An End-to-End Multi-Task Learning to Link Framework for Emotion-Cause Pair Extraction

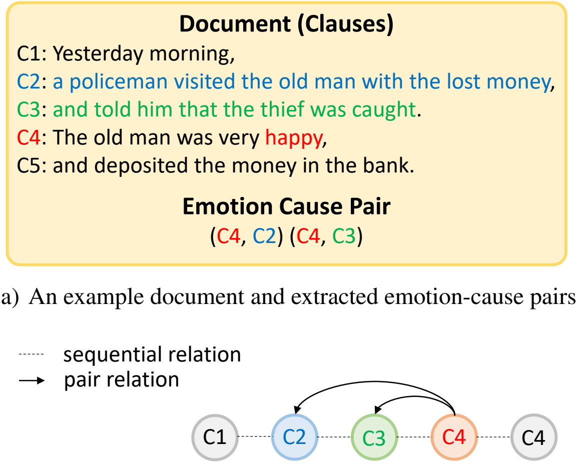
Haolin Song[1] *, **Chen Zhang**[1] *, **Qiuchi Li**[2], **Dawei Song**[1]
[1]Beijing Institute of Technology, Beijing, China
[2]University of Padua, Padua, Italy
{hlsong,czhang,dwsong}@bit.edu.cn, qiuchili@dei.unipd.it

Abstract

Emotion-cause pair extraction (ECPE), as an emergent natural language processing task, aims at jointly investigating emotions and their underlying causes in documents. It extends the previous emotion cause extraction (ECE) task, yet without requiring a set of pre-given emotion clauses as in ECE. Existing approaches to ECPE generally adopt a two-stage method, i.e., (1) emotion and cause detection, and then (2) pairing the detected emotions and causes. Such pipeline method, while intuitive, suffers from two critical issues, including error propagation across stages that may hinder the effectiveness, and high computational cost that would limit the practical application of the method. To tackle these issues, we propose a multi-task learning model that can extract emotions, causes and emotion-cause pairs simultaneously in an end-to-end manner. Specifically, our model regards pair extraction as a link prediction task, and learns to link from emotion clauses to cause clauses, i.e., the links are directional. Emotion extraction and cause extraction are incorporated into the model as auxiliary tasks, which further boost the pair extraction. Experiments are conducted on an ECPE benchmarking dataset. The results show that our proposed model outperforms a range of state-of-the-art approaches.

1 Introduction

Emotion cause extraction (ECE)(Lee, Chen, and Huang 2010) aims to extract possible causes for a given emotion clause. While ECE has attracted an increasing attention due to its theoretical and practical significance, it requires that the emotion signals should be given. In practice emotion annotation is rather labor intensive, limiting the applicability of ECE in practical settings. To address the limitation of ECE, the emotion-cause pair extraction (ECPE) task was recently proposed in(Xia and Ding 2019). Unlike ECE(Lee, Chen, and Huang 2010), ECPE aims to extract emotions and causes without any given emotion signals, and thus better aligns with real-world applications. An illustrative example is given in Figure 1(a), showing that clause 4 serves as the emotion and clauses 2 and 3 are the corresponding causes. Typically, ECPE is formulated as extracting emotion-cause pairs, e.g., (clause 4, clause 2) and (clause 4, clause 3), directly from provided documents.



Document (Clauses)
C1: Yesterday morning,
C2: a policeman visited the old man with the lost money,
C3: and told him that the thief was caught.
C4: The old man was very happy,
C5: and deposited the money in the bank.

Emotion Cause Pair
(C4, C2) (C4, C3)

(a) An example document and extracted emotion-cause pairs

(b) Directional graph of clauses in the document

Figure 1: Extracting emotion-cause pairs from the given document via learning to link.

ECPE is a challenging task, as it requires the extraction of emotions, causes and emotion-cause pairs. Existing work in ECPE mainly focuses on how to collaboratively extract emotions and causes and combine them in an appropriate way. Thus, a two-stage method(Xia and Ding 2019) is typically adopted, which divides pair extraction into two steps: firstly detecting emotions and causes, and then pairing them based on the likelihood of cartesian products between them. Such pipeline method is intuitive and straightforward. However, one critical issue arises, which is the error propagation from the first step to the second.

To tackle the issue, we propose an end-to-end multi-task learning model for predicting emotion-cause pairs, namely **E2EECPE**. Our model takes an innovative perspective by viewing ECPE as a link prediction problem, and connects the emotion/cause extraction and link prediction within one single stage. Particularly, as shown in Figure 1(b), the model predicts whether there exists a directional link from an emotion clause to a cause clause. Meanwhile, we incorporate into the model two auxiliary tasks, namely emotion extraction and cause extraction, which are oriented to further en-

*Equal contribution

hance the expressiveness of the intermediate emotion and cause representations. These are placed in a carefully designed end-to-end multi-task learning architecture, which helps resolving the issue of error propagation.

Extensive experiments are carried out on a benchmarking ECPE dataset. The experimental results demonstrate the effectiveness of the proposed **E2EECPE** model, in comparison with a variety of state-of-the-art baselines. Moreover, a further ablation study indicates that the auxiliary tasks are beneficial. We have also shown that the model can be applied to an extended task: emotion-cause triplet extraction, and achieves a promising pewrformance.

2 Related Work

First of all, our work is related to extracting causes based on emotions explicitly presented in documents, i.e., ECE(Lee, Chen, and Huang 2010). Earlier work views ECE as a word-level sequence tagging problem and tries to solve it with corresponding tagging techniques. Therefore, primary efforts have been made on discovering refined linguistic features(Chen et al. 2010; Lee et al. 2013), yielding improved performance. In line with other tagging related tasks such as named entity recognition (NER), support vector machines (SVMs)(Gui et al. 2014) and conditional random fields (CRFs)(Lafferty, McCallum, and Pereira 2001) have been used for ECE. More recently, instead of concentrating on word-level cause detection, clause-level extraction(Gui et al. 2016) is put forward in that the impact of individual words in a cause can span over the whole sequence in the clause.

With the emergence and development of deep representation learning, neural models has also been utilized in ECE. (Cheng et al. 2017) leverages long short-term memory networks (LSTMs)(Hochreiter and Schmidhuber 1997) to promote the context awareness of clause modelling. (Gui et al. 2017) views the information extraction problem as the retrieval task in question answering (QA) and examines the effect of memory networks(Sukhbaatar et al. 2015) for extraction. Likewise, taking advantage of attention mechanism(Bahdanau, Cho, and Bengio 2014), (Li et al. 2018) employs a co-attention based model and achieves the state-of-the-art performance.

In light of recent advances in multi-task learning, joint extraction of emotions and causes is investigated (Chen et al. 2018) to exploit the mutual information between two correlated tasks. However, these works do not explicitly combine two tasks into one. Thereafter, (Xia and Ding 2019) argues that, while co-extraction of emotions and causes is important, emotion-cause pair extraction (ECPE) is a more challenging problem that is worth putting more emphasis on. Nevertheless, (Xia and Ding 2019) adopts a two-stage approach, which performs emotion and cause extraction first and then pairs the extracted emotions and causes. As discussed in the previous section, such two stage approach suffers from error propagation and high computation cost.

Our work aims to tackle these challenges in ECPE. Rather than processing emotion-cause pair extraction as a two stage task (as used in the existing work), we consolidate two stages into a unified multi-task learning framework, and further consider it as a link prediction task which could be solved in an end-to-end manner.

3 Methodology

To solve ECPE in an end-to-end fashion, we take inspiration from the link prediction problem in graph learning, which aims at predicting potential edges between unconnected vertices in a graph. Essentially, if we consider emotion-cause pairs in a document as triplets in a graph, then the extraction of such pairs is a sort of link prediction from a graph that is at first armed with no edges but only vertices. In order to achieve above procedure, we borrow the idea of learning a graph-based dependency parser(Dozat and Manning 2016) and adapt it to our target task. Coupling link prediction with auxiliary emotion extraction and cause extraction tasks, our model is capable of jointly, and more effectively, extracting emotions, causes, and emotion-cause pairs.

3.1 Problem Formulation

Generally, for any provided document D, we could split it into a sequence of clauses based on punctuation, i.e., $D = \{c_i\}_{i=1}^{|D|}$, where c_i could further be decomposed into words, i.e., $c_i = \{w_j\}_{j=1}^{|c_i|}$. Here, $|D|$ is the number of clauses in the document and $|c_i|$ is the number of words in the i-th clause. ECPE aims to extract a set of $|P|$ emotion-cause pairs $P = \{(c_k^e, c_k^c)\}_{k=1}^{|P|}$ from the document D, where c_k^e, c_k^c represents the emotion clause and the cause clause in the k-th pair, respectively.

3.2 Overall Architecture

An overview of the proposed **E2EECPE** approach is shown in Figure 2. The bottom layer is a clause encoder (Section 3.3) and a document modelling layer (Section 3.4) which transform the word embeddings into the contextualized clause representations. The middle part consists of auxiliary tasks (Section 3.7), i.e., emotion extraction and cause extraction. The top most part is a biaffine attention layer (Section 3.5) which first encodes interaction between the emotion representation and cause representation, and then outputs a postion-weighted pair matrix for pair extraction.

3.3 Embedding & Clause Encoder

With the purpose of integrating words into clause-level neural models, we embed each word in a clause into low-dimensional vectors(Bengio et al. 2003), by which we could represent each word in the clause with its vector representation[1] $c_i = \{\mathbf{w}_j\}_{j=1}^{|c_i|}$, where $\mathbf{w}_j \in \mathbf{R}^{d_e}$ and d_e is the dimensionality of the embedding.

After that, we need to attain contextualized representations of clauses. Owing to the recognized performance and local context awareness of the convolutional neural networks (CNNs) on text classification benchmarks(Kim 2014), we adopt CNNs as the backbone of our clause encoder.

[1] If not specified, we use notations in bold as the vector representations of their original concepts.

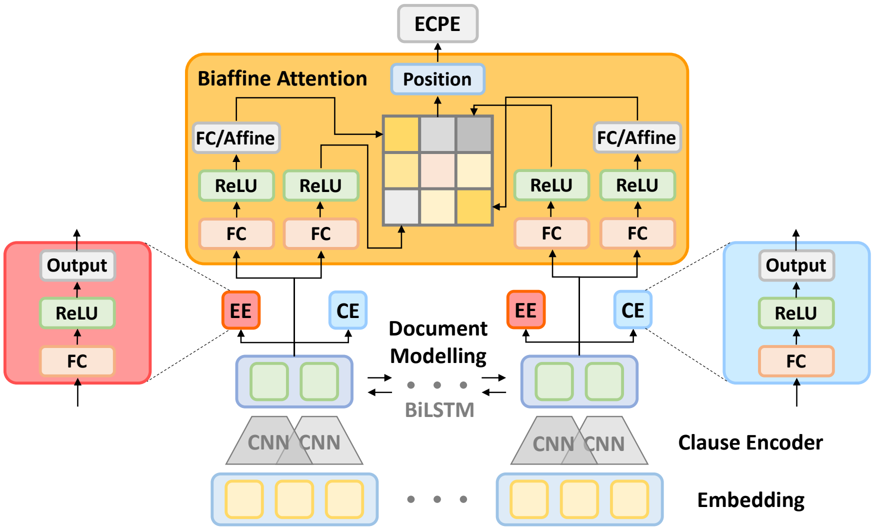


Figure 2: An overview of E2EECPE. EE, CE stands for emotion extraction, cause extraction respectively.

For an embedded clause $c = \{\mathbf{w}_j\}_{j=1}^{|c|}$, we apply one-dimensional convolution operations with kernels of different sizes over the word sequence:

$$\mathbf{C}_t = \sigma(\mathrm{conv}_t(\mathbf{w}_1, \cdots, \mathbf{w}_{|c|})) \tag{1}$$

where conv_t denotes the t-th convolution operation and $\mathbf{C}_t \in \mathbf{R}^{|c| \times d_c}$ is the output of the operation. d_c is the number of filters employed in one convolution operation and $\sigma(\cdot)$ used here is actually $\max(0, \cdot)$.

Then max-pooling is used to distill the features for concatenation. Hence, we finally get context-aware features for the clause:

$$\tilde{\mathbf{c}}_t = \mathrm{maxpool}(\mathbf{C}_t) \tag{2}$$

$$\mathbf{c} = [\oplus_t \tilde{\mathbf{c}}_t] \tag{3}$$

where $\mathbf{c} \in \mathbf{R}^{|t| \cdot d_c}$ is the convoluted feature and $\oplus$ means vector concatenation. $|t|$ is the total number of convolution operations.

3.4 Document Modelling

Since we have sequential clauses in a document, the influences brought by document-level structures become a crucial part that we should fit into our model. A straightforward idea is to leverage temporal relations among clauses with LSTMs.

Specifically, provided with the encoded clause representations $\{\mathbf{c}_i\}_{i=1}^{|c|}$, we employ a bidirectional LSTM to update clause-level features and get $\mathbf{h}_i \in \mathbf{R}^{2d_h}$:

$$\mathbf{h}_i = [\mathrm{LSTM}_f(\mathbf{c}_i) \oplus \mathrm{LSTM}_b(\mathbf{c}_i)] \tag{4}$$

where $\mathrm{LSTM}_f(\cdot)$ and $\mathrm{LSTM}_b(\cdot)$ denote the forward and backward unidirectional LSTMs, respectively. d_h is the dimensionality of hidden states for a unidirectional LSTM.

3.5 Biaffine Attention

Motivated by advances in link prediction(Schlichtkrull et al. 2018), we can directly compute the similarity scores among vertex representations (clause representations in our task), e.g., $\sigma(\mathbf{z}_p^\top \mathbf{z}_q)$ for any representations of vertex p and q, to make predictions. However, the above predictions are only concerned with undirectional circumstances since $\mathbf{z}_p^\top \mathbf{z}_q = \mathbf{z}_q^\top \mathbf{z}_p$, which is not adequate for emotion-cause pair extraction. To solve the problem, we utilize biaffine transform to complete the filling of adjacent matrices, which are called *pair matrices* in our work. This idea is similar to dependency parsing(Dozat and Manning 2016) that is also directional.

Emotion & Cause Representation According to biaffine attention mechanism, each vertex in the graph should have two independent representations, i.e., one is for pointing out and the other for pointed in. In doing so, the pair matrix output by the transformation is asymmetric and direction-aware.

The emotion representation and cause representation are separately offered as below:

$$\mathbf{z}_i^e = \sigma(\mathbf{W}^e \mathbf{h}_i + \mathbf{b}^e) \tag{5}$$

$$\mathbf{z}_i^c = \sigma(\mathbf{W}^c \mathbf{h}_i + \mathbf{b}^c) \tag{6}$$

where $\mathbf{W}^e \in \mathbf{R}^{d_z \times 2d_h}$, $\mathbf{b}^e \in \mathbf{R}^{d_z}$ and $\mathbf{W}^c \in \mathbf{R}^{d_z \times 2d_h}$, $\mathbf{b}^c \in \mathbf{R}^{d_z}$ are two sets of trainable weights and biases, respectively for the emotion and cause representations.

Biaffine Transform Then, we implement biaffine transform on the collected emotion and cause representations. In other words, with the purpose of merging these two kinds of representations into our aimed pair matrices, we fold emotion-cause dynamics into two components. On the one hand, we need to perform a bilinear like operation on each possible pair of emotion and cause. On the other hand, we believe bilinear transform is not enough to deal with such complicated interactions, and thus we facilitate it by injecting bias.

More specifically, we calculate each entry in the expected pair matrix as follows:

$$\mathbf{M}_{p,q} = (\mathbf{W}^m \mathbf{z}^e_p + \mathbf{b}^m)^\top \mathbf{z}^c_q \tag{7}$$

where $\mathbf{W}^m \in \mathbf{R}^{d_z \times d_z}$ and $\mathbf{b}^m \in \mathbf{R}^{d_z}$ are learnable parameters of affine transform, while $\mathbf{M}_{p,q}$ indicates an entry of the pair matrix in the p-th row, q-th column.

Constrained by the inherent property of an adjacent matrix, we further activate the pair matrix with the sigmoid function $g(\cdot)$:

$$\tilde{\mathbf{M}}_{p,q} = g(\mathbf{M}_{i,j}) \tag{8}$$

3.6 Position Weight Matrix

A trivial observation on the co-occurrence patterns of emotions and causes is that the emotion and the cause in a unique pair appear near each other in term of their absolute positions in the document. Thus, position embeddings are introduced to directly encode positions into vectors(Xia and Ding 2019). Different from the existing approaches, our work is based on graph learning, and thereby can not be aided by manipulation of embeddings. Instead, we apply proximity weights on features as in(Zhang, Li, and Song 2019), but extent it to matrices.

Moreover, we notice that in reality people are more likely to inform the causes before expressing emotions. This is verified in Figure 3, where the matrices of ground truth pairs in all documents with document length less than 20 are visualized. In the matrix of Figure 3, the horizontal axis represents the sequence number of the emotion clauses, the vertical axis represents the sequence number of the cause clauses, and each element represents the number of emotion-cause pairs. The diagonal position indicates the number of emotion-cause pairs in the same clauses. The brighter the color, the more pairs there are. It can be seen from the figure that causes tend to appear in the clause before emotions. It implies that we should consider constraining the original matrix representations with the position bias.

Therefore we propose two position weight methods: (1) Asymmetric Position Weight Matrix, and (2) Ground Truth Weight Matrix.

Asymmetric Position Weight Matrix We calculate asymmetric position weight matrix with all training documents:

$$\mathbf{A}^{asw}_{p,q} = \frac{|D| - |p - q - 1| + \epsilon_{asw}}{|D| + \epsilon_{asw}} \tag{9}$$

where ϵ_{asw} is a small number for smoothing.

Ground Truth Weight Matrix We directly use ground truth pairs in training documents to construct the weight matrix:

$$A^{gtw}_{p,q} = \sigma(\sum_{i=1}^{|D|} pairs_i + \epsilon_{gtw}) \tag{10}$$

where ϵ_{gtw} is a small number for smoothing and $pairs_i$ is all ground truth pairs in $document_i$, if there is pair (p, q) in $document_i$, then $pairs_i = 1$.

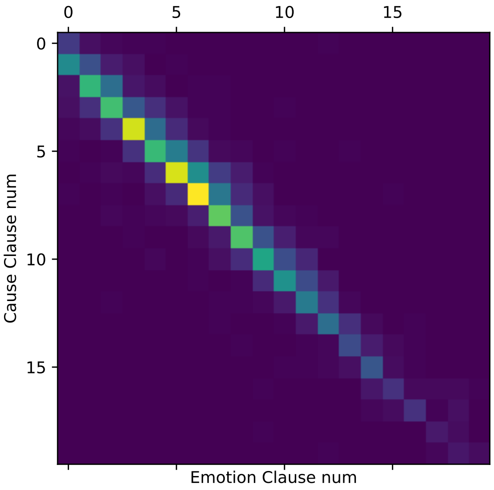


Figure 3: Visualization of values for ground truth pair matrices with document length less than 20 in all documents.

Finally, we obtain the features for indicating pair links as follows:

$$\hat{\mathbf{M}}_{p,q} = \tilde{\mathbf{M}}_{p,q} \odot \mathbf{A}_{p,q} \tag{11}$$

where $\odot$ denotes element-wise multiplication.

3.7 Multi-task Setting

As aforementioned, emotion extraction and cause extraction can be viewed as auxiliary tasks to augment the emotion and cause representations for constructing more expressive pair matrix. Hence we develop a multi-task paradigm, which shares the fundamental part of network structure for the main task with auxiliary tasks. To achieve this goal, we first acquire features dedicated for classification with following procedure:

$$\tilde{\mathbf{z}}^e_i = \sigma(\tilde{\mathbf{W}}^e \mathbf{h}_i + \tilde{\mathbf{b}}^e) \tag{12}$$

$$\tilde{\mathbf{z}}^c_i = \sigma(\tilde{\mathbf{W}}^c \mathbf{h}_i + \tilde{\mathbf{b}}^c) \tag{13}$$

where $\tilde{\mathbf{W}}^e \in \mathbf{R}^{d_z \times 2d_h}$, $\tilde{\mathbf{b}}^e \in \mathbf{R}^{d_z}$ and $\tilde{\mathbf{W}}^c \in \mathbf{R}^{d_z \times 2d_h}$, $\tilde{\mathbf{b}}^c \in \mathbf{R}^{d_z}$ are again two sets of trainable weights and biases, respectively.

Subsequently, predictions are produced by two fully connected layers followed by softmax normalization layers:

$$\hat{\mathbf{y}}^e = \mathrm{softmax}(\hat{\mathbf{W}}^e \tilde{\mathbf{z}}^e_i + \hat{\mathbf{b}}^e) \tag{14}$$

$$\hat{\mathbf{y}}^c = \mathrm{softmax}(\hat{\mathbf{W}}^c \tilde{\mathbf{z}}^c_i + \hat{\mathbf{b}}^c) \tag{15}$$

where $\hat{\mathbf{W}}^e \in \mathbf{R}^{2 \times d_z}$, $\hat{\mathbf{b}}^e \in \mathbf{R}^2$ and $\hat{\mathbf{W}}^c \in \mathbf{R}^{2 \times d_z}$, $\hat{\mathbf{b}}^c \in \mathbf{R}^2$ are weights and biases for learning.

3.8 Training Objective

Eventually, the whole structure can be trained by standard gradient descent. Accordingly, the objective function is a combination of cross entropy with L_2-norm regularization, formulated as below:

$$\begin{aligned}\mathcal{L}_{pair} = &-\sum_{p,q}\mathbf{Y}_{p,q}\log(\hat{\mathbf{M}}_{p,q}) \\ &-\sum_{p,q}(1-\mathbf{Y}_{p,q})\log(1-\hat{\mathbf{M}}_{p,q})\end{aligned} \tag{16}$$

$$\mathcal{L}_{aux} = -\sum_{i}[\sum_{k}\mathbf{y}^{e}_{k}\log(\hat{\mathbf{y}}^{e}_{k}) + \sum_{k}\mathbf{y}^{c}_{k}\log(\hat{\mathbf{y}}^{c}_{k})] \tag{17}$$

where (p, q) and i, k serve as enumerators over all elements. $\mathbf{y}^e$, $\mathbf{y}^c$, and $\mathbf{Y}$ are correspondingly the ground truth.

Furthermore, we add two coefficients to balance the influences of above two objective functions. The ultimate training objective then becomes:

$$\mathcal{L} = \mathcal{L}_{pair} + \beta\mathcal{L}_{aux} + \lambda||\theta||_2 \tag{18}$$

where the term β is used to adjust the potential influences brought by multi-task learning, which is refined according to a pilot study. θ stands for all parameters that need to be optimized, while λ is a coefficient for L_2-norm regularization.

3.9 Inference

With the well trained model, we can infer emotion-cause pairs by comparing each entry in $\hat{\mathbf{M}}$ with a predefined threshold η

$$\hat{\mathbf{Y}}_{p,q} = \begin{cases} 1, & \hat{\mathbf{M}}_{p,q} > \eta \\ 0, & \hat{\mathbf{M}}_{p,q} \leq \eta \end{cases} \tag{19}$$

where $\hat{\mathbf{Y}}$ is the inference result matrix with binary (1-0) indicators.

4 Experimental Setting

4.1 Dataset

We carry out experiments on a publicly available dataset released by(Xia and Ding 2019) for emotion-cause pair extraction, which is the only publically available dataset we can currently access, and most of published papers in the area are experimented based on it. Consisting of news crawled from web, the dataset is referred to as NEWS in the rest of the paper. To facilitate a fair comparison, we use the same data split method as in (Xia and Ding 2019). The dataset is randomly split into ten folds, each of them has a similar amount of data. In our experiments, the evaluation is done in 10 runs where each run uses 9 folds as training data and the remaining one as test data (different for different runs), which respectively take 90% and 10% of the data. Table 1 shows some basic statistics of the dataset. A key observation is that most documents only contain one emotion-cause pair therein, implying the sparsity of the pair matrix. Therefore the issue of label imbalance will be elaborated in following discussions. Moreover, a large amount of emotion-cause pairs have the emotion and the cause within 1 relative offset, suggesting the necessity of using proximity constraints (exactly what position weight matrix does) in the predicted pair matrix.

	NEWS
# of documents	1945
avg. # of clauses per document	14.77
# of EC pairs	2167
# of documents with 1 EC pair	1746
# of documents with over 1 EC pairs	199
# of EC pairs with 0 relative offset	511
# of EC pairs with 1 relative offset	1342
# of EC pairs with 2 relative offset	224
# of EC pairs with over 2 relative offset	90
maximum EC pair offset	12
avg. offset of EC pairs	0.9977

Table 1: Statistics of the dataset. EC stands for emotion-cause, and relative offset indicates the absolute distance between the emotion and the cause of a pair in the document.

4.2 Implementation Details

We use a RTX 2080Ti GPU with 11GB of memory as the experimental hardware base and conduct experiments on the Linux system. The programming language and deep learning framework we use are Python 3.6.3 and Pytorch 1.2.0 respectively.

4.3 Parameter Settings

For all our experiments, pre-trained word vectors on Weibo (a Chinese micro-bloging website) using Word2Vec(Mikolov et al. 2013) are leveraged to initialize the word embeddings. Specifically, the skip-gram used, which is the same as used in the baseline approaches (Xia and Ding 2019). The dimensionality of the embeddings (i.e., d_e) is set to 200. We use 4 convolutional layers (i.e., $|t|$) whose kernel sizes are {2,3,4,5} for the clause encoder and the number of filters for all the convolutional layers (i.e., d_c) is 50, for capturing gram-level features. In order to avoid overfitting, we apply dropout to embeddings and outputs of the clause encoder, yielding 0.5 probability of randomized zeroes on features. The dimensionality for hidden states of a unidirectional LSTM (i.e., d_h) is 300. The dimensionalities for all fully connected layers in the main task and auxiliary tasks (i.e., d_z) are 100. Moreover, the batch size and learning rate are determined through grid parameter search, which are 32 and 10^{-3}, respectively. The number of epoch is 100 with early stop. The coefficient for L_2-norm regularization (i.e., λ) is 10^{-5} . Based on a pilot study, we find the best value for the threshold (i.e., η) in the inference stage is 0.3, which will be detailed in next section. The coefficient for the trade-off in objective function (i.e., β) is 1. In addition, the smoothing term in the calculation of position weight matrix (i.e., ϵ_{asw} and ϵ_{gtw}) are 1 and 0.01, respectively. Furthermore, Adam is used as the optimizer and all trainable parameters are randomly initialized with uniform distribution(He et al. 2015).

Models	emotion extraction			cause extraction			emotion-cause pair extraction		
	P	R	F1	P	R	F1	P	R	F1
Indep w/o classifier	0.8483	0.7961	0.8208	0.6898	0.5648	0.6198	0.5932	0.5094	0.5470
Inter-CE w/o classifier	0.8458	0.8035	0.8263	0.6838	0.5754	0.6231	0.5827	0.5300	0.5531
Inter-EC w/o classifier	0.8406	**0.8097**	0.8242	0.6989	0.5991	0.6426	0.5975	0.5538	0.5716
Indep	0.8483	0.7961	0.8208	0.6898	0.5648	0.6198	**0.6943**	0.5047	0.5833
Inter-CE	0.8458	0.8035	0.8263	0.6838	0.5754	0.6231	0.6780	0.5254	0.5896
Inter-EC	0.8406	**0.8097**	0.8242	0.6989	0.5991	0.6426	0.6691	0.5503	0.6013
E2EECPE-asw	**0.8595**†	0.7915	0.8238	**0.7062**	0.6030	0.6503	0.6478	0.6105†	0.6280†
E2EECPE-gtw	0.8552†	0.8024	**0.8275**	0.7048	**0.6159**	**0.6571**†	0.6491	**0.6195**†	**0.6315**†

Table 2: Comparison results of emotion extraction, cause extraction, and emotion-cause pair extraction with precision, recall, and F1-measure as metrics. The results in bold are the best performing ones under each column. The results of emotion extraction and cause extraction for one-stage and two-stage models are exactly the same because one-stage models are ablated ones of two-stage models. † indicates results that are significantly better than best performing baseline Inter-EC with paired t-test (p is smaller than 0.05).

4.4 Baselines & Evaluation Metrics

Our approach[2] is compared with a range of strong baselines, which are the state-of-the-art methods proposed by(Xia and Ding 2019) for emotion-cause pair extraction. These baselines are either one-stage or two-stage models.

The two-stage models are listed below. They first extract emotions and causes with multi-task architectures independently or interactively, then classify the cartesian products of emotions and causes extracted in the first stage into pairs or non-pairs.

- **Indep** firstly considers emotion extraction and cause extraction as independent tasks and extract emotions and causes with multi-task learning, then pairs the extracted emotions and causes with a classifier.
- **Inter-CE** follows the procedure of **Indep**, however, uses cause extraction to assist emotion extraction in the first stage.
- **Inter-EC** is same as **Inter-CE** except utilizing emotion extraction to improve cause extraction.

We also include three one-stage baseline models. Basically they drop the second stage in the two-stage models, and use the cartesian products as predictions instead of using the classifier in the second stage. The resultant one-stage baseline models are listed as follows.

- **Indep w/o classifier** removes the classifier in the second stage of **Indep**.
- **Inter-CE w/o classifier** removes the classifier in the second stage of **Inter-CE**.
- **Inter-EC w/o classifier** removes the classifier in the second stage of **Inter-EC**.

Precision, recall, and macro F1 measures are adopted as effectiveness metrics in our experiments. The final results are obtained by averaging the ten folds results.

5 Result & Analysis

5.1 Results in Effectiveness

We perform a comparison of **E2EECPE-asw** (E2ECPE with asymmetric position weight matrix) and **E2EECPE-gtw** (E2ECPE with ground truth position weight matrix) with one-stage and two-stage baseline models to quantitatively understand in what ways **E2EECPE** is more effective than the baselines.

Table 2 gives the results in terms of precision, recall and macro-F1 measures. The comparison results demonstrate that our model **E2EECPE-asw** and **E2EECPE-gtw** consistently outperforms the baselines for the main task (emotion-cause pair extraction) with regard to recall and F1, indicating the representation power and the effectiveness of our model. Nevertheless, we also observe that our model performs less well in precision than the two-stage baseline models. With additional observation that the baseline models are performing poorly on recall, we conjecture the existing models suffer from predicting only few testing instances as pairs.

Furthermore, **E2EECPE** is superior on the two auxiliary tasks (emotion extraction and cause extraction). We attribute the improvement to multi-task structure in our model which combines auxiliary tasks and the main task. Apart from that, the one-stage models yield lower results than **E2EECPE** on cause extraction and emotion extraction, suggesting that error propagation is a comparably severe issue in the existing models but is alleviated in our model.

5.2 Ablation Study

To understand the efficacy of auxiliary tasks and position weight matrix, we conduct an ablation study on **E2EECPE-asw** and **E2EECPE-gtw**. Specifically, we separately ablate auxiliary tasks (i.e., emotion extraction and cause extraction) and position weight matrix from **E2EECPE**, and call them **E2EECPE** *w/o aux* and **E2EECPE** *w/o position*, respectively.

The results in Table 3 show a significant performance drop of **E2EECPE** *w/o auxiliary* and a relatively minor

[2] Code is available in https://github.com/shl5133/E2EECPE

Models	P	R	F1
E2EECPE-asw	0.6478	0.6105	0.6280
E2EECPE-gtw	**0.6491**	**0.6195**	**0.6315**
E2EECPE-asw *w/o aux*	0.5982	0.5340	0.5635
E2EECPE-gtw *w/o aux*	0.4562	0.4025	0.4272
E2EECPE *w/o position*	0.6421	0.6158	0.6275

Table 3: Ablation study results. The results in bold are the best performing ones under each column.

drop of **E2EECPE** *w/o position* compared with **E2EECPE-asw** and **E2EECPE-gtw**, verifying the remarkable benefit of the multi-task learning schema. Meanwhile, the results that **E2EECPE** *w/o position* only differs slightly from **E2EECPE** based on all metrics, indicating that imposing position information is still of importance.

5.3 Effect of Threshold in Inference Stage

Inference based on pair matrix is powerful, yet we do not exactly know what threshold (i.e., η) is the most suitable one for its expressiveness. It is therefore helpful to explore the effect of the threshold by altering it and examining the results.

η	E2EECPE-asw / E2EECPE-gtw		
	P	R	F1
0.2	0.5743 / 0.5887	**0.6456 / 0.6372**	0.6071 / 0.6087
0.3	0.6478 / 0.6491	0.6105 / 0.6195	**0.6280 / 0.6315**
0.4	0.6757 / 0.6873	0.5849 / 0.5780	0.6265 / 0.6265
0.5	0.7185 / 0.7062	0.5543 / 0.5640	0.6255 / 0.6260
0.6	**0.7326 / 0.7301**	0.5385 / 0.5396	0.6201 / 0.6201

Table 4: Effect of threshold. The results in bold are the best performing ones under each column.

From Table 4, we conclude that 0.3 is the most appropriate one for our studied task. With increases of η, drops of F1 are noted, implying potential loss of extracted pairs. In addition, we also speculate that the reason why the best value is not around 0.5 (the expectation of random variables ranging uniformly from 0 to 1) is that the element-wise multiplication of a position weight matrix with the sigmoid-activated pair matrix produces a smaller expectation (as upper bound decreases).

5.4 Issue of Label Imbalance

In order to measure the impact brought by label imbalance, typically in the form of pair matrix sparsity, we remove the examples containing more than one pair for test set in each fold to make up a `Hard` dataset, then record the mean results across ten folds correspondingly.

We can observe in Table 5 that our model encounters a failure on the `Hard` dataset with decreases on precision and F1 measure, suggesting that further investigation is needed to solve this problem.

Type	E2EECPE-asw / E2EECPE-gtw		
	P	R	F1
`Full`	0.6478 / 0.6491	0.6105 / 0.6195	0.6280 / 0.6315
`Hard`	0.6002 / 0.6322	0.6479 / 0.6078	0.6226 / 0.6186

Table 5: The results for verifying the issue of label imbalance.

5.5 An Extended Task: Emotion-Cause Triplet Extraction

Our model can also be extended to emotion-cause triplet extraction setting with minor modifications. Specifically, emotion-cause triplet differs from emotion-cause pair only on that a triplet should contain types of emotions while a pair does not. The emotion type extraction is also very important, just like in the *Sentiment Analysis* task. So we adjust the output of the Biaffine Transform from a two-dimensional tensor to a three-dimensional tensor. The extra dimension represents the emotion type.

The experimental results are shown in Table 6. It can be seen that using the E2EECPE model for the triplet extraction task can also achieve promising results.

	P	R	F1
triplet extraction	0.5824	0.4304	0.4940
emotion extraction	0.8417	0.7875	0.8131
cause extraction	0.6577	0.5770	0.6133

Table 6: Results of triplet extraction, emotion extraction, and cause extraction with precision, recall, and F1-measure as metrics.

6 Conclusions and Future Work

The emotion-cause pair extraction task is a new and more realistic task that seeks to identify emotion-cause pairs in documents. However, previous models are inherently limited by the idea of solving the task via two stages. To this end, we propose an end-to-end multi-task learning model that regards the problem as predicting directional links between emotions and causes via biaffine attention. Additionally, we also aid the model with auxiliary tasks and position weight matrix. Experimental results prove the superiority of our model over a series of baselines.

We believe there are some promising directions yet to be explored. Firstly, the position weigh matrix used in our model is directly obtained from document information. Further models such as graph neural networks are expected to be developed to incorporate with learned position information instead of refined one. Secondly, triplet extraction is a brand new direction and is worth researching. Thirdly, the label imbalance issue should be addressed with task-specific tactics.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* .

Bengio, Y.; Ducharme, R.; Vincent, P.; and Jauvin, C. 2003. A neural probabilistic language model. *Journal of machine learning research* 3(Feb): 1137–1155.

Chen, Y.; Hou, W.; Cheng, X.; and Li, S. 2018. Joint learning for emotion classification and emotion cause detection. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 646–651.

Chen, Y.; Lee, S. Y. M.; Li, S.; and Huang, C.-R. 2010. Emotion cause detection with linguistic constructions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 179–187. Association for Computational Linguistics.

Cheng, X.; Chen, Y.; Cheng, B.; Li, S.; and Zhou, G. 2017. An emotion cause corpus for chinese microblogs with multiple-user structures. *ACM Transactions on Asian and Low-Resource Language Information Processing (TALLIP)* 17(1): 6.

Dozat, T.; and Manning, C. D. 2016. Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv:1611.01734* .

Gui, L.; Hu, J.; He, Y.; Xu, R.; Qin, L.; and Du, J. 2017. A Question Answering Approach for Emotion Cause Extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1593–1602.

Gui, L.; Wu, D.; Xu, R.; Lu, Q.; and Zhou, Y. 2016. Event-Driven Emotion Cause Extraction with Corpus Construction. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 1639–1649.

Gui, L.; Yuan, L.; Xu, R.; Liu, B.; Lu, Q.; and Zhou, Y. 2014. Emotion cause detection with linguistic construction in chinese weibo text. In *Natural Language Processing and Chinese Computing*, 457–464. Springer.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, 1026–1034.

Hochreiter, S.; and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8): 1735–1780.

Kim, Y. 2014. Convolutional Neural Networks for Sentence Classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1746–1751.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289. Morgan Kaufmann Publishers Inc.

Lee, S. Y. M.; Chen, Y.; and Huang, C.-R. 2010. A text-driven rule-based system for emotion cause detection. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, 45–53. Association for Computational Linguistics.

Lee, S. Y. M.; Chen, Y.; Huang, C.-R.; and Li, S. 2013. DETECTING EMOTION CAUSES WITH A LINGUISTIC RULE-BASED APPROACH 1. *Computational Intelligence* 29(3): 390–416.

Li, X.; Song, K.; Feng, S.; Wang, D.; and Zhang, Y. 2018. A co-attention neural network model for emotion cause analysis with emotional context awareness. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 4752–4757.

Mikolov, T.; Chen, K.; Corrado, G.; and Dean, J. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781* .

Schlichtkrull, M.; Kipf, T. N.; Bloem, P.; Van Den Berg, R.; Titov, I.; and Welling, M. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, 593–607. Springer.

Sukhbaatar, S.; Weston, J.; Fergus, R.; et al. 2015. End-to-end memory networks. In *Advances in neural information processing systems*, 2440–2448.

Xia, R.; and Ding, Z. 2019. Emotion-Cause Pair Extraction: A New Task to Emotion Analysis in Texts. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 1003–1012. Florence, Italy: Association for Computational Linguistics. doi:10.18653/v1/P19-1096. URL https://www.aclweb.org/anthology/P19-1096.

Zhang, C.; Li, Q.; and Song, D. 2019. Syntax-Aware Aspect-Level Sentiment Classification with Proximity-Weighted Convolution Network. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1145–1148.