Multi-Span QA Approach to Extracting Emotion Cause Pairs

Zhenyu Lei *
zhenyulei@umass.edu

Yachan Liu
yachanliu@umass.edu

Yuanming Tao
yuanmingtao@umass.edu

Xiaocheng Zhang
xiaochengzha@umass.edu

1 Problem Statement

Exploring the underlying causes of emotions can have a wide range of practical uses, including but not limited to the analysis of customer feedback and tracking shifts in public opinions. The traditional emotion cause extracion (ECE) task (Gui et al., 2016) aims to judge if each clause in the document is the corresponding cause, provided the annotation of emotions in advance. The limitations of this task are that it needs a large number of emotion annotation, and the requirement that emotion annotation first and cause extraction last ignores inner relationship between emotions and causes.

To improve this task, a new interesting task, called emotion-cause pair extraction (ECPE) first proposed by Xia and Ding (2019), has emerged in the area of text emotion analysis. It aims at extracting the potential pairs of emotions and their corresponding causes in a document. Figure 1 provides an example of this task.

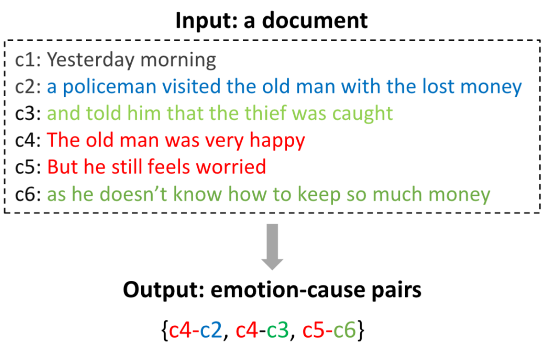

Figure 1: An example showing the emotion-cause pair extraction (ECPE) task. Figure adapted from (Ding et al., 2020b)

The mathematical formulation of the problem is as follows: Let $D = \left(c_1, c_2, \ldots, c_{|D|}\right)$ be a document, where $|D|$ represents the number of sentences. Each clause $c_i = \left(w_1^i, w_2^i, \ldots, w_{|c_i|}^i\right)$ in the document is a sequence of words. The goal is to extract all emotion-cause pairs in D, denoted by $\mathcal{P}$, given by:

$$\mathcal{P} = \left\{\left(c^{\mathrm{emo}_1}, c^{\mathrm{cau}_1}\right), \left(c^{\mathrm{emo}_2}, c^{\mathrm{cau}_2}\right), \ldots\right\} \quad (1)$$

Here, $\left(c^{\mathrm{emo_j}}, c^{\mathrm{cau_j}}\right)$ represents the j-th pair, where $c^{\mathrm{emo_j}} \in D$ is an emotion clause, and $c^{\mathrm{cau_j}} \in D$ is the corresponding cause clause. It is important to note that an emotion can have more than one cause, and the same cause can be the stimulus for multiple emotions. Additionally, the number of ground-truth emotion-cause pairs varies for different documents.

Although recent methods have achieved impressive progress, most of the methods are based on complex neural architectures that model the interrelationship between emotion clauses and cause clauses (Ding et al., 2020a,b; Wei et al., 2020; Wu et al., 2020). This creates two problems: a label sparsity issue due to the need to generate a pairing matrix by enumerating all possible combinations of clauses and then selecting valid emotion-cause pairs (Wei et al., 2020), or an unrealistic tagging scheme that attempts to model the relationship between different clause types within a predefined distance threshold (Ding et al., 2020b). Additionally, in practice, creating a corpus with annotated emotion-cause pairs is time-consuming. Therefore, we came up with the idea of extracting emotion-cause pairs from unannotated corpora.

In summary, the objective of this project is to address the ECPE task (Xia and Ding, 2019) using various approaches. Initially, we put forward our own approach to tackle this problem. Additionally, we aim to investigate the potential of Large Language Models (LLMs) in solving this task

*The authors are listed in an alphabetical order of surnames.

by employing effective prompt engineering techniques. Moreover, we aim to reframe the task as a multi-span extractive Question-Answering (QA) problem (Segal et al., 2020).

2 What you proposed vs. what you accomplished

We have outlined three tasks in our project proposal and now summarize our progress as follows:

Task 1:

- ~~We intended to rewrite the codebase of the ECPE-MLL method (Ding et al., 2020b), which was originally implemented using TensorFlow 1.x and Python 2, using PyTorch.~~ However, since the existing codebase cannot be executed, we have decided to develop our own method from scratch to address this challenge.

Task 2:

- ~~We planned to evaluate the performance of the LLM on this task.~~
- ~~We aimed to conduct zero, one, and few-shot experiments on the GPT-3.5 model and evaluate its performance.~~
- ~~We intended to test the impact of two types of prompts, namely English and Chinese versions.~~
- ~~We planned to perform error analysis to identify the emotion-cause pairs that GPT-3.5 fails to extract.~~

Task 3:

- ~~Reconceptualize the ECPE problem as an extractive QA task.~~

3 Related work

Pretrain and Fine-tuning Paradigm The existing methods to solve the ECPE task could be divided into two-stage framework (Xia and Ding, 2019) and end-to-end framework (Ding et al., 2020a,b; Wei et al., 2020; Wu et al., 2020). Both frameworks fall into the category of fine-tuning paradigm.

As an example for the two-stage framework, (Xia and Ding, 2019) first extracts emotion clauses and cause clauses separately, then obtain candidate emotion-cause pairs by the Cartesian product, and trains a classifier to filter out invalid pairs. The shortcomings are that the emotion-cause pairs are not extracted directly by the model, and the first step is prone to propagate error to the second step.

In terms of end-to-end techniques, as illustrated in Figure 2, they initially extract features at the clause level from pre-trained embeddings or language models such as BERT (Devlin et al., 2019). Then, a contextual encoder is created to generate contextual representations of clauses. Afterwards, these representations are combined with position information of the clause. Finally, the output is utilized for predicting the task. Ding et al. (2020a) extracts emotion-cause pairs through in forms of a 2D square matrix. Fan et al. (2020) proposed a transition-based framework to transform the emotion-cause pairs extraction into a task of parsing-like directed graph construction. Wei et al. (2020) tackled the ECPE task from a ranking perspective, modeled the inter-clause relationships with graph attention and kernel-based relative position embedding. Ding et al. (2020b) proposed a joint framework to solve the ECPE task by sliding window multi-label learning. However, these end-to-end approaches have two clear drawbacks. Firstly, the current methods learn the relationships among various task objectives implicitly, instead of explicitly modeling their connections. Secondly, including position information could make the model sensitive to data distribution and reduce its robustness.

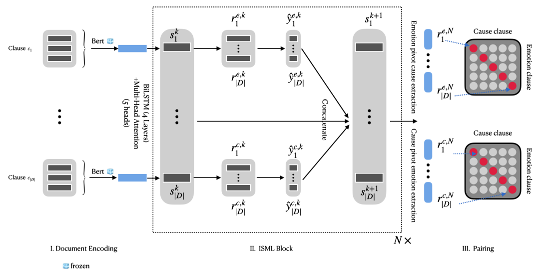

Figure 2: Illustration of the fine-tuning approach to the ECPE task. Figure adapted from (Zheng et al., 2022)

Prompting LLMs Paradigm Recently, LLMs such as ChatGPT, GPT-4 and BingChat are increasingly being used to analyze text and answer questions. Along with the explosive development of LLMs, prompting engineering is also a research area that has gained significant attention. Although it is already proved to be strong in understanding in-context, we still do not know whether it works well for the human-subjective emotions and sentiment analysis. In a paper released in April 2023, Wang et al. (2023a) evaluated ChatGPT on five typical emotion-analysis tasks in four cases including standard evaluation, polarity shift evaluation, open-domain evaluation, and sentiment inference evaluation. They found that ChatGPT is not as good as fine-tuned BERT and SOTA models in the most cases. The performance could be improved significantly by few-shot prompting, and proper prompts is not trivial.

Extractive QA Extractive QA refers to the task of identifying a span of text within a given context that provides an answer to a given question. In the single-span extractive QA, the output/answer is constrained to be a single contiguous span from the input. Figure 3 illustrates how BERT (Devlin et al., 2018) is utilized in the single-span extractive QA task. However, single-span models (Seo et al., 2018; Yu et al., 2018; Hu et al., 2018) cannot work for questions whose answer is a set of non-contiguous span from the input. To deal with these questions, multi-span extractive QA (Li et al., 2022), a type of QA task that involves identifying multiple relevant spans of text from a given passage to answer a question, is introduced. In recent years, multi-span approaches (Hu et al., 2019; Chen et al., 2020; Segal et al., 2020; Cui et al., 2021) have been gaining attention due to its ability to provide more complete and accurate answers compared to single-span approaches. On the other hand, shortcomings such as the limited span range and complex training procedure may still existing. As an example of multi-span method, a fully-differentiable framework is introduced by Segal et al. (2020), where QA is casted as a sequence tagging task. Each token is predicted whether it's included in the answer.

4 Experimental Setup

4.1 Summary Statistics of Dataset

We conducted the experiments on the ECPE corpus dataset[1] (Xia and Ding, 2019). This dataset was constructed based on a public Chinese emotion corpus from the SINA NEWS website [2]. The statistics of the dataset is summarized in Table ??.

[1]Available at: https://github.com/NUSTM/ECPE-MLL/tree/master/data

[2]https://news.sina.com.cn/

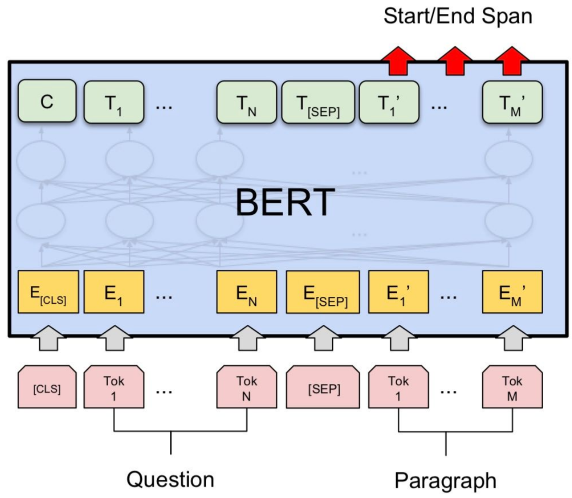

Figure 3: A BERT-based approach to solving the single-span extractive QA task. Figure adapted from (Devlin et al., 2018).

# of documents	1945
Avg. # of clauses per document	14.77
Max. # of clauses per document	73
# of documents with 1 EC pair	1759
# of documents with 2 EC pairs	164
# of documents with 3 EC pairs	21
# of documents with 4 EC pairs	21
# of EC pairs	2154
# of EC pairs with 0 relative offset	508
# of EC pairs with 1 relative offset	1333
# of EC pairs with 2 relative offset	224
# of EC pairs with over 2 relative offset	89
Max. EC pair offset	12
Avg. offset of EC pairs	0.9981

Table 1: Summary Statistics of the dataset. EC stands for absolute distance between emotion-cause pair.

Overall, the dataset contains 1,945 documents and 28,727 clauses. Among them, there are 1,746 documents with one emotion-cause pair, 177 documents with two emotion-cause pairs, and 22 documents with more than two emotion-cause pairs. In theoretical terms, if the average number of clauses per document is denoted as N, then the potential number of candidate pairs for a document would be N^2. However, as indicated in Table 1, out of 1945 documents, 1746 of them contain a single pair. This highlights the issue of label sparsity, where the majority of documents have limited labeled pairs.

Table 2 shows the distribution of emotion type. Note that some pairs can have more than one kind of emotions. This could pose another challendge for the model to learn the relationships between emotion clauses and cause clauses. More than 99 % of pairs has only one emotion.

Emotion	Number
Emotion with only one type	
Sadness	567
Happiness	549
Fear	402
Anger	283
Surprise	85
Emotion with two type	
Happiness & Fear	1
Anger & Sadness	1
Sadness & Disgust	1
Disgust & Anger	1
Happiness & Sadness	1
Disgust & Fear	1

Table 2: Distribution of emotion types

Due to causality and human habit, it's empirical and intuitive that the distance between cause clause position and emotion clause position is always short in a document. We also demonstrate this hypothesis. As shown in the Table 1 and Figure 4, the relative offset indicates the distance (number of clauses) between two clauses in the document. There are about 24 % and 62 % of all emotion-cause pairs have relative offset 0 or 1, respectively. In total, more than 96 % of all emotion-cause pairs have a quite small relative offset, that is, no greater than 2. This prior information can be useful in two ways: 1) it can help us to filter out the distant pairs in the documents when we design our own neural network, 2) it can potentially improve our prompting strategy by providing such information in the prompts.

4.2 Evaluation Metrics

In our experiments, we applies the same setting as the work (Xia and Ding, 2019). We use the same data split, specifically allocating 80% for the training set, 10% for the test set, and another 10% for the validation set. In terms of the evaluation metrics, the precision, recall and F1 score in Xia and Ding (2019) are defined as follows.

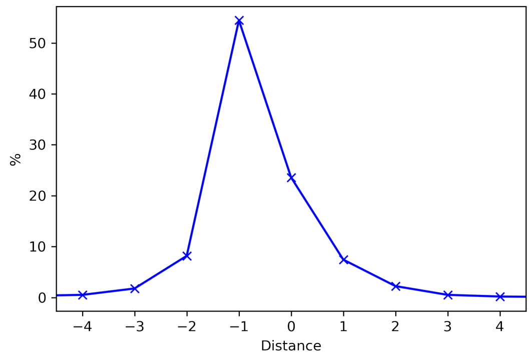

Figure 4: Distribution of distances between emotions and causes in our dataset.(Xia and Ding, 2019)

$$Precision = \frac{\text{\# of correctly predicted pairs}}{\text{\# of predicted pairs}}$$

$$Recall = \frac{\text{\# of correctly predicted pairs}}{\text{\# of ground-truth pairs}}$$

$$F_1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

5 Baselines

Upon discovering this ECPE task that aligned with our interests, we promptly conducted performance tests on ChatGPT. To our surprise, we observed unsatisfactory results after several attempts. Subsequently, we randomly selected 100 samples from the corpus and used simple prompts to test the performance of the GPT-3.5 model in a zero-shot setting. The resulting precision, recall, and F1 scores were 0.0552, 0.1797, and 0.0845, respectively. Based on these scores, we decided that the GPT-3.5 model would serve as our baseline for further experimentation.

6 Proposed approach

6.1 Task 1

6.1.1 Architecture

As we have discussed before, at the beginning of the study, we planned to re-implement (Ding et al., 2020b). However, we found that they use the out-of-date Python 2.7 and TensorFlow 1.0, which is uninterpretable and confusing. Besides, we cannot even run their code based on their guideline on GitHub. Therefore, it is impossible for us to build our system on top of their system and only modify some parameters or features. Instead, we turn to re-write the whole system using Python 3 and Pytorch by ourselves, where we designed a neural architecture different from the paper (Ding et al., 2020b). Specifically, we refer to the structure of theirs, as well as add some features to improve its ability to learn.

Our model is as Figure 5 shows. We first use a BERT tokenizer to get the vectorized tokens of those Chinese words and then feed them into a pre-trained BERT model. Both the tokenizer and the pre-trained model are BERT-Base-Chinese. The output of BERT is fed into a Bi-LSTM layer, followed by a multi-head attention module. After that, Iterative Synchronized Multitask Learning (ISML) model is applied to learn the embedding of emotion-cause pairs, as well as predict whether the clause is emotion/cause or not. As Figure 5 shows, the ISML model contains several cascaded ISML blocks. In each block, we duplicate the input (s^k) into two symmetric branches: the emotion branch and the cause branch. Take the emotion branch as an example, we first apply a Bi-LSTM layer to learn the emotion-specific representation $r_i^{e,k}$, then we use a linear layer with softmax to predict the probability of each clause to be an emotion clause ($\hat{y}^{e,k}$). Finally, we concatenate the input of the current ISML block k (s^k) and the output of both emotion branch ($\hat{y}^{e,k}$) and cause branch ($\hat{y}^{c,k}$) along the final dimension ($[s^k, \hat{y}^{e,k}, \hat{y}^{c,k}]$). The output will be the input of next ISML block $k+1$.

After N ISML blocks in total, we get the final representation of emotion ($r_i^{e,N}$) and cause ($r_i^{c,N}$). In order to predict the pairs of emotions and causes, we first utilize a linear layer to map $r_i^{e,N}$ into a $D \times D$ Emotion-pivot cause extraction (D is the max number of clauses in a paragraph). This extraction shows the potential relationship of a cause clause to the current emotion clause. According to previous works (Wei et al., 2020), most emotion-cause pairs are in a distance of $\leq$ 3 clauses. Therefore, in order to reduce the searching space, we apply a sliding window mask on top of the scores to only focus on the adjacent clauses of the current clause. In other words, for emotion clause c_i, we only consider $2|W|+1$ candidate cause clauses: $c_{i-|w|}, \ldots, c_{i-1}, c_i, c_{i-1}, \ldots, c_{i+|w|}$.

Finally, we turn these two scores into probabilities using the following calculation, aligned with (Ding et al., 2020b):

$$
\begin{aligned}
p(y_i^{cml_j} = 1|c_i) &= \hat{y_i}^{cml_j} \\
&= \frac{1}{1 + \exp^{\mathbf{W}^{cml_j}\mathbf{r}_i^{e,N} + \mathbf{b}^{cml_j}}} \\
p(y_i^{cml_j} = 0|c_i) &= 1 - \hat{y_i}^{cml_j}
\end{aligned}
$$

We also use the same method to get the Cause-pivot emotion extraction probabilities for the cause branch. In the end, we use logic OR strategy to make the decision of prediction. Specifically, clause p and clause q are predicted as an emotion-cause pair if and only if $\hat{y_i}^{cml_j} > threshold$ and $\hat{y_i}^{eml_j} > threshold$. The threshold tuning will be discussed later.

As for the loss function, we use the same loss component as (Ding et al., 2020b): L^{ISML-N},L^{CMLL}, and L^{EMLL}, where:

$$
\begin{aligned}
L^{ISML-N} = &- \sum_{t=1}^{N}(\sum_{i=1}^{|d|}(\mathbf{y}_i^e \times \log(\hat{\mathbf{y}}_i^{e,t})) \\
&+ \sum_{i=1}^{|d|}(\mathbf{y}_i^c \times \log(\hat{\mathbf{y}}_i^{c,t}))) \\
L^{CMLL} = &- \sum_{i=1}^{|d|}\sum_{j=-|w|}^{|w|}(y_i^{cml_j} \times \log \hat{y}_i^{cml_j} \\
&+ (1 - y_i^{cml_j}) \times \log(1 - \hat{y}_i^{cml_j})) \\
L^{EMLL} = &- \sum_{i=1}^{|d|}\sum_{j=-|w|}^{|w|}(y_i^{eml_j} \times \log \hat{y}_i^{eml_j} \\
&+ (1 - y_i^{eml_j}) \times \log(1 - \hat{y}_i^{eml_j}))
\end{aligned}
$$

6.1.2 Issue of the referred paper and our improvement

We've demonstrated the architecture of our model in the last section. Here, we will focus on the difference between our model and the model in (Ding et al., 2020b) and the reasons why we take these steps.

In the beginning, we actually used the exact same model as (Ding et al., 2020b). However, during our preliminary study for training, we found that this model is very likely to underfit: the learning of the model converges too fast no matter how we tune the learning rate and other parameters, while the precision is much lower (around 1%) than their reported values. As a result, we decide to increase its ability to represent the features and learn the relationship between clauses:

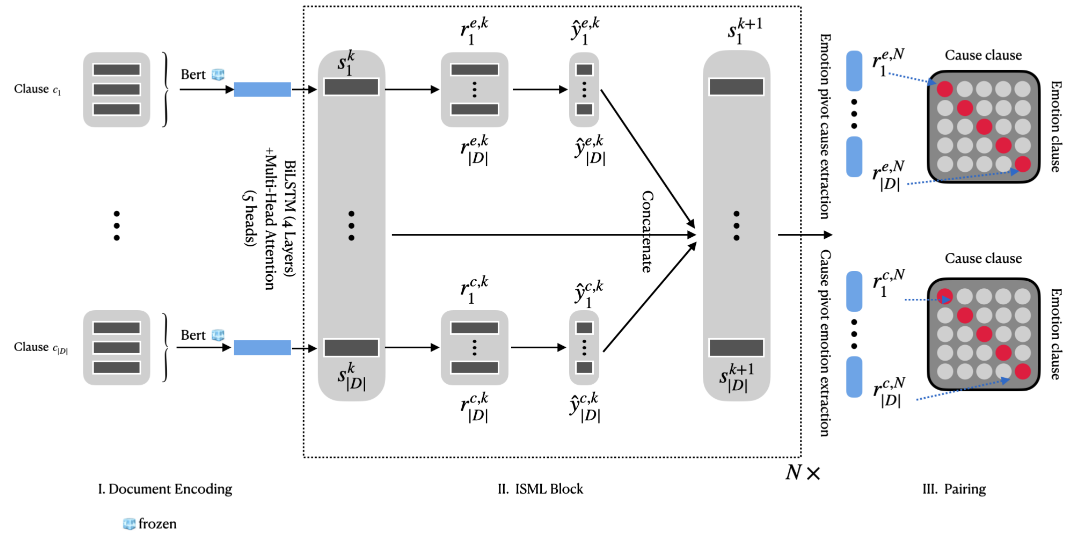

Figure 5: Overview of our proposed neural architecture to solve the emotion cause pair extraction task.

1. **Use pretrained BERT-Base-Chinese tokenizer and model to encode the input texts instead of Word2Vec only:** The emotion-cause pairing task is difficult. The origin model only uses Word2Vec + BiLSTM + Linear, which we think is not enough to represent the embedding relationship among the clauses. A large encoder could help with this problem. However, it could cost too much time and money for us to train one from scratch by ourselves. As a result, a pre-trained model can be a good solution, since at the beginning stage of the model, we just need to learn some latent correlations and it does not need to be very task-related. During training time, this BERT encoding layers are frozen.

2. **Use multi-head transformer instead of self-attention:** Single-head single-layer self-attention is of low capability. Besides, this is a multi-label task, which means the same clause could have several different labels. Therefore, a multi-head transformer makes more sense.

3. **Increase the number of layers in Bi-LSTM:** To increase the capability of Bi-LSTM.

4. **Add activation layer:** In the origin paper, there's no activation at all, even for linear layers after the output of Bi-LSTM. However, without activation, the capability of the model could be downgraded, since adjacent linear layers could couple together and work as a single layer.

After applying the above methods to improve the capability of the model. We realize another as well as the biggest problem: the dataset and labels are seriously imbalanced. In fact, take one paragraph of D clauses as an example, we can have $\frac{D \times D}{2}$ possible emotion-cause pairs as a full permutation of all clause pairs, where only one or two of them are true positives, as Figure 6 shows. In other words, the ratio of false labels (not e-c pairs) to true labels (e-c pairs) could be of the magnitude of D^2. Note that the maximum D we set is 50, so this ratio could be more than a thousand.

Due to such an imbalance, it is very hard for us to train the network: if using normal training strategies, the model will predict every pair as false (i.e. not an e-c pair). The reason is the number of 0 (false) is too much more than 1 (true), so the loss contributed by false positive ($0 \rightarrow 1$) is much larger than false negative ($1 \rightarrow 0$). During optimization, the model tends to increase the probability of 0, because even if it predicts 1 as 0, the loss punishment is negligible.

In order to solve this imbalance issue, (Ding et al., 2020b) proposes a sliding window mechanism to reduce the output searching space. We also utilize this in our implementation. However, only using a sliding window is far from enough, because although we reduce the number of unrelated label 0s, its amount is still far more than that of label 1. And the authors of that paper have no solution to the remaining imbalance problem.

In that case, we strongly doubt the results provided by (Ding et al., 2020b). There are 3 main reasons: **(1)** They do not deal with the imbalanced data in an effective way as ours, which will be discussed below. **(2)** They use simple and weak structures (Word2Vec + Bi-LSTM) but achieve very high performance (70%), which even much outperforms GPT 3.5 (our baseline as well). This is hard to be persuasive. **(3)** We found issues and wrong implementation in their code, which is different from their paper. This makes their correctness a doubt.

As a result, we decide not to re-implement their paper totally and not to use their results as a baseline. We use GPT 3.5 as a baseline instead, as mentioned in section 5.

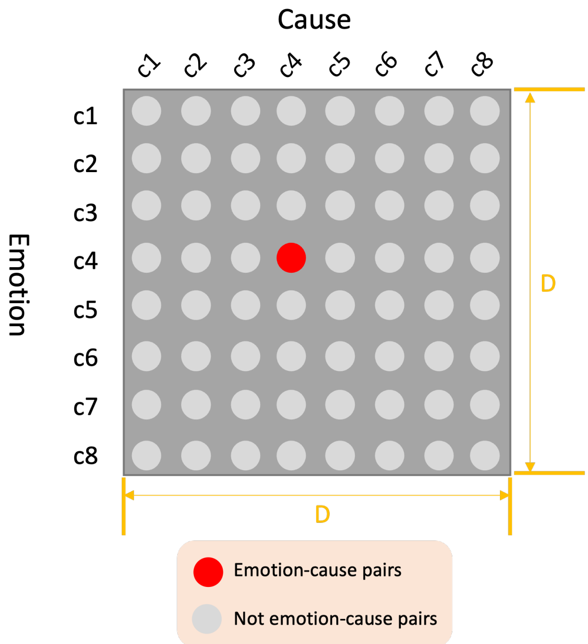

Figure 6: The demonstration of inherently imbalanced data labels

Therefore, in order to solve the imbalance issue, we need extra efforts and tricks.

6.1.3 Solving the issue of imbalanced data and labels

The imbalance of the dataset is unpreventable: the emotion-cause pairs are inherently much less than other pairs. As a result, vanilla data augmentation methods may not work. Besides, contrastive learning usually requires heavy data augmentation and a very large batch size. As a result, we can not apply such strategies easily, and that's beyond the scope of our report.

In contrast, we apply several simpler but as effective tricks as below:

1. **Cost-sensitive training:** As we discussed in the above section, the normal loss functions (such as cross-entropy) are contributed more by false positive ($0 \to 1$) than false negative ($1 \to 0$) since there are much more 0 than 1. In order to balance their influence on the result, we use a biased loss function. Specifically, we tune the weight of loss of false negative ($1 \to 0$) to be larger than that of false positive ($0 \to 1$). In other words, we have $loss_{bias} = w \times Loss_{CE}(FN) + 1 \times Loss_{CE}(FP)$, where w controls the weight amplification to the original cross-entropy loss of false negative. We found that this method prevents the model from only outputting 0 effectively.

2. **Imbalanced decision threshold**: Only tuning the loss function is still not enough. It's very hard to control. If w is too small, the model will predict too many 0; while if w is too large, the model will predict too many 1 since now the punishment on FN is larger. Therefore, we also need to introduce other variables such as changing the threshold we use for decisions. In a normal balanced dataset, the criteria should be $\hat{y_i}^{cml_j} > 0.5$ and $\hat{y_i}^{eml_j} > 0.5$, since this will lead to the same threshold for both 0 and 1. In contrast, in our more imbalanced dataset, we tune the threshold also in an imbalanced way to a value greater than 0.5. This will lead to more careful prediction of label 1.

3. **dynamic loss function:** We believe that the order and frequency that the model deals with different loss are very important. In other words, optimizing the loss on judging whether a clause is emotion/cause (L^{ISML-N}) and the loss on predicting the emotion-cause pairs (L^{CMLL},L^{EMLL}) is different and don't have to take place simultaneously. Therefore, we can first let the model learn the emo/cause clause finding and then

pair finding, alternatively. We implement this but do not actually use it for our final results.

6.1.4 Nonuniform input length

Since the number of clauses varies for different documents, the input length of our model is different. Consequently, we have to use padding first to make the size suitable. Besides, we also filter the output to the range of the length of the paragraph, as well as masking the intermediate outputs between different components of the model to predict reasonable output and stop the unwanted gradients from flowing backward.

6.1.5 Computing resource

We utilize Google Cloud Compute services for model training. The hardware consists of an Intel Skylake CPU (13GB) platform along with a single NVIDIA T4 GPU (with 16GB of memory). In order to execute our code successfully, it is necessary for the GPU to possess a memory capacity exceeding 8GB.

6.2 Task 2

In this task, we explored the ability of Large Language Models (LLMs) to solve this ECPE problem through proper prompt engineering techniques. We attempted various prompt formats and methods to improve precision, recall, and F1 scores. The main methods employed were instruction prompt and few-shot learning. In the few-shot learning approach, we used a maximum of 5 examples and their corresponding answers as the training data. Including more examples would exceed the token limit due to increased information length.

For the instruction prompt, we adopted a Chain-of-Thought format similar to the one described in the paper (Wang et al., 2023b). This particular instruction prompt yielded the highest score: precision 0.1509. The specific instructions included tasks such as "1. Describe in one sentence the emotion contained in the given document and its corresponding reason. 2. Output the identifier of the emotion clause from Task 1, considering only the one with the strongest intensity," and so on.

Due to time constraints, we also intended to explore the combination of few-shot learning with the aforementioned instruction prompt to achieve more complex prompts. However, the combination of complex prompts and few-shot learning examples posed limitations in terms of the maximum token count and the need for annotation. Task-specific step-by-step instructions consumed a significant number of tokens, and adding few-shot examples and their corresponding answers easily exceeded the token limit. Moreover, annotating the few-shot examples required additional resources and time, which we were unable to allocate before writing the paper.

In our code, the file `task2/n_shot/ecpetest.ipynb` contains zero-shot, one-shot, and five-shot experiments with Chinese data and English instruction prompts. The file `task2/n_shot/ecpetest2.ipynb` includes zero-shot experiments with Chinese data and instruction prompts incorporating "pseudo Chain-of-Thought (CoT)". The file `task2/zeroshot_example/response.csv` contains GPT-3.5's responses to all documents. It is worth noting that not all answers are the pairs of clause numbers we want. Bad answers contain unexpected words or symbols. After removing these bad responses, responses to 530 documents (about a quarter of all documents) remained. The main difference between our approach and the baselines lies in the performance of few-shot learning. As we utilized OpenAI's GPT3.5 `text-davinci-003` model, there was no need to run any models locally for this task. The execution process is straightforward, requiring only the provision of an API key. However, it comes at a considerable cost.

7 Results

7.1 Task 1

7.1.1 Implementation Details

After heuristic searching of hyperparameters, we achieve a result of precision of 0.2475, recall of 0.7329, and F1 score of 0.3700.

We have a total of 1945 documents, which are split into training, validation, and test sets with a ratio of 8:1:1. For all sets, we split each of them further into 20 pieces for flexibility in training and testing. To train our model, we utilize the Adam optimizer with a learning rate of 0.0003 and a batch size of 36 for 30 epochs. We use linear warmup with a warmup proportion of 0.1. For the biased loss, we use a weight of 30. We use an imbalanced decision threshold of 0.75 (to predict "is" an emotion-cause pair). We use 16 ISML blocks. Additionally, we set the l_2 regularization coefficient to 1e-5. The hidden size of all layers is 100. The selection of the best model is based on the highest F1 scores on the validation set, and we evaluate the model's performance on the test set.

7.1.2 Ablation Studies

We conduct ablation studies to analyze the effects of different components in our proposed neural architectures.

The initial ablation study aimed to compare the performance of self-attention and multi-head attention. Table 3 presents the results, demonstrating that multi-head attention significantly outperforms self-attention across various evaluation metrics.

Modules	Precision	Recall	F1
Self-Attention	0.0632	0.3572	0.1073
Multi-head Attention	0.2475	0.7329	0.3700

Table 3: Ablation studies on effects of different word attention modules of the proposed network.

In order to investigate the impact of the ISML block, we perform an additional ablation study by adjusting the number of ISML blocks. As shown in Table 4, it is observed that as the number of iterations increases, the model's performance on this task generally improves, particularly when going from 1 to 2 iterations. One potential explanation for this is that ISML-2 initially introduces the interaction between emotion and cause. Notably, when the number of iterations reaches 16, we achieve the best performance in terms of the evaluation metrics.

Modules	Precision	Recall	F1
ISML-1	0.0396	0.0895	0.0498
ISML-2	0.1023	0.2055	0.1408
ISML-16	0.2475	0.7329	0.3700

Table 4: Ablation studies on the number of ISML blocks of the proposed network.

7.2 Task 2

Based on the results obtained from running GPT3.5 `text-davinci-003`, we have summarized the following observations:

1. The performance of prompts that combine Chinese data with English instructions is noticeably inferior to that of prompts consisting of Chinese data with Chinese instructions. This indicates that GPT text-davinci-003 does not effectively integrate information from different languages within the ECPE project. For data in different languages, it is recommended to use instruction prompts in the corresponding language.

2. The performance of few-shot learning is significantly higher than that of zero-shot learning. Although we only conducted experiments and observed the performance of few-shot and zero-shot learning in an environment where Chinese data was combined with English instructions, the improved performance of GPT still suggests the effectiveness of few-shot learning in LLM for handling ECPE tasks.

8 Error analysis

Sometimes GPT consistently provides incorrect responses. Many emotionless sentences are also flagged as having emotions by GPT. It seems that GPT is unable to accurately capture the implied emotions in sentences through understanding the context. Besides, another main issue encountered was the inability to capture the most prominent emotion, resulting in many instances where the corresponding cause lacked logical coherence. When successfully identifying the main emotion, there was a high probability of obtaining the correct result. The cases where this problem arose typically involved multiple descriptions of psychological activities or metaphors. Due to the presence of more than 10 sentences in each text, we did not perform a comprehensive analysis of numerous examples. Instead, our approach involved randomly selecting 10 incorrect and 10 correct results, comparing all the choices provided by GPT, and analyzing the patterns.

9 Contributions of group members

List what each member of the group contributed to this project here. For example:

- Zhenyu Lei: implemented the second half network in task 1(from ISML blocks to prediction); implemented several features, such as imbalanced loss functions and threshold; debugged and tuned the hyperparameters and conducted experiments for task 1;

	Zero-Shot			Five-Shot		
	Precision	Recall	F1	Precision	Recall	F1
GPT 3.5 (Baseline)	0.0552	0.1797	0.0845	0.1155	0.2260	0.1529
GPT 3.5 Prompt with Pairs Number	0.0381	0.1217	0.0581	0.1170	0.2696	0.1632
GPT 3.5 Prompt with Sequence Hint	0.05785	0.2434	0.0935	0.1125	0.2260	0.1502
GPT 3.5 Prompt with Both	0.0607	0.2	0.0931	0.1164	0.2521	0.1593
GPT 3.5 Chinese Instruction Prompt	0.1509	0.2087	0.1751	-	-	-
GPT 3.5 English Instruction Prompt	0.1156	0.1478	0.1297	-	-	-

Table 5: Baseline shows in the left part of Figure 8: zero shot with instruction prompt. Prompt with Pairs Number represents a hint about limitation of the number of pairs expected to output. Prompt with Sequence Hint represents a hint about the sequence of the appearance about the cause sentence and emotion sentence. Prompt with Both represents a hint includes both two guidence mentioned above. Chinese Instruction Prompt and English Instruction Prompt shows in the Figure 8.

- Yachan Liu: implemented and did some zero-shot experiments for task2, implemented task0, did a simple example for task3.
- Yuanming Tao: implemented the dataloader, the first half of the neural network modules, the training and evaluation loops, prepared figures, and conducted ablation studies for Task 1, conducted experiments to obtain preliminary results for Task 3.
- Xiaocheng Zhang: implemented and did the experiment on half of task2. Include zero-shot on Chinese prompt and English prompt, few-shot with different prompts.

All members contributed to the report writing.

10 Conclusion

10.1 Task 1

It is remarkable that despite using significantly fewer parameters than GPT-3.5, our model achieved a large precision improvement from 0.1509 to 0.2475. This demonstrates the effectiveness and efficiency of our approach in optimizing the model's performance for the seriously imbalanced dataset and labels. However, since the recall is still higher than the precision, it seems that the imbalance issue has not been solved thoroughly, although we've already achieved great progress compared to general LLM such as GPT 3.5.

In the future, data augmentation and contrastive learning are worth studying for this task. The implementation will not be trivial: the inherent sparsity of emotion-cause pairs prevents the vanilla data augmentation from working, so we need a smart way to do this. Besides, contrastive learning needs contrastive loss and heavy data augmentation. We believe these methods could be promising to solve the imbalance in the future.

10.2 Task 2

In conclusion, the obtained results are reasonable and predictable. The primary challenge encountered during this task stemmed from the cost associated with using LLM. The longer the text, the higher the cost incurred for each piece of information, necessitating a reduction in interactions with GPT and a simplification of the prompts. Additionally, the constraint of maximum token length prevented us from attempting few-shot learning with more than 5 examples. If this task were to continue, we would consider incorporating instruction prompts with ”pseudo Chain-of-Thought (CoT)” and the few-shot learning approach, exploring new prompts to assess whether GPT can achieve better performance. Furthermore, we would explore different GPT models such as GPT-3.5-turbo.

Prompt:
Given this text:
1 男友认为
2 高女士欺骗了他
3 经常称她假女人
4 其实她只是对鼻子和脸进行了整形
5 花费了上万元
6 但是旁边其他丰胸广告等
7 极易让人产生联想
8 男友甚至质疑
9 自己还有很多事欺骗了他
10 两人关系迅速恶化
11 她伤心地说
12 3年恋爱中
13 两人本来恩爱有加
14 4月底
15 高女士才将此事告诉父母
16 而男友已经离开四川
Extract emotion-cause pairs and return a list of tuples. In each tuple, the first element is clause numbers of emotion and the second element is clause numbers of emotion.

GPT's Feedback:
To extract emotion-cause pairs from the given text, we need to identify the clauses related to emotions and the clauses indicating the causes. Here's a list of tuples representing the emotion-cause pairs:
1. (2, 3): 高女士欺骗了他 (cause) -> 经常称她假女人 (emotion)
2. (4, 5): 其实她只是对鼻子和脸进行了整形 (cause) -> 花费了上万元 (emotion)
3. (7, 8): 旁边其他丰胸广告等 (cause) -> 极易让人产生联想 (emotion)
4. (9, 10): 自己还有很多事欺骗了他 (cause) -> 两人关系迅速恶化 (emotion)
5. (11, 12): 她伤心地说 (emotion) -> 3年恋爱中 (cause)
6. (14, 15): 高女士才将此事告诉父母 (cause) -> 男友已经离开四川 (emotion)
Please note that the numbering of clauses in the text is used to indicate the clause numbers in each emotion-cause pair.

Prompt:
Given this text:
1 The boyfriend believes
2 Ms. Gao deceived him
3 Often called her a fake woman
4 In fact, she only had plastic surgery on her nose and face
5 It cost over ten thousand yuan
6 But there were other breast enhancement advertisements nearby
7 Which easily led to associations
8 The boyfriend even questioned
9 That she had deceived him about many other things
10 The relationship between them quickly deteriorated
11 She said sadly
12 In the three years of their relationship
13 They were originally deeply in love
14 At the end of April
15 Ms. Gao finally told her parents about this matter
16 While the boyfriend had already left Sichuan
Extract emotion-cause pairs and return a list of tuples. In each tuple, the first element is clause numbers of emotion and the second element is clause numbers of emotion.

GPT's Feedback:
To extract emotion-cause pairs from the given text, we need to identify the clauses related to emotions and the clauses indicating the causes. Here's a list of tuples representing the emotion-cause pairs:
1. (2, 3): Ms. Gao deceived him (cause) -> Often called her a fake woman (emotion)
2. (4, 5): In fact, she only had plastic surgery on her nose and face (cause) -> It cost over ten thousand yuan (emotion)
3. (7, 8): Other breast enhancement advertisements nearby (cause) -> Which easily led to associations (emotion)
4. (9, 10): She had deceived him about many other things (cause) -> The relationship between them quickly deteriorated (emotion)
5. (11, 12): She said sadly (emotion) -> In the three years of their relationship (cause)
6. (14, 15): Ms. Gao finally told her parents about this matter (cause) -> The boyfriend had already left Sichuan (emotion)
Please note that the numbering of clauses in the text is used to indicate the clause numbers in each emotion-cause pair.

Figure 7: A example of zero-shot GPT prompt, the left part shows the baseline prompt and GPT's feedback. The right part is a translation of the text.

Prompt
Think step by step. Give a response to the question in this format
---
Question:
{$ Question needs answer}
Examples:
[{$ Several representative examples of the given question with answers}]
Content:
{$ Passage with index which includes the emotion-cause pairs}

Question:
Extract emotion-cause pairs and return a list of tuple based on the given content.
In each tuple, the first element is the index of the emotion clause and the second element is the index of cause clause.
Examples:
[
{Content:
1 三十多年相伴
2 彭晓玉已经离不开丈夫
...
18 我才更有动力
Answer:[[12, 11]]}
]
Content:
1 男友认为
2 高女士欺骗了他
...
16 而男友已经离开四川

GPT's Feedback:
Answer: [[2, 3], [8, 9]]

Prompt
Think step by step. Give a response to the question in this format
---
Question:
{$ Question needs answer}
Examples:
[{$ Several representative examples of the given question with answers}]
Content:
{$ Passage with index which includes the emotion-cause pairs}

Question:
Extract 1 to 4 emotion-cause pairs and return a list of tuple based on the given content.
In each tuple, the first element is the index of the emotion clause and the second element is the index of cause clause.
The cause clause usually appears in the front of emotion clause.
Examples:
[
{Content:
1 三十多年相伴
2 彭晓玉已经离不开丈夫
...
18 我才更有动力
Answer:[[12, 11]]}
]
Content:
1 男友认为
2 高女士欺骗了他
...
16 而男友已经离开四川

GPT's Feedback:
Answer: [[2, 3]]

Prompt
文档："
1 男友认为
2 高女士欺骗了他
...
16 而男友已经离开四川"

以上文档的每一行代表一个子句，每行开头的数字表示该子句的编号。我们将包含情绪表达的子句称为"情绪子句"，将导致情绪发生的子句称为"原因子句"。基于以上内容，分别完成以下任务。
1. 用一句话描述以上文档中包含的情绪以及它对应的原因。
2. 输出任务1中情绪子句的编号，你只需要找到情绪强度最强的那一个子句。
3.对于任务2中的每个情绪子句，找到导致其情绪的原因子句，并输出原因子句的编号，你只需要找到最符合要求的那一个子句。
4.将以上具有因果关系的子句匹配成'(情绪子句编号，原因子句编号)'的形式，最后将所有的编号配对按照集合的形式输出，例如 任务：[[1,2],[3,4]]。注意：情绪子句和原因子句有可能是同一个子句，且最终只需要输出情绪和原因最明显的配对。

GPT's Feedback:
任务1：
情绪：男友认为高女士欺骗了他，并质疑自己被欺骗的事实。
原因：高女士对鼻子和脸进行整形手术，花费了上万元，导致男友产生了怀疑和恶化关系的情绪。
任务2：
情绪子句的编号：8
任务3：
情绪子句8的原因子句编号：9
任务4：[(8, 9)]

Prompt
Content: "
1 男友认为
2 高女士欺骗了他
...
16 而男友已经离开四川"

Each line of the content represents a clause, and the number at the beginning of each line indicates the clause number. We refer to the clauses that contain emotional expressions as "emotion clauses" and the clauses that cause emotions as "cause clauses". Based on the content, complete the following tasks:
1. Describe the emotion contained in the document in one sentence and its corresponding cause.
2. Output the clause number of the emotion clause for Task 1. You only need to find the clause with the strongest emotional intensity.
3. For each emotion clause in Task 2, find the cause clause that caused the emotion and output the clause number of the cause clause. You only need to find the clause that best meets the requirements.
4. Match the clauses with causal relationships in the form of '[emotion clause number, cause clause number]', and output all paired numbers in list format, for example, Task 4: [[1,2],[3,4]].
Note: the emotion clause and the cause clause may be the same clause, and only the most obvious emotion-cause pairing needs to be outputted.
Besure to output in this format: Task 1: ... Task 2: ...

GPT's Feedback:
Task 1: The emotion contained in the document is sadness, and its corresponding cause is the boyfriend's belief that the woman deceived him.
Task 2: The clause number of the emotion clause with the strongest emotional intensity is 10.
Task 3: The cause clause that caused the emotion in clause 10 is clause 9.
Task 4: [[10, 9]]

Figure 8: Four examples of few-shot GPT prompt. Upper left graph is a one-shot example without any futher hint. Upper right graph is a one-shot example with both hint includes the limitation of pair number and sequence. Lower left graph is the Chinese Instruction Prompt which used Chinese with several tasks which guide GPT to generate 'relative' result. Lower right graph is the English version of the Chinese Instrction Prompt.

AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - No.

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - your response here
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - your response here

References

Chen, X., Liang, C., Yu, A. W., Zhou, D., Song, D., and Le, Q. V. (2020). Neural symbolic reader: Scalable integration of distributed and symbolic representations for reading comprehension. In *International Conference on Learning Representations*.

Cui, P., Hu, D., and Hu, L. (2021). Listreader: Extracting list-form answers for opinion questions.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ding, Z., Xia, R., and Yu, J. (2020a). ECPE-2D: Emotion-Cause Pair Extraction based on Joint Two-Dimensional Representation, Interaction and Prediction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3161–3170, Online. Association for Computational Linguistics.

Ding, Z., Xia, R., and Yu, J. (2020b). End-to-end emotion-cause pair extraction based on sliding window multi-label learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3574–3583, Online. Association for Computational Linguistics.

Fan, C., Yuan, C., Du, J., Gui, L., Yang, M., and Xu, R. (2020). Transition-based Directed Graph Construction for Emotion-Cause Pair Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3707–3717, Online. Association for Computational Linguistics.

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*, pages 1639–1649, Austin, Texas. Association for Computational Linguistics.

Hu, M., Peng, Y., Huang, Z., and Li, D. (2019). A multi-type multi-span network for reading comprehension that requires discrete reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1596–1606, Hong Kong, China. Association for Computational Linguistics.

Hu, M., Peng, Y., Huang, Z., Qiu, X., Wei, F., and Zhou, M. (2018). Reinforced mnemonic reader for machine reading comprehension.

Li, H., Tomko, M., Vasardani, M., and Baldwin, T. (2022). MultiSpanQA: A dataset for multi-span question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1250–1260, Seattle, United States. Association for Computational Linguistics.

Segal, E., Efrat, A., Shoham, M., Globerson, A., and Berant, J. (2020). A simple and effective model for answering multi-span questions. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3074–3080, Online. Association for Computational Linguistics.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2018). Bidirectional attention flow for machine comprehension.

Wang, Z., Xie, Q., Ding, Z., Feng, Y., and Xia, R. (2023a). Is chatgpt a good sentiment analyzer? a preliminary study. *arXiv preprint*.

Wang, Z., Xie, Q., Ding, Z., Feng, Y., and Xia, R. (2023b). Is chatgpt a good sentiment analyzer? a preliminary study. *ARXIV.ORG*.

Wei, P., Zhao, J., and Mao, W. (2020). Effective Inter-Clause Modeling for End-to-End Emotion-Cause Pair Extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3171–3181, Online. Association for Computational Linguistics.

Wu, S., Chen, F., Wu, F., Huang, Y., and Li, X. (2020). A Multi-Task Learning Neural Network for Emotion-Cause Pair Extraction. *Santiago de Compostela*.

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*, pages 1003–1012, Florence, Italy. Association for Computational Linguistics.

Yu, A. W., Dohan, D., Luong, M.-T., Zhao, R., Chen, K., Norouzi, M., and Le, Q. V. (2018). Qanet: Combining local convolution with global self-attention for reading comprehension.

Zheng, X., Liu, Z., Zhang, Z., Wang, Z., and Wang, J. (2022). UECA-prompt: Universal prompt for emotion cause analysis. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 7031–7041, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.