281**	0.7104	**0.6812**	**0.6907**

scheme adopted by RANKCP+BERT. We will further discuss this phenomenon in the case study.

By comparing the third group of baselines with previous two groups of baselines, we can find that BiLSTM-CRF outperforms two-step models, ECEECPE, and MTNECP on F1-score. By using BERT as the clause encoder, BERT-Softmax and BERT-CRF have an improvement of about 7% and 6% in Table II and Table III. Besides, they even outperform ECPE-2D+BERT on F1-score in Table II. This indicates that our proposed unified sequence labeling scheme is effectiveness for the ECPE task and even a simple sequence labeling method can work well on the ECPE task.

For the emotion extraction and cause extraction sub-tasks,[2] we can find that UTOS+BERT achieves the overall best cause extraction performance on F1-score in both Table II and Table III, while the best emotion extraction performance on F1-score is achieved by ECPE-2D+BERT in Table II and RANKCP+BERT in Table III. By carefully observing the relationship between the performance of models on two sub-tasks and the target ECPE task, we can find that models with good performance on cause extraction sub-task usually achieve good performance on the target ECPE task. This implies that once the accuracy of emotion extraction reaches a certain level, the performance on ECPE task may be more related to cause extraction than emotion extraction. In addition, we can find that the performance of BiLSTM-CRF on the tasks of emotion extraction and cause extraction is comparable with the corresponding performance of our UTOS model, but UTOS outperforms BiLSTM-CRF on the ECPE task. This may be because that UTOS has a better pairing capability than BiLSTM-CRF, which is further explored in the following section of error analysis.

Considering that most documents contain only one emotion-cause pair, we attempt to further verify the performance of our model on documents with multiple emotion-cause pairs. Same as Wei *source* [4], we divide the test set into two subsets: one subset contains documents with only one emotion-cause pair, and the other subset contains documents with two or more emotion-cause pairs.

TABLE IV
COMPARATIVE RESULTS FOR DOCUMENTS WITH ONLY ONE AND MORE THAN ONE EMOTION-CAUSE PAIR. THE BEST RESULTS ARE IN BOLD

# Pairs	Model	P	R	F1
One per doc.	**RANKCP+BERT**	0.6731	**0.7964**	0.7327
	UTOS+BERT	**0.7358**	0.7548	**0.7446**
Two or more per doc.	**RANKCP+BERT**	0.6762	**0.4813**	0.5651
	UTOS+BERT	**0.7612**	0.4622	**0.5734**

As shown in Table IV, the best baseline model RANKCP+BERT is used to make a comparison with our model UTOS+BERT. It can be seen that our model consistently outperforms RANKCP+BERT by 1.19% and 0.83% on F1-score on the test documents with one pair and with two or more pairs respectively. This indicates that our model can work well when there are multiple pairs in the document. Besides, it is worth noting that the performance of both models on test documents with two or more pairs is worse than that with one pair, which means that processing documents with multiple pairs is a bottleneck of the ECPE task.

E. Model Analysis

To avoid the influence of BERT, in this part, we conduct model analysis on the basic UTOS model. We explore how the components of the proposed UTOS model affect its performance on the ECPE task and analyze the robustness of the UTOS model on test documents that only has unpaired emotion. Unless otherwise specified, all subsequent analyses are conducted under the more widely used experimental setting (i.e., the stochastic data split of 9:1).

1) Ablation Study: In this part, we explore the effects of the components designed specific to the unified sequence labeling, by removing each of them from UTOS individually. As shown in the right part of Figure 2, these components include: the connection from the target encoder to the sequence decoder (i.e., TC1), the connection from the target encoder to the unified

[2] For the models under the proposed unified labeling scheme, the labels with E and B are treated as emotion clause, and the labels with C and B are treated as cause clause.

TABLE V
ABLATION STUDY OF THE PROPOSED UTOS MODEL

		P	R	F1
Full model		0.6911	0.6193	0.6524
Target Information	**– TC1**	0.6858	0.6089	0.6441
	– TC2	0.6980	0.6048	0.6457
	– TC1&TC2	0.6308	0.3533	0.4515
Context Information	**– SR**	0.6934	0.6081	0.6491
	– DS	0.6896	0.6050	0.6439
	– GCA	0.6790	0.6221	0.6476
	– SR&DS&GCA	0.6917	0.5922	0.6367
Label Information	**– LE**	0.6862	0.5920	0.6343

TABLE VI
THE PERFORMANCE OF USING DIFFERENT SAMPLING STRATEGIES. AVERAGED RESULTS OVER 5 RUNS ARE REPORTED

Strategy	P	R	F1
Without sampling	0.6640	0.6242	0.6429
Uniform sampling	0.6902	0.6125	0.6481
Always sampling	0.6881	0.6138	0.6484
Scheduled sampling	0.6911	0.6193	0.6524

TABLE VII
THE STATISTICS OF ERROR EMOTION-CAUSE PAIRS

Category	Number	Proportion
Emotion error	2	2.6%
Cause error	26	33.8%
Both error	31	40.3%
Missing error	18	23.4%

TABLE VIII
THE PERFORMANCE OF OUR MODEL ON TWO SYNTHETIC DATASET

Test Data	Synthesis Strategy	Emotion Extraction			Correctly-Extracted Emotion	
		P	R	F1	Paired	Unpaired
I	Clause Removement	0.7289	0.8013	0.7634	0.8036	0.1964
II	Document Segmentation	0.6944	0.8197	0.7519	0.2069	0.7931

labeling (i.e., TC2), the sequence representation used as the initial state h_0 of GRU (denoted as SR), the decoded state from GRU to the unified labeling (i.e., DS), global context vector from GCA to the unified labeling (i.e., GCA), and the label embedding from former decoded label to sequence decoder (i.e., LE). We divide these components into three groups according to the information they carry. TC1 and TC2 provide the information of target clause to the sequence decoder and the unified labeling, thus they belong to the group of *target information*. SR, DS, and GCA provide the information of the entire clause sequence and the clause context to the unified labeling, thus they belong to the group of *context information*. LE provides the information of former decoded unified label to the sequence decoder, thus it belongs to the group of *label information*.

As shown in Table V, for the *target information*, by removing either TC1 or TC2 from the model, the performance drops a little (about 0.7% on F1-score). But if both of them are removed from the model, the performance drops a lot (about 20%). This indicates that the information of target clause is essential for the unified sequence labeling, and using one of them is only a little worse than using both of them. For the *context information*, by removing one of the three context related components (i.e., SR, DS and GCA) from the model, the performance drops a little (about 0.3% for RS, 0.8% for DS and 0.45% for GCA on F1-score).

But if all of them are removed[3] from the model, the performance drops to 0.6367 on F1-score, which is close to the performance of BiLSTM-CRF (refers to Table II). This indicates that the information of context is also important for the unified sequence labeling, and DS is a little more effective than GCA. For the *label information*, by removing LE from the model, the performance drops to 0.6343 on F1-score, which indicates that the information of former decoded label is important for the unified sequence labeling.

2) Effects of Sampling Strategy: In this part, we consider the exposure bias problem and explore the effect of using different sampling strategies to mitigate it. We compare the adopted scheduled sampling to other three sampling strategies: without sampling, uniform sampling, and always sampling. The without sampling strategy and uniform sampling strategy correspond to the case $\epsilon_i = 0$ and $\epsilon_i = 0.5$ in Eq. 10 respectively. The only difference between the always sampling and scheduled sampling is that the always sampling gets the ϵ_i once per sequence instead of once per clause.

As shown in Table VI, we can find that the scheduled sampling strategy achieves the best performance compared to other strategies. This indicates that the way of changing ϵ_i according to the number of epochs is effective. Besides, we can find that the uniform sampling strategy also achieves a better performance than the without sampling strategy. This phenomenon indicates that it is effective to narrow the gap between training and inference, and even the uniform sampling strategy can bring a certain performance improvement.

3) Robustness on Documents With Unpaired Emotion: In this part, we explore the robustness of our model on test document that only has unpaired emotion (unpaired cause is meaningless, thus not considered). Due to the reason that there is no unpaired emotion in this dataset, we synthesize documents that only contain emotion for experiments. Specifically, we design two synthesis strategies: clause removing and document segmentation. The former strategy generates synthetic documents by simply removing the cause clauses in the test documents. The latter strategy first splits a test document into segments with cause clause as separator, and then select one segment that contains emotion clause as a synthetic document. As shown in Table VIII, we test the UTOS (trained on original data) on these two sets of synthetic documents and report the performance.

[3] In this case, LE will be directly fed to the unified labeling to avoid to be removed.

TABLE IX
THE STATISTICS OF ILLEGAL PAIRING. THE BEST RESULTS ARE IN BOLD

Method	Number	Percentage
UTOS	**3**	**1.5%**
UTOS w/o LE	31	15.9%
BiLSTM-CRF	35	17.8%

By observing the performance on the emotion extraction task in Table VIII, we can find that the performance of the UTOS model on the synthetic test data and the real test data (in Table II) is very similar, only the precision is slightly reduced. This indicates that the UTOS's ability of emotion extraction is robust to the synthetic data. Furthermore, among the correctly-extracted emotions, we can find that some of them are paired with a cause by UTOS model (0.8036 for test data I and 0.2069 for test data II), while also some of them are correctly predicted as unpaired emotions. This indicates that UTOS can handle some of the case of unpaired emotion correctly, but due to the reason that UTOS is trained on the real training data, UTOS still tends to find cause(s) for an emotion.

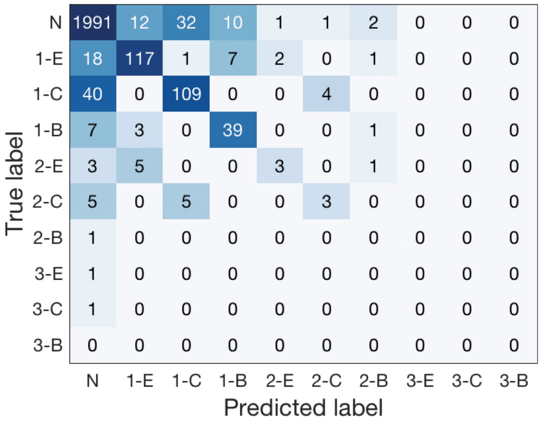
F. Error Analysis

In this section, we perform error analysis on our model from three aspects: emotion-cause pair extraction, illegal sequence labeling, and unified label prediction.

1) Error Analysis of Emotion-Cause Pair Extraction: For the extraction of emotion-cause pairs, we report four categories of errors on test set in Table VII. We collect all the error pairs on test set from the perspectives of both precision and recall, and divided them into four categories: emotion error, cause error, both error, and missing error. The former three categories of errors refer that in the predicted pairs, either one of emotion and cause or both them are incorrectly extracted. The missing error refers that there exist some ground-truth pairs that are not extracted by our model. As shown in Table VII, the proportions of cause error and both error are relatively large. This implies that model has some shortcomings in accurately extracting the cause. Besides, there exist many missing errors, which implies that the ability of our model on the extraction of both emotion and cause should be further strengthened. The case that there are relatively few emotion errors implies that once the cause is identified, our model has the ability to find its corresponding emotion.

2) Error Analysis of Illegal Pairing: Under the unified labeling scheme, the pairing part of unified label is used for clause pairing. To verify whether the pairing parts are assigned self-consistently to a sequence, we explore the phenomenon of illegal pairing, which refers that there exists an emotion or cause clause having no corresponding pairing clause to form an emotion-cause pair. For example, if a predicted label sequence contains the label `1-E` but does not contain the label `1-B` or the label `1-C`, then the emotion clause labeled as `1-E` does not have corresponding cause clause.

We count the documents with such kind of clause and report the statistical results on test set in Table IX. Here, we compare our UTOS model with BiLSTM-CRF which uses the transition matrix to model the label dependency [42].



	N	1-E	1-C	1-B	2-E	2-C	2-B	3-E	3-C	3-B
N	1991	12	32	10	1	1	2	0	0	0
1-E	18	117	1	7	2	0	1	0	0	0
1-C	40	0	109	0	0	4	0	0	0	0
1-B	7	3	0	39	0	0	1	0	0	0
2-E	3	5	0	0	3	0	1	0	0	0
2-C	5	0	5	0	0	3	0	0	0	0
2-B	1	0	0	0	0	0	0	0	0	0
3-E	1	0	0	0	0	0	0	0	0	0
3-C	1	0	0	0	0	0	0	0	0	0
3-B	0	0	0	0	0	0	0	0	0	0

Fig. 3. Confusion matrix of unified label prediction on test set.

We can find that the number of illegal pairings generated by our model is significantly less than that generated by BiLSTM-CRF. Recalling the phenomenon in Table II that BiLSTM-CRF performs comparably with UTOS on tasks of emotion extraction and cause extraction but worse than UTOS on the ECPE task, we can conclude that UTOS has a better pairing capability than BiLSTM-CRF.

In addition, we further ablate the label embedding from our UTOS model (i.e., UTOS w/o LE). We can find that the number of illegal pairings rises from 3 to 31 which is very close to the performance of BiLSTM-CRF. This shows that label embedding is important for our UTOS to model the dependencies among unified labels and achieve better performance on clause pairing.

3) Error Analysis of Unified Label Prediction: For the prediction of unified labels, we report the confusion matrix of unified labels on test set in Figure 3. The off-diagonal elements can be divided into three types of errors: `N`-related error, pairing error, and content error. We can find that the `N`-related error occupies a large proportion of errors, which suggests that our model should be further strengthened. For the pairing error, there exist some cases that the content part is right but the pairing part is wrong. This type of error is caused by the pairing part shifting, which does not necessarily affect the extraction of emotion-cause pair seriously. For the content error, we can find that `E` is sometimes predicted as `B`, and vice versa. A more interesting phenomenon is that there are few cases that the context part `E` or `B` is mispredicted as the context part `C`, and vice versa. This indicates that our model can effectively distinguish between emotion and cause. In addition, we can find that there only two clauses are labeled with pairing index `3` and they are not predicted correctly. This may be because that there are only a few documents that require the pairing index `3` in the training set, so the corresponding weights in the weight matrix before the softmax layer may not be trained well. May be a more balanced dataset is needed to further validate the model performance on documents with larger pairing index.

TABLE X
THREE EXAMPLES FOR THE CASE STUDY

Document	**Ground-truth**	**RANKCP+BERT**	**UTOS+BERT**
In the past few days (c_1), many people have expressed their anger at Yuan's behavior (c_2). At the same time (c_3), in order to prevent them from suffering too much pressure and causing secondary harm (c_4), it also calls on the society to give them more privacy protection (c_5).	(c_2, c_2)	(c_2, c_1), (c_2, c_2), (c_2, c_4)	(c_2, c_2)
I have been following the progress of the incident (c_1), and while being angry at the man's behavior (c_2), I also expressed heartache for the lack of self-defense awareness of the female (c_3). When a female is harmed (c_4), the first consideration should not be to escape (c_5), but to stand up bravely and use legal weapons to protect their legal rights (c_6).	(c_2, c_2), (c_3, c_3)	(c_2, c_2), (c_3, c_2), (c_3, c_3)	(c_2, c_2), (c_3, c_3)
Because I broke up with my girlfriend (c_1), and couldn't find her (c_2), I was depressed and drank a bottle of wine. (c_3). Then I got drunk, (c_4), and such a dramatic scene happened (c_5).	(c_3, c_1), (c_3, c_2)	(c_2, c_1), (c_3, c_1), (c_3, c_2)	(c_3, c_1), (c_3, c_2)

G. Case Study

For the case study, we select three examples in the test dataset to analyze the difference of performance between our UTOS+BERT model and the best baseline model RANKCP+BERT on the ECPE task. Table X shows the three examples and the corresponding three kinds of emotion-cause pairs: the ground-truth emotion-cause pairs, the extracted pairs by RANKCP+BERT, and the extracted pairs by UTOS+BERT.

In the first example, we can find that both UTOS+BERT and RANKCP+BERT extract the correct emotion-cause pair (c_2, c_2), but RANKCP+BERT extracts extra other two pairs (c_2, c_1) and (c_2, c_4). Among the clauses involved in these two pairs, c_1 is a clause describing the time and c_4 is a long clause describing the intention. While both of them are neither emotion clause nor cause clause, RANKCP+BERT identifies them as the cause of the clause c_2. This suggests that compared with UTOS+BERT, RANKCP+BERT may prefer to identify more causes for an identified emotion.

In the second example, we can find that both UTOS+BERT and RANKCP+BERT extract the correct emotion-cause pairs (c_2, c_2) and (c_3, c_3), but RANKCP+BERT extracts an extra pair (c_3, c_2). Although c_3 and c_2 are emotion and cause respectively, c_2 is not the corresponding cause of c_3 and they thus can not form a valid emotion-cause pair. This suggests that even the emotion and cause are identified correctly, RANKCP+BERT may extract some invalid pairs.

In the third example, it shows a case that one emotion clause is paired with two cause clauses. We can find that both UTOS+BERT and RANKCP+BERT extract the correct emotion-cause pairs (c_3, c_1) and (c_3, c_2). This indicats that both UTOS+BERT and RANKCP+BERT are able to handle such one-to-many case (i.e., one emotion clause is paired with multiple cause clauses). Besides, RANKCP+BERT extracts an extra pair (c_2, c_1). Again, the cause clause c_2 is predicted as both emotion clause and cause clause, which indicates that RANKCP is easy to output inconsistent predictions.

The above three examples show that compared with UTOS+BERT, RANKCP+BERT tends to generate more emotion-cause pairs. This also explains why RANKCP+BERT has a lower precision but a higher recall than our UTOS+BERT.

H. Discussion

In this section, we make discussions on our proposed unified labeling scheme from two aspects. One is the coverage of all potential emotion-cause pairs, and the other is the upper limit of the pairing part index.

1) The Coverage of all Potential Pairs: Since the coverage of all potential pairs is hard to calculated theoretically, we analyze the possible cases of emotion-cause pairs that cannot be extracted by our scheme and validate the upper bound performance of our scheme on the whole dataset in the following.

Due to that each clause in a document is only labeled once in our unified labeling scheme, it is intractable that a clause should be assigned two or more unified labels. For example, if there are two emotion-cause pairs in a document with three clauses (denoted as c_1, c_2, and c_3) and the two pairs are (c_1, c_2) and (c_2, c_3) respectively, then the clause c_2 needs to be labeled with unified labels of both `1-E` and `2-C`. Thus, labeling c_2 with either `1-E` or `2-C` lead to the missing of one of the two pairs. To validate how frequently such kind of uncovered cases of our scheme occurs in the dataset, we check each document in dataset. For a certain document, if there does not exist a sequence of unified labels that can derive all the emotion-cause pairs in it, this document is considered to be uncovered by our scheme. Finally, we find that only one document is uncovered by our scheme and the upper bound F1 score for our scheme on this dataset is 99.97%, which implies that our unified labeling scheme is practical.

2) The Upper Limit of Pairing Part Index: From the task definition, it can be found that the maximum value `k` of pairing part index is determined by the training set. Thus, the model UTOS trained on the training set would have an upper limit of the pairing part index, and can not handle the situation that the test document need to be labeled with a pairing part index larger than `k`.

To eliminate this limitation, a promising solution is to first divide the test document into several text segments, then label these segments separately and aggregate the results. As for how to segment the document and how to aggregate the results, so as to achieve a good performance of emotion-cause pair extraction, this is a direction worthy of further exploration.

V. Conclusion

In this paper, we reframe the ECPE task as a unified sequence labeling problem through the specially-designed unified label. The specially-designed unified label consists of the content part and the pairing part. The content part indicates whether the clause contains emotion or cause, while the pairing part indicates which clauses should be paired. Then we propose the UTOS model based on sequence-to-sequence learning to address it by performing end-to-end unified sequence labeling. Our proposed method can take full advantage of the meaning of whole sequence and the previous label during decoding process. The experimental results demonstrate the effectiveness of both the proposed unified sequence labeling scheme and UTOS model.

References

[1] R. Xia and Z. Ding, "Emotion-cause pair extraction: A new task to emotion analysis in texts," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, pp. 1003–1012.

[2] Z. Ding, R. Xia, and J. Yu, "ECPE-2D: Emotion-cause pair extraction based on joint two-dimensional representation, interaction and prediction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3161–3170.

[3] C. Fan, C. Yuan, J. Du, L. Gui, M. Yang, and R. Xu, "Transition-based directed graph construction for emotion-cause pair extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3707–3717.

[4] P. Wei, J. Zhao, and W. Mao, "Effective inter-clause modeling for end-to-end emotion-cause pair extraction," in *Proc. 58th Annu. Meeting Assoc. Comput. Linguistics*, 2020, pp. 3171–3181.

[5] H. Song, C. Zhang, Q. Li, and D. Song, "End-to-end emotion-cause pair extraction via learning to link," 2020, *arXiv:2002.10710*.

[6] S. Wu, F. Chen, F. Wu, Y. Huang, and X. Li, "A Multi-Task Learning Neural Network for Emotion-Cause Pair Extraction," *ECAI*, vol. 325, pp. 2212–2219, 2020.

[7] Z. Cheng, Z. Jiang, Y. Yin, H. Yu, and Q. Gu, "A symmetric local search network for emotion-cause pair extraction," in *Proc. 28th Int. Conf. Comput. Linguistics, COLING 2020*, pp. 139–149.

[8] S. Y. M. Lee, Y. Chen, and C.-R. Huang, "A text-driven rule-based system for emotion cause detection," in *Proc. NAACL HLT 2010 Workshop Comput. Approaches Anal. Gener. Emotion Text*, 2010, pp. 45–53.

[9] L. Gui, D. Wu, R. Xu, Q. Lu, and Y. Zhou, "Event-driven emotion cause extraction with corpus construction," in *Proc. 2016 Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 1639–1649.

[10] Y. Chen, S. Y. M. Lee, S. Li, and C. Huang, "Emotion cause detection with linguistic constructions," in *Proc. 23rd Int. Conf. Comput. Linguistics*, 2010, pp. 179–187.

[11] K. Gao, H. Xu, and J. Wang, "Emotion cause detection for chinese micro-blogs based on ecocc model," in *Proc. Adv. Knowl. Discov. Data Mining - 19th Pacific-Asia Conf.*, 2015, pp. 3–14.

[12] L. Gui, L. Yuan, R. Xu, B. Liu, Q. Lu, and Y. Zhou, "Emotion cause detection with linguistic construction in chinese weibo text," in *Proc. Natural Lang. Process. Chin. Comput.*, 2014, pp. 457–464.

[13] D. Ghazi, D. Inkpen, and S. Szpakowicz, "Detecting emotion stimuli in emotion-bearing sentences," in *Proc. Comput. Linguistics Intell. Text Process.*, 2015, pp. 152–165.

[14] L. Gui, J. Hu, Y. He, R. Xu, Q. Lu, and J. Du, "A question answering approach for emotion cause extraction," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2017, pp. 1593–1602.

[15] X. Li, K. Song, S. Feng, D. Wang, and Y. Zhang, "A co-attention neural network model for emotion cause analysis with emotional context awareness," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 4752–4757.

[16] Y. Chen, W. Hou, X. Cheng, and S. Li, "Joint learning for emotion classification and emotion cause detection," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2018, pp. 646–651.

[17] Z. Ding, H. He, M. Zhang, and R. Xia, "From independent prediction to reordered prediction: Integrating relative position and global label information to emotion cause identification," in *33rd AAAI Conf. Artif. Intell.,* 2019, pp. 6343–6350.

[18] X. Li, S. Feng, D. Wang, and Y. Zhang, "Context-aware emotion cause analysis with multi-attention-based neural network," *Knowl.-Based Syst.*, vol. 174, pp. 205–218, 2019.

[19] C. Fan *et al.*, "A knowledge regularized hierarchical approach for emotion cause analysis," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 5613–5623.

[20] R. Xia, M. Zhang, and Z. Ding, "RTHN: A RNN-transformer hierarchical network for emotion cause extraction," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 5285–5291.

[21] S. Zheng, F. Wang, H. Bao, Y. Hao, P. Zhou, and B. Xu, "Joint extraction of entities and relations based on a novel tagging scheme," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1227–1236.

[22] C. Yuan, C. Fan, J. Bao, and R. Xu, "Emotion-cause pair extraction as sequence labeling based on a novel tagging scheme," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 3568–3573.

[23] X. Chen, Q. Li, and J. Wang, "A unified sequence labeling model for emotion cause pair extraction," in *Proc. 28th Int. Conf. Comput. Linguistics, 2020*, pp. 208–218.

[24] K. Cho *et al.*, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[25] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3104–3112.

[26] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

[27] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[28] J. Tan, X. Wan, and J. Xiao, "Abstractive document summarization with a graph-based attentional neural model," in *Proc. 55th Annu. Meeting Assoc. Comput. Linguistics*, 2017, pp. 1171–1181.

[29] H. Kameoka, K. Tanaka, D. Kwasny, T. Kaneko, and N. Hojo, "ConvS2S-VC: Fully convolutional sequence-to-sequence voice conversion," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 1849–1863, 2020.

[30] Q. T. Do, S. Sakti, and S. Nakamura, "Sequence-to-sequence models for emphasis speech translation," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 26, no. 10, pp. 1873–1883, Oct. 2018.

[31] J. Nam, E. L. Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5419–5429.

[32] P. Yang, X. Sun, W. Li, S. Ma, W. Wu, and H. Wang, "SGM: Sequence generation model for multi-label classification," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3915–3926.

[33] D. Zhang, X. Ju, J. Li, S. Li, Q. Zhu, and G. Zhou, "Multi-modal multi-label emotion detection with modality and label dependence," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2020, pp. 3584–3593.

[34] P. Colombo, E. Chapuis, M. Manica, E. Vignon, G. Varni, and C. Clavel, "Guiding attention in sequence-to-sequence models for dialogue act prediction," in *Proc. 34th AAAI Conf. Artif. Intell.*, AAAI Press, 2020, pp. 7594–7601.

[35] Z. Li, J. Cai, S. He, and H. Zhao, "Seq2seq dependency parsing," in *Proc. 27th Int. Conf. Comput. Linguistics*, 2018, pp. 3203–3214.

[36] X. Ma, Z. Hu, J. Liu, N. Peng, G. Neubig, and E. H. Hovy, "Stack-pointer networks for dependency parsing," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1403–1414.

[37] D. Ma, S. Li, F. Wu, X. Xie, and H. Wang, "Exploring sequence-to-sequence learning in aspect term extraction," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, pp. 3538–3547.

[38] G. Lample, M. Ballesteros, S. Subramanian, K. Kawakami, and C. Dyer, "Neural architectures for named entity recognition," in *Proc. North Amer. Chapter Assoc. Comput. Linguistics, HLT*, 2016, pp. 260–270.

[39] Y. Zhang and J. Yang, "Chinese NER using lattice LSTM," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 1554–1564.
[40] Z. Li, J. Chao, M. Zhang, and W. Chen, "Coupled sequence labeling on heterogeneous annotations: POS tagging as a case study," in *Proc. 53rd Annu. Meeting Assoc. Comput. Linguistics 7th Int. Joint Conf. Natural Lang. Process. Asian Federation Natural Lang. Process.*, 2015, pp. 1783–1792.
[41] H. Xu, B. Liu, L. Shu, and P. S. Yu, "Double embeddings and CNN-based sequence labeling for aspect extraction," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 592–598.
[42] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. P. Kuksa, "Natural language processing (almost) from scratch," *J. Mach. Learn. Res.*, vol. 12, pp. 2493–2537, 2011.
[43] Z. Huang, W. Xu, and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging," CoRR, 2015, *arXiv:1508.01991*. [Online]. Available: http://arxiv.org/abs/1508.01991
[44] J. P. C. Chiu and E. Nichols, "Named entity recognition with bidirectional LSTM-CNNS," *Trans. Assoc. Comput. Linguistics*, vol. 4, pp. 357–370, 2016.
[45] X. Ma and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF," in *Proc. 54th Annu. Meeting Assoc. Comput. Linguistics*, vol. 1, pp. 1064–1074, 2016.
[46] H. Yan, B. Deng, X. Li, and X. Qiu, "TENER: Adapting transformer encoder for named entity recognition," CoRR, 2019, *arXiv:1911.04474*.
[47] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics: Human Lang. Technol.*, 2019, pp. 4171–4186.
[48] L. Cui and Y. Zhang, "Hierarchically-refined label attention network for sequence labeling," in *Proc. Conf. Empirical Methods Natural Lang. Process. 9th Int. Joint Conf. Natural Lang. Process.*, 2019, pp. 4113–4126.
[49] F. Zhai, S. Potdar, B. Xiang, and B. Zhou, "Neural models for sequence chunking," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, pp. 3365–3371.
[50] Y. Liu, F. Meng, J. Zhang, J. Xu, Y. Chen, and J. Zhou, "GCDT: A global context enhanced deep transition architecture for sequence labeling," in *Proc. 57th Conf. Assoc. Comput. Linguistics*, 2019, pp. 2431–2441.
[51] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[52] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[53] M. Ranzato, S. Chopra, M. Auli, and W. Zaremba, "Sequence level training with recurrent neural networks," in *Proc. 4th Int. Conf. Learn. Representations*, 2016.
[54] S. Bengio, O. Vinyals, N. Jaitly, and N. Shazeer, "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks," *NIPS*, pp. 1171–1179, 2015, *arXiv:1506.03099*.
[55] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," *NIPS*, pp. 3111–3119, 2013.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Representations*, 2015.

Zifeng Cheng received the B.E. degree from Qingdao University, Qingdao, China, in 2018. He is currently working toward the Ph.D. degree with the Department of Computer Science, Nanjing University, Nanjing, China. His research interests include natural language processing, sentiment analysis, and machine learning.

Zhiwei Jiang received the Ph.D. degree in computer science from Nanjing University, Nanjing, China, in 2018. He is currently an Assistant Professor with the Department of Computer Science and Technology, Nanjing University. His research interests include natural language processing and machine learning.

Yafeng Yin received the Ph.D. degree in computer science from Nanjing University, Nanjing, China, in 2017. She is currently an Assistant Professor with the Department of Computer Science and Technology, Nanjing University. Her research interests include human activity recognition, mobile sensing, and wearable computing. She has authored or coauthored more than 20 papers in the IEEE TRANSACTIONS ON MOBILE COMPUTING, IEEE TRANSACTIONS ON COMPUTERS, *ACM Transactions on Sensor Networks*, ACM UbiComp, and IEEE INFOCOM.

Na Li received the B.E. degree in 2014 from Xiangtan University, Xiangtan, China and the M.Sc. degree in 2017 from Nanjing University, Nanjing, China, where she is currently working toward the Ph.D. degree with the Department of Computer Science. Her research interests include natural language processing, information extraction, and machine learning.

Qing Gu is currently a Professor and Ph.D. Advisor with the Department of Computer Science and Technology, Nanjing University, Nanjing, China, and a Member of the National Key Laboratory of Novel Software Technology. His research interests include natural language processing, machine learning, and software engineering.