Identifying Chinese Opinion Expressions with Extremely-Noisy Crowdsourcing Annotations

Xin Zhang[1], **Guangwei Xu, Yueheng Sun**[2], **Meishan Zhang**[3]*, **Xiaobin Wang, Min Zhang**[3]
[1]School of New Media and Communication, Tianjin University, China
[2]College of Intelligence and Computing, Tianjin University, China
[3]Institute of Computing and Intelligence, Harbin Institute of Technology (Shenzhen), China
{hsinz,yhs}@tju.edu.cn, {ahxgwOnePiece, mason.zms}@gmail.com
czwangxiaobin@foxmail.com, zhangmin2021@hit.edu.cn

Abstract

Recent works of opinion expression identification (OEI) rely heavily on the quality and scale of the manually-constructed training corpus, which could be extremely difficult to satisfy. Crowdsourcing is one practical solution for this problem, aiming to create a large-scale but quality-unguaranteed corpus. In this work, we investigate Chinese OEI with extremely-noisy crowdsourcing annotations, constructing a dataset at a very low cost. Following Zhang et al. (2021), we train the annotator-adapter model by regarding all annotations as gold-standard in terms of crowd annotators, and test the model by using a synthetic expert, which is a mixture of all annotators. As this annotator-mixture for testing is never modeled explicitly in the training phase, we propose to generate synthetic training samples by a pertinent mixup strategy to make the training and testing highly consistent. The simulation experiments on our constructed dataset show that crowdsourcing is highly promising for OEI, and our proposed annotator-mixup can further enhance the crowdsourcing modeling.

1 Introduction

Opinion mining is a fundamental topic in the natural language processing (NLP) community, which has received great attention for decades (Liu and Zhang, 2012). Opinion expression identification (OEI) is a standard task of opinion mining, which aims to recognize the text spans that express particular opinions (Breck et al., 2007). Figure 1 shows two examples. This task has been generally solved by supervised learning (İrsoy and Cardie, 2014) with the well-established corpus annotated by experts. Almost all previous studies are based on English datasets such as MPQA (Wiebe et al., 2005).

By carefully examining this task, we can find that the corpus annotation of opinion expressions is by no means an easy process. It is highly ambiguous across different persons. As shown in Figure 1, it is very controversial to define the boundaries of opinion expressions (Wiebe et al., 2005). Actually, this problem is extremely serious for languages such as Chinese, which is based on characters even with no explicit and clearly-defined word boundaries. Thus, Chinese-alike languages will inevitably involve more ambiguities.

Figure 1: Two examples of opinion expression identification with crowdsourcing and expert annotations in our constructed dataset. The left and right sentences are of positive and negative polarities, respectively.

In order to obtain a high-quality corpus, we usually need to train the annotators with great efforts, making them acquainted with a specific fine-grained guideline drafted by experts, and then start the data annotation strictly. Finally, it is better with a further expert checking on borderline cases where the annotators disagree most to ensure the quality of the annotated corpus. Apparently, the whole process is quite expensive. Thus, crowdsourcing with no training (just a brief guideline) and no expert checking is more practical in real considerations (Snow et al., 2008). While on the other hand, the difficulty of the Chinese OEI task might lead to very low-quality annotations by crowdsourcing.

In this work, we present the first study of Chinese OEI by using crowdsourcing. We manually construct an OEI dataset by crowdsourcing, which is used for training. Indeed, the dataset is cheap but with a great deal of noises according to our initial observation. We also collect the small-scale devel-

*Corresponding author.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 2801 - 2813
May 22-27, 2022 ©2022 Association for Computational Linguistics

opment and test corpus with expert annotations for evaluation.[1] Our dataset is constructed over a set of Chinese texts closely related to the COVID-19 topic. Following, we start our investigation by using a strong BERT-BiLSTM-CRF model, treating the OEI task as a standard sequence labeling problem following the previous studies (Breck et al., 2007; İrsoy and Cardie, 2014; Katiyar and Cardie, 2016). Our primary goal is to answer whether these extremely-noisy crowdsourcing annotations include potential value for the OEI task.

In order to make the best use of our crowdsourcing corpus, we follow Zhang et al. (2021) to treat all crowd annotations as gold-standard in terms of different annotators. We introduce the annotator-adapter model, which employs the crowdsourcing learning approach of Zhang et al. (2021) in OEI for the first time. It jointly encodes both texts and annotators, then predicts the corresponding crowdsourcing annotations in the BERT-BiLSTM-CRF architecture. Concretely, we train the annotator-adapter model by each individual annotator and the corresponding annotations, then test the model by using a pseudo expert annotator, which is a linear mixture of crowd annotators. Considering that this expert is never modeled during the training, we further exploit a simple mixup (Zhang et al., 2018) strategy to simulate the expert decoding accurately.

Experimental results show that crowdsourcing is highly competitive, giving an overall F1 score of 53.86 even with a large-scale of noises, while the F1 score of expert corpus trained model is 57.08. We believe that this performance gap is totally acceptable for building OEI application systems. In addition, our annotator-mixup strategy can further boost the performance of the annotator-adapter model, giving an F1 increase of $54.59 - 53.86 = 0.73$. We conduct several analyses to understand the OEI with crowdsourcing and our suggested methods comprehensively.

In summary, we make three majoring contributions as a whole in this work:

- We present the initial work of investigating the OEI task with crowdsourcing annotations, showing its capability on Chinese.
- We construct a Chinese OEI dataset with crowdsourcing annotations, which is not only valuable for Chinese OEI but also instructive for crowdsourcing researching.
- We introduce the annotator-adapter for crowdsourcing OEI and propose the annotator-mixup strategy, which can effectively improve the crowdsourcing modeling.

All of our codes and dataset will be available at github.com/izhx/crowd-OEI for research purpose.

[1] In addition, we provide expert annotations of trainset to train a upper-bound model.

No.	Chinese / English
1	澳大利亚籍返京女子不隔离外出跑步 / The Australian woman running outside without isolation in Beijing
2	单玉厚 / Yuhou Shan
3	李文亮医生 / Dr. Li Wenliang
4	是谁发现了病毒 / Who finds the virus
5	方方日记 / Fang Fang's Diary
6	歌诗达赛琳娜号 / Goethe Serena
7	新冠可通过气溶胶传播 / COVID-19 can transmit via aerosol

Table 1: Seven hot topics we targeted.

2 Dataset

The outbreak of COVID-19 brings strong demand for building robust Chinese opinion mining systems, which are practically built in a supervised manner. A large-scale training corpus is the key to the system construction, while almost all existing related datasets are in English (Wiebe et al., 2005). Hence, we manually construct a Chinese OEI dataset by crowdsourcing. We focus on opinion expressions with **positive** or **negative** polarities only. The construction consists of four steps: (1) text collection, (2) annotator recruitment, (3) crowd annotation, and (4) expert checking and correction.

2.1 Text Collection

We choose the Sina Weibo[2], which is a Chinese social media platform similar to Twitter, as our data source. To collect the texts strongly related to COVID-19, we select around 8k posts that are created from January to April 2020 and related to seven hot topics (Table 1). To make these posts ready for annotating, we use HarvestText[3] to clean them and segment the resulting texts into sentences. Next, we conduct another cleaning step to remove the duplicates and sentences with relatively poor written styles (e.g., high-proportion of non-Chinese symbols, very short /long length, etc.).

After the above procedure, there are still a large proportion of sentences that involve no sentiment.

[2] https://weibo.com

[3] https://github.com/blmoistawinde/HarvestText

So we filter out them by a BERT sentiment classifier that trained on an open-access Weibo sentiment classification dataset.[4] Only sentences with high confidence of not expressing any sentiment are dropped,[5] we can therefore keep the most valuable contents while avoiding unnecessary annotations and thus reduce the overall annotating cost.

2.2 Annotator Recruitment

We have five professionals who have engaged in the annotation of sentiment and opinion-related tasks previously and are with rich experience as experts. They annotate 100 sentences together as examples (i.e., label the positive and negative opinion expressions inside the texts), and establish a simple guideline based on their consensus after several discussions. The guideline includes the task definition and a description of annotation principle.[6]

Next, we recruit 75 (crowd) students in our university for annotating. They come from different grades and different majors, such as Chinese, Literature, and Translation. We offer them the above annotation guideline to understand the task. We choose the doccano[7] to build up our annotation platform, and let these annotators be familiar with our task by the expert-annotated examples.

2.3 Crowd Annotation

When all crowd workers are ready, we start the crowd annotation phase. The prepared texts are split into micro-tasks so that each one consists of 500 sentences. Then we assign 3 to 5 workers to each micro-task, and their identities are remained hidden from each other. Each worker will not access a new task unless their current one is finished.

In the annotation of each sentence, workers need to label the positive and negative opinion expressions according to the guideline and their understandings. The number of positive or negative expressions in one sentence has no limit. They can also mark a sentence as "No Opinion" and skip it if they think there are no opinion expressions inside.

2.4 Expert Checking and Correction

After all crowd annotations are accomplished, we randomly select a small proportion of sentences and let experts reannotate them, resulting in the **gold-standard** *development* and *test* corpus.[8] Specifically, for each sentence, we let 2 experienced experts individually reannotate it with references from the corresponding crowdsourcing annotations. They will give the final annotation of each sentence if their answers reach an agreement. And if they have divergences, a third expert will help them to modify answers and reach the agreement.

Then, we let all five experts go through the remaining dataset[9], selecting the best annotations for each sentence, which can be regarded as the **silver-standard** *training* corpus. In the selection, Each sentence is assigned to 1 expert, and the expert is only allowed to choose one (or several identical) best answer(s) from all the candidate crowdsourcing annotations. Finally, only for comparisons, we also annotated the **gold-standard** *training* corpus, which will not be used in our model training.

Dataset		Number of			Average
Section	Quality	Unique Annotation	Positive Expression	Negative Expression	Span Length
Train	crowd	32582	11640	35263	5.05
	silver	8047	4167	11411	4.71
	gold	8047	3488	10096	4.79
Dev	crowd	3427	2338	3905	5.22
	gold	803	706	1035	5.02
Test	crowd	6265	3573	5290	4.48
	gold	1517	999	1373	4.30

Table 2: Data statistics of our constructed dataset. For gold and silver corpus, each annotation corresponds to one sentence. For the crowd corpus, each sentence has 3 to 5 annotations. So we have a total number of $803 + 1517 + 8047 = 10,367$ unique sentences and $32,582 + 3427 + 6265 = 42,274$ crowd annotations.

2.5 Dataset Statistics

In the end, we arrive at $42,274$ crowd annotations by 70 valid annotators,[10] covering $10,367$ sentences. A total number of $803 + 1517 = 2320$ sentences, including expert annotations, would be used for development and test evaluations. Table 2 shows the overall data statistics. The average number of annotators per sentence is 4.05, and each annotator labels an average of 827 sentences in the whole corpus. The overall Cohen's Kappa value of the crowd annotations is 0.35. When ignoring the

[4] ChineseNlpCorpus - weibo_senti_100k

[5] Note that there are still a small number of sentences in our final dataset that have no opinion expression inside.

[6] We share the guideline in the Appendix A.

[7] https://github.com/doccano/doccano

[8] The corresponding crowdsourcing annotations consist of the **crowdsourcing** *development* and *test* corpus.

[9] The remaining part is the **crowdsourcing** *training* corpus.

[10] We removed 5 annotators who gave up this work in their first assigned task as a basic quality assurance.

characters which no annotators think that they are in any expression, the Kappa is only 0.17.[11]

The Kappa values are indeed very low, indicating the great and unavoidable ambiguities of the task with natural annotations.[12] However, these values do not make much sense since we do not impose any well-designed comprehensive guidelines during annotation. In fact, a comprehensive guideline for crowd workers is almost impracticable in our task, because they are quite often to disagree with a particular guideline by their own unique and naive understandings. If we impose such a guideline to them forcibly, the annotation cost would be increased drastically (i.e., at least ten times more expensive according to our preliminary investigation) for their reluctance as well as endless expert guidance. In the remaining of this work, we will try to verify the real value of these crowdsourcing annotations empirically: Is the collected training corpus really beneficial for our Chinese OEI task?

3 Methodology

The OEI task aims to extract all polarized text spans that express certain opinions in a sentence. It can be naturally converted into a sequence labeling problem by using the BIO schema, tagging each token by the boundary information of opinion expressions, where "B-X" and "I-X" (i.e., "X" can be either "POS" or "NEG" denoting the polarity) indicate the start and other positions of a certain expression, and "O" denotes a token do not belong to any expression. In this work we adopt the CRF-based system (Breck et al., 2007) to the neural setting and enhance it with BiLSTM encoder as well as pre-trained BERT representation.

3.1 BERT-BiLSTM-CRF Baseline

Given a sentence $\boldsymbol{x} = x_1 \cdots x_n$ (where n denotes the sentence length), we first convert it into contextual representations $\boldsymbol{r}_1 \cdots \boldsymbol{r}_n$ by the pre-trained BERT with adapter tuning (Houlsby et al., 2019):

$$\boldsymbol{r}_1 \cdots \boldsymbol{r}_n = \text{ADBERT}(x_1 \cdots x_n). \quad (1)$$

Unlike the standard BERT exploration, ADBERT introduces two extra adapter modules inside each transformer layer, as shown in Figure 2 for the details. With this modification, we do not need fine-tuning all BERT parameters, and instead, learning the parameters of adapters is enough for obtaining a strong performance. Thus ADBERT is more parameter efficient. The standard adapter layer can be formalized as:

$$\begin{aligned} \text{down-proj:} \quad & \boldsymbol{h}_{\text{mid}} = \text{GELU}(\boldsymbol{W}_{\text{down}}\boldsymbol{h}_{\text{in}} + \boldsymbol{b}_{\text{down}}), \\ \text{up-proj:} \quad & \boldsymbol{h}_{\text{out}} = \boldsymbol{W}_{\text{up}}\boldsymbol{h}_{\text{mid}} + \boldsymbol{b}_{\text{up}} + \boldsymbol{h}_{\text{in}}, \end{aligned} \quad (2)$$

where $\boldsymbol{W}_{\text{down}}$, $\boldsymbol{W}_{\text{up}}$, $\boldsymbol{b}_{\text{down}}$ and $\boldsymbol{b}_{\text{up}}$ are model parameters, which are much smaller than the parameters of transformer in scale, and the dimension size of $\boldsymbol{h}_{\text{mid}}$ is also smaller than that of the corresponding transformer dimension.[13]

The rest part of the baseline is a standard BiLSTM-CRF model, which is a stack of BiLSTM, MLP and CRF layers, and then we can obtain sequence-level scores for each candidate output $\boldsymbol{y}$:

$$\begin{aligned} \text{score}(\boldsymbol{y}) &= \text{BiLSTM-CRF}([\boldsymbol{r}_1 \cdots \boldsymbol{r}_n]), \\ p(\boldsymbol{y}) &= \frac{\exp\big(\text{score}(\boldsymbol{y})\big)}{\sum_{\tilde{Y}} \exp\big(\text{score}(\tilde{\boldsymbol{y}})\big)}, \end{aligned} \quad (3)$$

where $p(\boldsymbol{y})$ is the probability of the given ground-truth, and $\tilde{Y}$ is all possible outputs for score normalization. The model parameters are updated by the sentence-level cross-entropy loss $\mathcal{L} = -\log p(\boldsymbol{y}^*)$ when $\boldsymbol{y}^*$ is regarded as gold-standard.

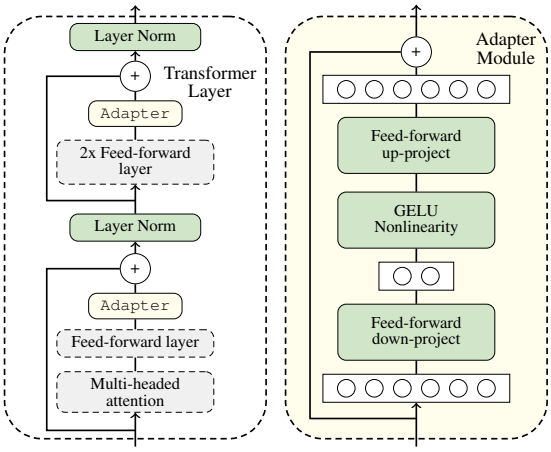

Figure 2: The Adapter (right) and Transformer integrated with Adapter inside (left). During the adapter tuning, green layers are trainable, including the adapters, the LayerNorm, and other task-specific modules.

[11] To compute the Kappa value of sequential annotations, we treat each token (not sentence) as an instance, and then aggregate the results of one sentence by averaging.

[12] The average value of F1 scores that each annotator against the expert is 41.77%, which is significantly lower than 60%+ of crowdsourcing NER dataset (Rodrigues et al., 2014b).

[13] The dimension sizes of $\boldsymbol{h}_{\text{in}}$ and $\boldsymbol{h}_{\text{out}}$ are consistent with the corresponding transformer hidden states.

Crowdsourcing training. In the crowdsourcing setting, we only have annotations from multiple non-expert annotators, thus no gold-standard label is available for our training. To handle the situation, we introduce two straightforward and widely-used methods. First, we treat all annotations uniformly as training instances, despite that they may offer noises for our training objective, which is denoted by `All` for short. Second, we exploit majority voting[14] to obtain an aggregated answer of each sentence for model training, denoted as `MV`.

3.2 Annotator Adapter

In most previous crowdsourcing studies, there is a common agreement that crowd annotations are noisy, which should be rectified during training (Rodrigues et al., 2014a; Nguyen et al., 2017; Simpson and Gurevych, 2019). Zhang et al. (2021) propose to regard all crowdsourcing annotations as gold-standard, and introduce a representation learning model to jointly encode the sentence and the annotator and extract annotator-aware features, which models the unique understandings of annotators (this setting is indeed very consistent with our corpus). Since our constructed dataset has no gold-standard training labels[15], we adopt their unsupervised representation learning approach, which is named `annotator-adapter`. It applies the Parameter Generator Network (PGN) (Platanios et al., 2018; Jia et al., 2019; Üstün et al., 2020) to generate annotator-specific adapter parameters for the ADBERT, as shown in Figure 3.

Given an input sentence-annotator pair $(\boldsymbol{x} = x_1, \ldots, x_n, a)$, we exploit an embedding layer to convert the annotator ID a into its vectorial form $\boldsymbol{e}^a$, and then PGN is used to generate the model parameters of several high-level adapter layers inside BERT conditioned by $\boldsymbol{e}^a$. Concretely, we apply PGN to the last p layers of BERT, where p is one hyper-parameter of our model. We refer to PGN-ADBERT for the updated input representation.

Formally, for an adapter defined by Equation 2, all its parameters are dynamically generated by:

$$\begin{aligned} \boldsymbol{W}_{\text{down}} &= \boldsymbol{T}_{\boldsymbol{W}_{\text{down}}} \times \boldsymbol{e}^a, \quad \boldsymbol{b}_{\text{down}} = \boldsymbol{T}_{\boldsymbol{b}_{\text{down}}} \times \boldsymbol{e}^a, \\ \boldsymbol{W}_{\text{up}} &= \boldsymbol{T}_{\boldsymbol{W}_{\text{up}}} \times \boldsymbol{e}^a, \quad \boldsymbol{b}_{\text{up}} = \boldsymbol{T}_{\boldsymbol{b}_{\text{up}}} \times \boldsymbol{e}^a, \end{aligned} \tag{4}$$

[14]The voting is conducted at the token-level and then merge continuous tokens if they belong to a same-type expression.

[15]We have added the gold-standard annotations in the revision of this work, but we keep this data setting.

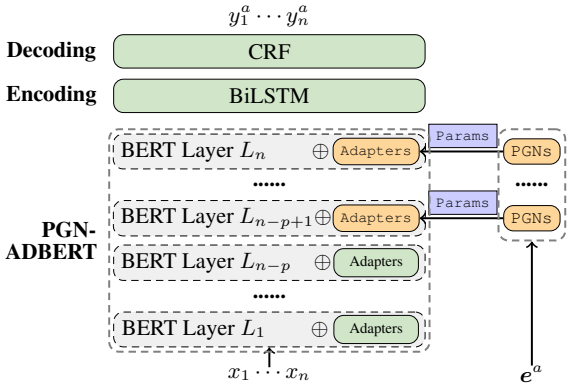

Figure 3: The annotator-adapter model. Given a joint input of the text $x_1 \cdots x_n$ and the annotator ID a, we first convert a to its embedding $\boldsymbol{e}^a$. Then, PGN use $\boldsymbol{e}^a$ generate annotator-specific parameters for the adapters in top p BERT layers (i.e., from L_n to L_{n-p+1}) to compute annotator-aware input representations. Finally, the BiLSTM encode the representations to high-level features and the CRF decoder predict the labels $y_1^a \cdots y_n^a$ that a gives to $x_1 \cdots x_n$.

where $\boldsymbol{T}_{\boldsymbol{W}_{\text{down}}}$, $\boldsymbol{T}_{\boldsymbol{b}_{\text{down}}}$, $\boldsymbol{T}_{\boldsymbol{W}_{\text{up}}}$ and $\boldsymbol{T}_{\boldsymbol{b}_{\text{up}}}$ are learnable model parameters for the PGN-ADBERT. For any matrix-format model parameter $\boldsymbol{W} \in \mathbb{R}^{M \times N}$, we have $\boldsymbol{T}_{\boldsymbol{W}} \in \mathbb{R}^{M \times N \times d}$, where d is the dim of the annotator embedding. Similarly, for the vectorial parameter $\boldsymbol{b} \in \mathbb{R}^N$, we have $\boldsymbol{T}_{\boldsymbol{b}} \in \mathbb{R}^{N \times d}$.

Thus, the overall input representation of the annotator-adapter can be rewritten as:

$$\boldsymbol{r}_1 \cdots \boldsymbol{r}_n = \text{PGN-ADBERT}(x_1 \cdots x_n, \boldsymbol{e}^a), \tag{5}$$

which jointly encodes the text and the annotator.

At the training stage, it uses the embedding of crowd annotators to generate crowd model parameters to learn crowd annotations. At the inference stage, it uses the centroid point of all annotator embeddings to estimate the expert, predicting the high-quality opinion expressions for raw texts. This expert embedding can be computed directly by:

$$\boldsymbol{e}^{\text{expert}} = \frac{1}{|A|} \sum_{a \in A} \boldsymbol{e}^a, \tag{6}$$

where A represents all annotators.

3.3 Annotator Mixup

By scrutinizing the annotator-adapter model, we can find that there is a minor mismatch during the model training and testing. During the training, the input annotators are all encoded individually. While during the testing, the input expert is a mixture of the crowd annotators, which is never modeled. To tackle this divergence, we introduce the

mixup (Zhang et al., 2018) strategy over the individual annotators to generate a number of synthetic samples with linear mixtures of annotators, making the training and testing highly similar.

The mixup strategy is essentially an effective data augmentation method that has received increasing attention recently in the NLP community (Zhang et al., 2020; Sun et al., 2020). The method is applied between two individual training instances originally, by using linear interpolation over a hidden input layer and the output. In this work, we confine the mixup onto the two training instances with the same input sentence for annotator mixup.

Formally, given two training instances $(\boldsymbol{x}_1 \circ a_1, \boldsymbol{y}_1)$ and $(\boldsymbol{x}_2 \circ a_2, \boldsymbol{y}_2)$, the mixup is executed only when $\boldsymbol{x}_1 = \boldsymbol{x}_2$, thus the interpolation is actually performed between $(a_1, \boldsymbol{y}_1)$ and $(a_2, \boldsymbol{y}_2)$. Concretely, the input interpolation is conducted at the embedding layer, and the output interpolation is directly mixed at the sentence-level:

$$\begin{aligned} \boldsymbol{e}^{\text{mix}} &= \lambda \boldsymbol{e}^{a_1} + (1-\lambda)\boldsymbol{e}^{a_2}, \\ \boldsymbol{y}_{\text{mix}} &= \lambda \boldsymbol{y}_1 + (1-\lambda)\boldsymbol{y}_2, \end{aligned} \tag{7}$$

where $\lambda \in [0, 1]$ is a hyper-parameter which is usually sampled from the $\text{Beta}(\alpha, \alpha)$ distribution, and $\boldsymbol{y}_*$ is the one-hot vectorial form, where $* \in [1, 2, \text{mix}]$.[16] Finally, the loss objective of the new instance is calculated by:

$$\mathcal{L}_{\text{mix}} = -\log \frac{\exp\left(\text{score}(\boldsymbol{y}_{\text{mix}})\right)}{\sum_{\widetilde{Y}} \exp\left(\text{score}(\tilde{\boldsymbol{y}})\right)}, \tag{8}$$

where all scores are computed based on $\boldsymbol{x}_1/\boldsymbol{x}_2$ and $\boldsymbol{e}^{\text{mix}}$, and $\widetilde{Y}$ is all possible outputs for $\boldsymbol{x}_1/\boldsymbol{x}_2$.

Finally, we can produce a number of augmented instances by the annotator mixup. These instances, together with the original training instances, are used to optimize our model parameters. The enhanced model is able to perform inference more robustly by using the mixture (i.e, average) of annotators, which is the estimation of the expert.

4 Experiment

4.1 Setting

Evaluation. We use the span-level precision (P), recall (R) and their F1 for evaluation, since OEI is essentially a span recognition task. Following Breck et al. (2007); İrsoy and Cardie (2014), we exploit three types of metrics, namely *exact* matching, *proportional* matching and *binary* matching, respectively. The *exact* metric is straightforward and has been widely applied for span-level entity recognition tasks, which regards a predicted opinion expression as correct only when its start-end boundaries and polarity are all correct. Here we exploit the *exact* metric as the major method. The two other metrics are exploited because the exact boundaries are very difficult to be unified even for experts. The *binary* method treats an expression as correct when it contains an overlap with the ground-truth expression, and the *proportional* method uses a balanced score by the proportion of the overlapped area referring to the ground-truth.

We use the best-performing model on the development corpus to evaluate the performance of the test corpus. All experiments are conducted on a single RTX 2080 Ti card at an 8-GPU server with a 14 core CPU and 128GB memory. We run each setting by 5 times with different random seeds, and the median evaluation scores are reported.

Hyper-parameters. We exploit the bert-base-chinese for input representations.[17] The adapter bottleneck size and the BiLSTM hidden size are set to 128 and 400, respectively. For the annotator-adapter, we set the annotator embedding size $d = 8$ and generate the adapter parameters for the last $p = 6$ BERT layers. For the annotator mixup, we set α of the $\text{Beta}(\alpha, \alpha)$ distribution to 0.5.

We apply the sequential dropout to the input representations, which randomly sets the hidden vectors in the sequence to zeros with a probability of 0.2, to avoid overfitting. We use the Adam algorithm to optimize the parameters with a constant learning rate 1×10^{-3} and a batch size 64, and apply the gradient clipping mechanism by a maximum value of 5.0 to avoid gradient explosion.

Baselines. Two annotator-agnostic baselines (i.e., `ALL` and `MV`) and the silver-corpus trained model `Silver` are all implemented in the same baseline structure and hyper-parameters. We also implement two annotator-aware methods presented in Nguyen et al. (2017), where the annotator-dependent noises have been modeled explicitly. The `LSTM-Crowd` model encodes the output label bias (i.e., noises) for each individual annotator (biased-distributions) towards the expert (zeroed-distribution), and the `LSTM-Crowd-cat` model

[16] Note that $\boldsymbol{y}_*$ is at the sentence-level, where the dimension size is the number of all possible outputs of the given input. We mix the loss of $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ instead of themself in practice.

[17] https://github.com/google-research/bert

Method	Exact			Proportional			Binary		
	P	R	F1	P	R	F1	P	R	F1
Gold	61.12	53.54	57.08	81.97	72.28	76.82	85.79	77.51	81.44
Silver	55.27	53.25	54.24	75.79	73.01	74.37	81.23	78.25	79.71
ALL	61.06	45.49	52.14	82.47	61.44	70.42	86.98	64.80	74.27
MV	53.95	50.97	52.42	74.23	70.13	72.12	78.98	74.62	76.74
LSTM-Crowd (Nguyen et al., 2017)	60.55	47.68	53.35	**83.79**	61.32	70.82	**88.71**	64.92	74.98
LSTM-Crowd-cat (Nguyen et al., 2017)	59.07	47.51	52.66	77.56	62.39	69.15	83.70	67.33	74.63
BSC-seq (Simpson and Gurevych, 2019)	40.80	**59.27**	48.33	55.35	**82.41**	66.23	60.66	**90.33**	72.58
Annotator-Adapter (Zhang et al., 2021) †	61.08	48.16	53.86	81.70	65.40	72.65	87.20	69.81	77.55
Annotator-Adapter + mixup †	**61.27**	49.22	**54.59**	81.82	68.30	**74.45**	87.02	71.48	**78.49**

Table 3: The test results, where all methods are backended by BERT-BiLSTM-CRF for a fair comparison. The `Gold` and `Silver` denotes models trained with expert annotations and sentence-level expert aggregation (silver-standard in §2.4), respectively. The † indicates statistical significance compared to baselines with $p < 0.01$ by paired t-test.

applies a similar idea but implementing at the BiLSTM hidden layer. During the testing, zero-vectors are exploited to simulate the expert accordingly. Their main idea is to reach a robust training on the noisy dataset, which is totally different from our approach. In addition, we aggregate crowd labels of the training corpus by a Bayesian inference method (Simpson and Gurevych, 2019), namely `BSC-seq`, based on their code[18] and then evaluate its results with the same BERT-BiLSTM-CRF architecture.

4.2 Main Results

Table 3 shows the test results on our dataset. In general, the *exact* matching scores are all at a relatively low level, demonstrating that precise opinion boundaries are indeed difficult to identify. With the gradual relaxation of metrics (from *exact* to *binary*), scores are increased accordingly, showing that these models can roughly locate the opinion expressions to a certain degree.

Dataset comparison. Similar to the tasks like NER (Zhou et al., 2021), POS tagging, dependency parsing (Straka, 2018) and so on, in which English models have performed better than the Chinese, we see the same pattern in our OEI task. The exact matching F1 57.08 of the `Gold` corpus trained model still has a performance gap compared with that of the English MPQA dataset (i.e., 63.71 by a similar BERT-based model of Xia et al. (2021)). This may due to (1) the opinion boundaries in the word-based English MPQA are easier to locate than our character-based Chinese dataset; (2) the social media domain of our dataset, is more difficult than the news domain of MPQA.

[18] https://github.com/UKPLab/arxiv2018-bayesian-ensembles

Method comparison. First, we compare two annotator-agnostic methods (i.e., `All` and `MV`) with annotator-aware ones (i.e., the rest of models). As shown in Table 3, we can see that annotator-aware modeling is effective as a whole, bringing better performance on *exact* matching. In particular, our basic annotator-adapter model is able to give the best F1 among these selected baselines, demonstrating its advantage in crowdsourcing modeling. When the annotator-mixup is applied, the test scores are further boosted, showing the effectiveness of our annotator mixup. The overall tendencies of the two other metrics are similar by comparing our models with the others.

Our final performance is not only comparable to the `silver` corpus trained model, which we can take it as a weak upper-bound. but also close to the upper-bound model with expert annotations (i.e., `Gold`). Thus, our result for Chinese OEI is completely acceptable, demonstrating that crowdsourcing annotations are indeed with great value for model training. The observation indicates that crowdsourcing could be a highly-promising alternative to build a Chinese OEI system at a low cost.

4.3 Analysis

Here we conduct fine-grained analyses to better understand the task and these methods in-depth, where the evaluation by *exact* matching is used in this subsection. There are several additional analyses which are shown in the Appendix.

Performance by the opinion expression length. Intuitively, the identification of opinion expressions can be greatly affected by the length of the expressions, and longer expressions might be more challenging to be identified precisely. Figure 4 shows

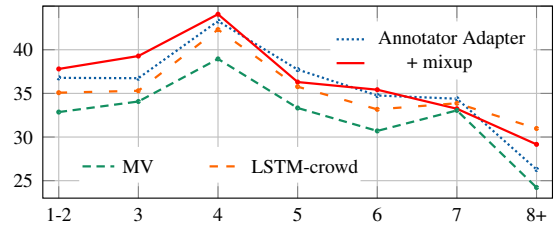

Figure 4: F1 scores of *exact* matching in terms of the opinion expression length. We bucket the opinion expressions into seven categories, where each category includes more than 100 opinion expressions.

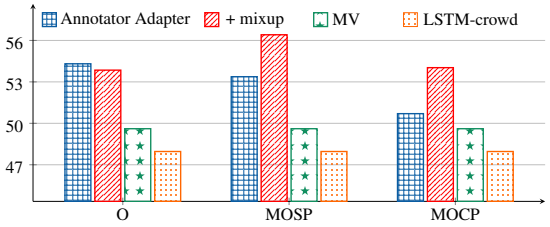

Figure 5: F1 scores of *exact* matching by following three category sentences: (1) one-opinion (**O**), (2) multiple-opinion single-polarity (**MOSP**), and (3) multiple-opinion contradict-polarity (**MOCP**).

Model	Exact P	Exact R	Exact F1
ALL	52.24	34.17	41.32
MV	43.79	38.70	41.09
LSTM-Crowd	46.57	38.19	41.97
LSTM-Crowd-cat	52.10	32.79	40.25
Annotator-Adapter	**55.81**	42.80	**48.45**
Annotator-Adapter + mixup	52.76	**43.68**	47.79

Table 4: The evaluation results on the crowd test set, i.e., we compute F1 scores between model predictions and crowd annotations. The ALL and MV have no modifications. The other annotator-aware models have replaced the expert vector with the specific annotator vector corresponding to the annotations when testing.

the F1 scores in terms of expression lengths by the four models we focused. We can see that the F1 score decreases dramatically when the expression length becomes larger than 4, which is consistent with our intuition. In addition, the annotator-adapter model is better than previous methods, and the mixup model can reach the best performance on almost all the categories, indicating the robustness of our annotator mixup.

Influence of the opinion number per sentence. One sentence may have more than one opinion expressions, where these opinions might be mutually helpful or bring increased ambiguities. It is interesting to study the model behaviors in terms of opinion numbers. Here we conduct experimental comparisons by dividing the test corpus into three categories: (1) only one opinion expression exists in a sentence; (2) at least two opinions exist, and they are of the same sentiment polarity; (3) both positive and negative opinion expressions exist. As shown in Figure 5, the sentences with multiple opinions of a consistent polarity can obtain the highest F1 score. The potential reason might be that the expressed opinions of these sentences are usually highly affirmative with strong sentiments, and the consistent expressions can be mutually helpful according to our assumption. For the other two categories, it seems that they are equally difficult according to the final scores. For all three categories, two annotator-adapter models demonstrate better performance than the others.

Self-evaluation of crowd annotators. The annotator adapter uses a pseudo expert embedding to predict opinion expressions and evaluate performance on the gold-standard annotations of experts. It is interesting to examine the self-evaluation performance on the crowd annotations of the test corpus as well. During the inference, we use the crowd annotators as inputs, and calculate the model performance on the corresponding crowd annotations.

Table 4 shows the results. First, two annotator-agnostic models (i.e., `ALL` and `MV`) have similar poor performance since they are trying to estimate the expert annotation function rather than learn crowd annotations. Second, the performance of two annotator-noise-modeling methods, `LSTM-Crowd` and `LSTM-Crowd-cat`, respectively, is close to the annotator-agnostic ones, showing that they are also incapable to model individual annotators. Then, our two annotator-adapter models achieve leading performance compared with all baseline methods, giving a significant gap (at least $47.79 - 41.97 = 5.82$ in F1). They are more capable of predicting crowd annotations, demonstrating the ability to model the annotators effectively. To our surprise, the mixup annotator-adapter model does not exceed the basic one, indicating that the mixed annotator embeddings in training could slightly hurt the modeling of individual annotators.

5 Related Work

OEI is one important task in opinion mining (Liu, 2012), and has received great interests (Breck et al.,

2007; İrsoy and Cardie, 2014; Xia et al., 2021). The early studies can be dated back to Wilson et al. (2005) and Breck et al. (2007), which exploit CRF-based methods for the task with manually-crafted features. SemiCRF is exploited next in order to exploit span-based features (Yang and Cardie, 2012). Recently, neural network models have attracted the most attention. İrsoy and Cardie (2014) present a deep bi-directional recurrent neural network (RNN) to identify opinion expressions. BiLSTM is also used in Katiyar and Cardie (2016) and Zhang et al. (2019), showing improved performance on OEI. Fan et al. (2019) design an Inward-LSTM to incorporate the opinion target information for identifying opinion expressions given their target, which can be seen as a special case of our task. Xia et al. (2021) employ pre-trained BERT representations (Devlin et al., 2019) to increase the identification performance of joint extraction of the opinion expression, holder and target by a span-based model.

All the above studies are in English and based on the MPQA (Wiebe et al., 2005), or customer reviews (Wang et al., 2016, 2017; Fan et al., 2019) since there are very few datasets available for other languages. Hence, we construct a large-scale Chinese corpus for this task by crowdsourcing, and borrow a novel representation learning model (Zhang et al., 2021) to handle the crowdsourcing annotations. In this work, we take the general BERT-BiLSTM-CRF architecture as the baseline, which is a competitive model for OEI task.

Crowdsourcing as a cheap way to collect a large-scale training corpus for supervised models has been gradually popular in practice (Snow et al., 2008; Callison-Burch and Dredze, 2010; Trautmann et al., 2020). A number of models are developed to aggregate a higher-quality corpus from the crowdsourcing corpus (Raykar et al., 2010; Rodrigues et al., 2014a,b; Moreno et al., 2015), aiming to reduce the gap over the expert-annotated corpus. Recently, modeling the bias between the crowd annotators and the oracle experts has been demonstrated effectively (Nguyen et al., 2017; Simpson and Gurevych, 2019; Li et al., 2020), focusing on the label bias between the crowdsourcing annotations and gold-standard answers, regarding crowdsourcing annotations as annotator-sensitive noises. Zhang et al. (2021) do not hold crowdsourcing annotations as noisy labels, while regard them as ground-truths by the understanding of individual crowd annotators. In this work, we follow the idea of Zhang et al. (2021) to explorate our crowdsourcing corpus, and further propose the annotator mixup to enhance the learning of the expert representation for the test stage.

6 Conclusion

We presented the first work of Chinese OEI by crowdsourcing, which is also the first crowdsourcing work of OEI. First, we constructed an extremely-noisy crowdsourcing corpus at a very low cost, and also built gold-standard dataset by experts for experimental evaluations. To verify the value of our low-cost and extremely-noisy corpus, we exploited the annotator-adapter model presented by Zhang et al. (2021) to fully explore the crowdsourcing annotations, and further proposed an annotator-mixup strategy to enhance the model. Experimental results show that the annotator-adapter can make the best use of our crowdsourcing corpus compared with several representative baselines, and the annotator-mixup strategy is also effective. Our final performance can reach an F-score of 54.59% by exact matching. This number is actually highly competitive by referring to the model trained on expert annotations (57.08%), which indicates that crowdsourcing can be highly recommendable to set up a Chinese OEI system fast and cheap, although the collected corpus is extremely noisy.

Ethical/Broader Impact

We construct a large-scale Chinese opinion expression identification dataset with crowd annotations. We access the original posts by manually traversing the relevant Weibo topics or searching the corresponding keywords, and then copy and anonymize the text contents. All posts we collected are open-access. In addition, we also anonymize all annotators and experts (only keep the ID for the research purpose). All annotators were properly paid by their actual efforts. This dataset can be used for both the Chinese opinion expression identification task as well as crowdsourcing sequence labeling.

Acknowledgments

We thank all reviewers for their hard work. This research is supported by grants from the National Key Research and Development Program of China (No. 2018YFC0832101) and the National Natural Science Foundation of China (No. 62176180).

References

Eric Breck, Yejin Choi, and Claire Cardie. 2007. Identifying expressions of opinion in context. In *Proc. of the IJCAI*, pages 2683–2688.

Chris Callison-Burch and Mark Dredze. 2010. Creating speech and language data with Amazon's Mechanical Turk. In *Proc. of the NAACL-HLT*, pages 1–12, Los Angeles. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of the NAACL-HLT*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhifang Fan, Zhen Wu, Xin-Yu Dai, Shujian Huang, and Jiajun Chen. 2019. Target-oriented opinion words extraction with target-fused neural sequence labeling. In *Proc. of the NAACL-HLT*, pages 2509–2518, Minneapolis, Minnesota. Association for Computational Linguistics.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for NLP. In *Proc. of the ICML*, volume 97 of *Proc. of Machine Learning Research*, pages 2790–2799. PMLR.

Ozan İrsoy and Claire Cardie. 2014. Opinion mining with deep recurrent neural networks. In *Proc. of the EMNLP*, pages 720–728, Doha, Qatar. Association for Computational Linguistics.

Chen Jia, Xiaobo Liang, and Yue Zhang. 2019. Cross-domain NER using cross-domain language modeling. In *Proc. of the ACL*, pages 2464–2474, Florence, Italy. Association for Computational Linguistics.

Arzoo Katiyar and Claire Cardie. 2016. Investigating LSTMs for joint extraction of opinion entities and relations. In *Proc. of the ACL*, pages 919–929, Berlin, Germany. Association for Computational Linguistics.

Maolin Li, Hiroya Takamura, and Sophia Ananiadou. 2020. A neural model for aggregating coreference annotation in crowdsourcing. In *Proc. of the COLING*, pages 5760–5773, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Bing Liu. 2012. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan & Claypool Publishers.

Bing Liu and Lei Zhang. 2012. A survey of opinion mining and sentiment analysis. In Charu C. Aggarwal and ChengXiang Zhai, editors, *Mining Text Data*, pages 415–463. Springer.

Pablo G. Moreno, Antonio Artés-Rodríguez, Yee Whye Teh, and Fernando Pérez-Cruz. 2015. Bayesian nonparametric crowdsourcing. *J. Mach. Learn. Res.*, 16:1607–1627.

An Thanh Nguyen, Byron Wallace, Junyi Jessy Li, Ani Nenkova, and Matthew Lease. 2017. Aggregating and predicting sequence labels from crowd annotations. In *Proc. of the ACL*, pages 299–309, Vancouver, Canada. Association for Computational Linguistics.

Emmanouil Antonios Platanios, Mrinmaya Sachan, Graham Neubig, and Tom Mitchell. 2018. Contextual parameter generation for universal neural machine translation. In *Proc. of the EMNLP*, pages 425–435, Brussels, Belgium. Association for Computational Linguistics.

Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. 2010. Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322.

Filipe Rodrigues, Francisco C. Pereira, and Bernardete Ribeiro. 2014a. Gaussian process classification and active learning with multiple annotators. In *Proc. of the ICML*, volume 32 of *JMLR Workshop and Conference Proceedings*, pages 433–441. JMLR.org.

Filipe Rodrigues, Francisco C. Pereira, and Bernardete Ribeiro. 2014b. Sequence labeling with multiple annotators. *Mach. Learn.*, 95(2):165–181.

Edwin Simpson and Iryna Gurevych. 2019. A Bayesian approach for sequence tagging with crowds. In *Proc. of the EMNLP-IJCNLP*, pages 1093–1104, Hong Kong, China. Association for Computational Linguistics.

Rion Snow, Brendan O'Connor, Daniel Jurafsky, and Andrew Ng. 2008. Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks. In *Proc. of the EMNLP*, pages 254–263, Honolulu, Hawaii. Association for Computational Linguistics.

Milan Straka. 2018. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proc. of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 197–207.

Lichao Sun, Congying Xia, Wenpeng Yin, Tingting Liang, Philip Yu, and Lifang He. 2020. Mixup-transformer: Dynamic data augmentation for NLP tasks. In *Proc. of the COLING*, pages 3436–3440, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Dietrich Trautmann, Johannes Daxenberger, Christian Stab, Hinrich Schütze, and Iryna Gurevych. 2020. Fine-grained argument unit recognition and classification. In *AAAI 2020*, pages 9048–9056. AAAI Press.

Ahmet Üstün, Arianna Bisazza, Gosse Bouma, and Gertjan van Noord. 2020. UDapter: Language adaptation for truly Universal Dependency parsing. In *Proc. of the EMNLP*, pages 2302–2315, Online. Association for Computational Linguistics.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2016. Recursive neural conditional random fields for aspect-based sentiment analysis. In *Proc. of the EMNLP*, pages 616–626, Austin, Texas. Association for Computational Linguistics.

Wenya Wang, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao. 2017. Coupled multi-layer attentions for co-extraction of aspect and opinion terms. In *Proc. of the AAAI*, pages 3316–3322. AAAI Press.

Janyce Wiebe, Theresa Wilson, and Claire Cardie. 2005. Annotating expressions of opinions and emotions in language. *Lang. Resour. Evaluation*, 39(2-3):165–210.

Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff, and Siddharth Patwardhan. 2005. OpinionFinder: A system for subjectivity analysis. In *Proc. of HLT/EMNLP 2005 Interactive Demonstrations*, pages 34–35, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Qingrong Xia, Bo Zhang, Rui Wang, Zhenghua Li, Yue Zhang, Fei Huang, Luo Si, and Min Zhang. 2021. A unified span-based approach for opinion mining with syntactic constituents. In *Proc. of the NAACL-HLT*, pages 1795–1804, Online. Association for Computational Linguistics.

Bishan Yang and Claire Cardie. 2012. Extracting opinion expressions with semi-Markov conditional random fields. In *Proc. of the EMNLP-CoNLL*, pages 1335–1345, Jeju Island, Korea. Association for Computational Linguistics.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. 2018. mixup: Beyond empirical risk minimization. In *Proc. of the ICLR*. OpenReview.net.

Meishan Zhang, Qiansheng Wang, and Guohong Fu. 2019. End-to-end neural opinion extraction with a transition-based model. *Information Systems*, 80:56 – 63.

Rongzhi Zhang, Yue Yu, and Chao Zhang. 2020. SeqMix: Augmenting active sequence labeling via sequence mixup. In *Proc. of the EMNLP*, pages 8566–8579, Online. Association for Computational Linguistics.

Xin Zhang, Guangwei Xu, Yueheng Sun, Meishan Zhang, and Pengjun Xie. 2021. Crowdsourcing learning as domain adaptation: A case study on named entity recognition. In *Proc. of the ACL-IJCNLP*, pages 5558–5570, Online. Association for Computational Linguistics.

Xuan Zhou, Xiao Zhang, Chenyang Tao, Junya Chen, Bing Xu, Wei Wang, and Jing Xiao. 2021. Multi-grained knowledge distillation for named entity recognition. In *Proc. of the NAACL*, pages 5704–5716.

Model	Exact F1	Prop F1	Binary F1
p			
4	46.00	60.95	65.02
6	**53.86**	**72.65**	77.55
8	53.39	72.54	**78.32**
10	53.21	72.55	78.08
12	52.82	71.04	74.52
α in Beta(α, α)			
0.2	54.86	72.92	78.19
0.5	55.15	**73.37**	**77.79**
0.8	**55.18**	73.00	77.48
1.0	54.78	72.35	76.98
Mixup Training Strategy			
One-Stage	**55.15**	73.37	77.79
Two-Stage	54.59	**74.45**	**78.49**
Fine-tuning Based Models			
ALL	52.35	69.58	76.99
MV	47.52	69.07	76.09
Silver	**54.47**	**73.16**	**79.99**
(2017)	53.17	70.81	77.23
(2017)-cat	53.01	70.03	76.95

Table 5: Experimental results of various settings.

A Annotation Guideline

In this annotation task, we will give a number of sentences that have a high probability of expressing positive or negative sentiment, and your goal is to label the words that expresses these sentiments in each sentence. An intuitive criteria for determining whether words are expressing sentiment is that if these words are replaced, the sentiment expressed by the sentence will also change. Sentimental words will not usually be names of people, places, time or pronouns, etc. It is important to note that (1) you need to carefully understand the emotion expressed by the sentence, not judge it according to your own values, and (2) the labeled words usually do not include the target of the sentiment, such as pronouns, names of people, etc., which are generally not affected by the replacement of these words.

B Hyper-parameter Tuning

We also implement the baseline models in the fine-tuning style, results (in Table 5) show that the adapter-based models are comparable and parameter-efficient.

PGN Adapter Layers First, we examine the influence of PGN adapter layers mentioned in §3.2 by p, which is a hyper-parameter in our annotator-adapter. As shown in Table 5, we can see that the performance is stable between $p \in [6, 8, 10]$. After considering both the parameter scale and the capability of our model, we set $p = 6$ for a trade-off.

Annotator Mixup The mixup includes a hyper-parameter α to control the interpolation by the distribution $\mathrm{Beta}(\alpha, \alpha)$. Here we show the influence of α by setting it with 0.2, 0.5, 0.8, and 1.0. We find that the model performance has no significant differences between these values, as shown in Table 5. To train our mixup model, we also have a reasonable small trick: training the mixup model in two stages. First, the model is trained only with the original corpus. When the model achieves the best performance on the devset, we begin the second-stage training by using the original corpus as well as the augmented corpus. Their performance difference is shown in Table 5, which indicates that the two-stage training is important for our mixup model.

C Expert-Evaluation of Crowd Annotators

We evaluate the performance of each learned annotator of three annotator-aware models towards the expert's view. The goal is achieved by using the individual annotator embeddings as input to obtain the output predicted by this specific annotator, and then measure the output performance based on the gold-standard test corpus. Table 7 shows the results. There is a huge discrepancy between the scores of different learned annotators of `LSTM-Crowd` or `annotator-adapter`, demonstrating annotators have different abilities in predicting gold labels. This is mainly because the annotators have different abilities meanwhile the annotations they gave have different qualities. All annotators in the annotator-adapter model are unable to outperform the expert (centroid point), verifying that the estimated expert is strong and reasonable. In addition, the learned annotators of our mixup model have closer performances since the annotator-mixup change the learning objective from modeling annotators to modeling the expert, which can further boost the performance of the estimated expert.

D Case Study

For a more intuitive understanding of our task and various models, we offer a paradigmatic example from the test set to analyze their outputs. Table 6 shows the gold annotation and model predictions. As shown, the `ALL` method can correctly recognize all three opinions, but fails to predict the correct boundaries. The `MV` method splits one opinion into two, and is able to recall one full opinion expression exactly. The `LSTM-Crowd` is similar to `ALL` yet slightly better. Both the annotator-adapter and our mixup models can obtain better results for this example. Note that all three opinions are difficult to be fully recognized even by crowd annotators.

Model	Text and Opinions
Gold	现在驱车在这清冷寂寥的街路上，这些热闹的闪亮的灯光倒让人有心安的感觉。
	Now driving on this cold and lonely street, these lively and shiny lights make me ease.
ALL	现在驱车在这清冷寂寥的街路上，这些热闹的闪亮的灯光倒让人有心安的感觉。
	Now driving on this cold and lonely street, these lively and shiny lights make me ease.
MV	现在驱车在这清冷寂寥的街路上，这些热闹的闪亮的灯光倒让人有心安的感觉。
	Now driving on this cold and lonely street, these lively and shiny lights make me ease.
LSTM-Crowd	现在驱车在这清冷寂寥的街路上，这些热闹的闪亮的灯光倒让人有心安的感觉。
	Now driving on this cold and lonely street, these lively and shiny lights make me ease.
Our Vanilla	现在驱车在这清冷寂寥的街路上，这些热闹的闪亮的灯光倒让人有心安的感觉。
	Now driving on this cold and lonely street, these lively and shiny lights make me ease.
Our Final	现在驱车在这清冷寂寥的街路上，这些热闹的闪亮的灯光倒让人有心安的感觉。
	Now driving on this cold and lonely street, these lively and shiny lights make me ease.

Table 6: Case Study. The blue rectangles and red boxes with round corners are negative and positive, respectively.

Annotator ID	LSTM-Crowd	Annotator-Adapter	+ mixup	Annotator ID	LSTM-Crowd	Annotator-Adapter	+ mixup	Annotator ID	LSTM-Crowd	Annotator-Adapter	+ mixup
0	50.76	47.31	54.63	24	28.45	11.99	44.04	48	40.27	40.52	55.27
1	44.02	40.05	51.84	25	51.69	47.50	55.56	49	47.28	50.45	54.80
2	53.30	48.20	55.18	26	49.05	40.18	53.26	50	51.27	47.89	52.72
3	38.63	13.01	45.14	27	51.58	48.08	54.19	51	50.86	49.45	54.42
4	43.37	29.78	55.22	28	51.69	38.10	48.99	52	54.92	40.82	49.88
5	55.02	47.95	53.84	29	51.46	46.31	55.06	53	47.63	31.20	52.81
6	45.02	46.13	54.66	30	45.30	33.83	55.20	54	49.60	43.54	54.85
7	52.93	43.56	55.60	31	46.19	44.14	49.29	55	54.88	41.97	55.44
8	35.40	22.55	46.86	32	50.02	41.63	53.52	56	55.98	52.35	56.12
9	46.61	37.30	54.58	33	36.78	40.17	54.43	57	53.56	44.90	53.13
10	50.33	45.37	54.76	34	39.01	34.48	52.80	58	45.19	31.42	48.81
11	49.98	48.87	54.17	35	48.17	49.09	52.66	59	53.09	43.95	53.65
12	53.90	48.53	55.69	36	54.45	47.14	56.18	60	35.27	13.13	52.97
13	54.51	49.11	54.88	37	53.32	43.87	54.44	61	52.46	34.26	54.79
14	49.86	48.65	53.08	38	51.08	43.25	52.08	62	41.95	38.49	51.39
15	41.64	32.81	49.25	39	42.33	31.08	52.68	63	35.73	43.76	54.10
16	53.51	41.33	53.95	40	46.63	42.81	53.46	64	52.56	40.93	52.93
17	50.11	34.24	52.71	41	46.50	40.38	53.45	65	48.70	34.95	51.83
18	52.80	41.83	54.98	42	50.31	44.68	51.85	66	46.21	30.29	52.67
19	42.29	35.71	51.46	43	54.73	48.57	51.47	67	46.24	33.75	49.53
20	51.38	47.30	52.00	44	47.34	31.86	52.75	68	35.36	15.34	50.08
21	35.39	37.10	47.02	45	46.83	28.98	54.59	69	32.06	22.67	52.54
22	52.62	43.67	53.10	46	54.26	40.30	52.05	Expert	53.35	53.86	54.59
23	53.49	47.08	54.62	47	49.73	41.55	54.41				

Table 7: The F1 scores by using different crowd annotators as input on the gold testset. Exact matching scores are reported. The LSTM-Crowd just learns an estimation of expert assisted by modeling the label bias of annotators, while the annotator-adapter model learns the different understandings of each annotator but not the expert annotations. Our final mixup model is much more stable across different annotators. The observation indicates that, with the application of annotator-mixup, all annotators can learn from each other and improve towards the expert level together, which can enhance the expert-modeling.