# Meta-based Self-training and Re-weighting for Aspect-based Sentiment Analysis

Kai He, Rui Mao, Tieliang Gong, Chen Li, *Member, IEEE* and Erik Cambria*, *Fellow, IEEE*

**Abstract**—Aspect-based sentiment analysis (ABSA) means to identify fine-grained aspects, opinions, and sentiment polarities. Recent ABSA research focuses on utilizing multi-task learning (MTL) to achieve less computational costs and better performance. However, there are certain limits in MTL-based ABSA. For example, unbalanced labels and sub-task learning difficulties may result in the biases that some labels and sub-tasks are overfitting, while the others are underfitting. To address these issues, inspired by neuro-symbolic learning systems, we propose a meta-based self-training method with a meta-weighter (MSM). We believe that a generalizable model can be achieved by appropriate symbolic representation selection (in-domain knowledge) and effective learning control (regulation) in a neural system. Thus, MSM trains a teacher model to generate in-domain knowledge (e.g., unlabeled data selection and pseudo-label generation), where the generated pseudo-labels are used by a student model for supervised learning. Then, the meta-weighter of MSM is jointly trained with the student model to provide each instance with sub-task-specific weights to coordinate their convergence rates, balancing class labels, and alleviating noise impacts introduced from self-training. The following experiments indicate that MSM can utilize 50% labeled data to achieve comparable results to state-of-arts models in ABSA and outperform them with all labeled data.

**Index Terms**—Aspect-based sentiment analysis, Meta learning, Self-training

## 1 Introduction

Aspect-based sentiment analysis (ABSA) is an entity-oriented and fine-grained sentiment analysis task. The task consists of three sub-tasks, including aspect term extraction (AE), opinion extraction (OE), and aspect-level sentiment classification (SC) [1]. As shown in Fig. 1, given a sentence "The restaurant is crowded but with efficient and accurate service", ABSA aims to collect triplets {crowded, restaurant, negative} and {(efficient, accurate), service, positive}. Such fine-grained ABSA can help recognize, interpret, and simulate human emotions better and further support many downstream applications, such as precise recommendation systems and intelligent dialogue systems [2], [3], [4].

Typically, AE and SC sub-tasks were respectively regarded as a two-step pipelined task [5], [6], and OE was performed separately [7], [8]. Considering the requirements of end-to-end processing and a lighter model in real-world application scenarios, these three tasks were also formulated as a multi-task learning (MTL) issue. Using MTL can simultaneously obtain aspects, their polarities, and opinions with a shared encoder, which can save an amount of computing resource and gain performance-boosting improvements from the learnt knowledge of each sub-task [9]. However, MTL-based ABSA faces additional challenges, compared with single-task learning solutions, e.g., various sub-tasks annotated in an instance may cause an imbalanced label distribution. It is hard to sample sufficient instances with equal numbers of labels for each sub-task in an MTL setting.

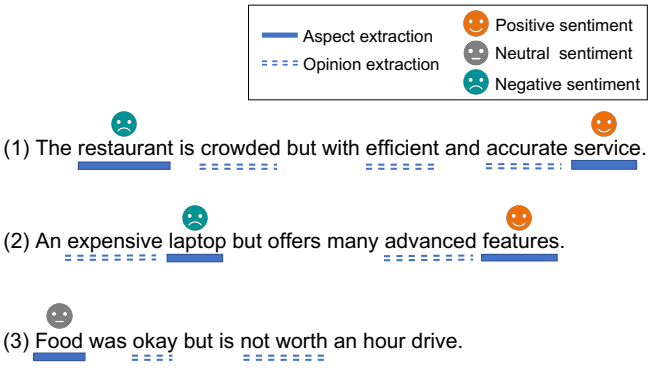



Fig. 1. The examples of aspect-based sentiment analysis (ABSA) task, which consists of aspect extraction (AE), opinion extraction (OE), and aspect-level sentiment classification (SC).

This problem becomes more obvious in a sequence labeling task in which the class of "others" dominates its tag sets. Some label instances in a sub-task may be insufficient to train a robust neural network. Taking a utilized ABSA corpus [10] in this paper as an example, the number of instances with the most common label versus the most uncommon one reaches 760:1. With a limited dataset, such extremely imbalanced data may result that some classes insufficiently support an MTL neural network to play its functions properly.

To address the above issue, inspired by neuro-symbolic learning systems, self-training communities seek to a robust method to generate in-domain knowledge, e.g., selecting confident input (symbolic representations) and generating pseudo labels, to mitigate the impacts of insufficient and imbalanced data [11], [12], [13]. Such a method can enhance the predictive ability of a model by training with more data that are automatically labeled by the model itself.

* *Corresponding author: Erik Cambria.*

- *Kai He, Teiliang Gong, Chen Li are with the School of Computer Science and Technology, Xi'an Jiaotong University, Shanxi, China, 710049. E-mail: hk52025804@stu.xjtu.edu.cn, {gongtl, cli}@xjtu.edu.cn.*
- *Rui Mao and Erik Cambria are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore, 639798. E-mail: {rui.mao, cambria}@ntu.edu.sg.*

However, employing self-training inevitably introduces noisy pseudo labels [14], [15], which may cause the problem of gradual drift [16]. To figure out this problem, a direct and typical method is re-weighting [17]. When obtaining the generated pseudo labels, some self-training methods utilized uncertainty and utility thresholds to evaluate the quality of the pseudo-labeled data [18], [19], and allocated small weights for low-quality pseudo-labels. When applying self-training technology in an MTL and sequence-labeling ABSA, re-weighting becomes more challenging in the following aspects. First, the generated pseudo-labeled data need smaller weights to alleviate noise influences. Second, classes with fewer annotated instances should be allocated with larger weights to help the model learn from insufficient data [20], [21]. Lastly, considering the fact that different sub-tasks have different convergence rates and importance (main tasks vs. auxiliary tasks), the sub-tasks may need different weights to be coordinated comprehensively. Typical re-weighting research [22], [23] likely focused on one of the above conditions. They manually designed a pre-defined function to re-weight data, solving the problem of either imbalanced distributions or denoising. It is hard to manually design a pre-defined weighting function to satisfy these three requirements simultaneously, and working well under different conditions adaptively.

To deal with the above problems, e.g., insufficient and imbalanced data, and sub-task re-weighing for ABSA, this paper proposes a **M**eta-based **S**elf-training method with **M**ix-weighter (MSM). MSM employs self-training to obtain more training data and automatically calculates different weights for multiple tasks under different conditions. As shown in Fig. 2, MSM consists of three components: a teacher model, a student model, and an extra meta-weighter. The teacher model owns the same structure as the student model, while the teacher model is only trained by gold-labeled data to avoid the influences of pseudo labels. The student model is trained with both gold and pseudo labels that are re-weighted by the meta-weighter, so that the student model can obtain a fully and unbiased supervision. The meta-weighter provides sub-task-specific weights of mixed labeled and unlabeled instances for the student model. Besides, to ensure the meta-weighter can generate proper weights, we employ a three-step meta updating method for jointly training the meta-weighter and student model of MSM. By maintaining two sets of parameters, the meta-weighter can employ current feedback from the student model to generate weights for current inputs. The meta-weighter offers the student model a better convergence, because they are already updated by the same input. Finally, the student model provides final predictions under the supervision of the teacher model and meta-weighter.

The main contributions of this work are as follows:

- We propose MSM, which achieves self-training by employing a typical teacher-student framework plus a novel meta-weighter. The meta-weighter can generate sub-task-specific weights to mitigate noise impacts, coordinating sub-tasks, and balancing class labels in ABSA.
- We design a three-step meta training method to update the meta-weighter and student model jointly. Utilizing the proposed method, MSM can take advantage of current feedback from the student model to lead MSM to a temperature gradient direction.
- We utilize 50% training data to achieve comparable performance of state-of-art models and outperform them with all labeled data in ABSA tasks. The experimental results demonstrate the effectiveness of the proposed MSM.

## 2 Related Work

Multi-task learning is a popular technology in recent ABSA studies [24], [25], [26]. Compared with pipeline methods, MTL-based solutions simultaneously deal with AE, OE, and SC rather than extracting aspect terms first then identifying sentiment polarities. The most recent research focus on how to enhance the interactions among sub-tasks. GBM [24] proposed a novel Gated Bridging Mechanism for sharing useful information between different sub-tasks, while filtering out useless information. IMN [25] introduced a message passing mechanism among different tasks through a shared set of latent variables, which jointly learnt multiple related tasks at both the token level and the document level. RACL [9] proposed a MTL method to encode collaborative signals between different sub-tasks in a stacked multi-layer network.

However, besides enhancing information interactions, unique data features and requirements should also be paid attention in MTL-based sequence labeling, such as ABSA. Considering the potentially insufficient and imbalanced data features, self-training is a promising technology to mitigate the problems. Generating pseudo labels is a welcome method in self-learning. Flipping, rotating, and translating are common operations in computer vision field to generate pseudo-labeled instances [14], [27], [28], [29]. Generating pseudo-labeled data is more challenging in the natural language processing (NLP) field, because of the ambiguity of language [30], [31]. Randomly deleting, inserting, and switching some tokens in a sentence [32] may cause semantic incoherence and side effects for token-level sequence labeling tasks. Hence, self-labeling is a more effective method in NLP tasks [19]. SFLM [33] proposed a prompt-based learner with self-training for few-shot tasks. Given two views of an instance via weak and strong augmentations, SFLM generated a pseudo label on the weakly augmented version and fine-tuned with the strongly augmented version. Du et al. [34] introduced SentAugment to generate task-specific data for a given task from a bank of billions of unlabeled sentences and utilized a supervised teacher model to label these sentences. The work of Wang et al. [35] is similar to ours, which focused on addressing the label scarcity challenge of sequence labeling tasks with self-training. However, they just explored a single-task setting while we try to deal with multiple sequence labeling tasks.

One challenge for self-training is that pseudo labels are noisy, making it hard for the neural model to achieve effective convergence. To alleviate impacts from noise, the generated pseudo labels need to be cherry-picked. Some early work simply utilized the predicted softmax score [36], [37]. In addition, a part of recent active learning works [38], [39] also depended on the model output scores to select

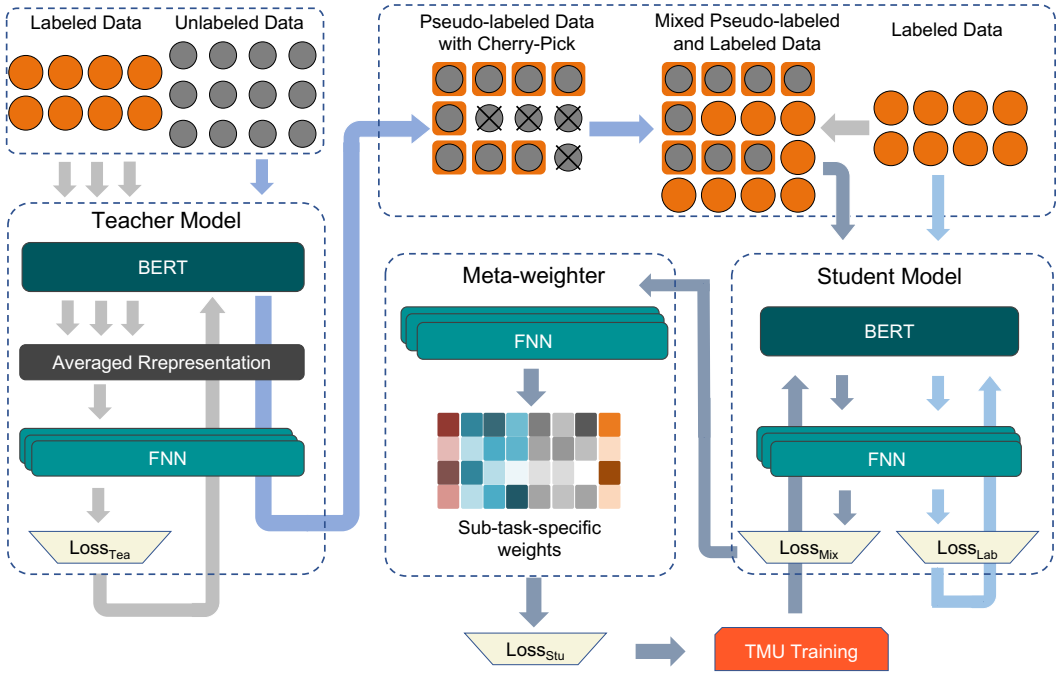



Fig. 2. The overview of the proposed meta-based self-training method with a meta-weighter (MSM). Different data stream is exhibited with different colors. The details of TMU training are illustrated in Fig. 3.

samples. However, a prediction with a high softmax score cannot be fully trusted. This is because these models, trained for generating pseudo labels, are limited with their generalization ability and the number of training data. Thus, they are likely to confidently allocate wrong labels [40]. For such a reason, some studies [41], [42] leveraged curriculum learning to select data instances from simple to complex. Some research about self-paced learning [43], [44] calculated data uncertainty as the criterion for data selection. In this work, our model takes both data uncertainty and utility into account for pseudo-labeled data selection.

## 3 METHODOLOGY

This section formulates the ABSA task, introduces the overview of MSM, and describes the training method of meta-weigher in detail. As shown in Fig. 2, the proposed MSM consists of three components: a teacher model, a meta-weighter, and a student model. The teacher model adopts gold data to learn the task, generate pseudo labels, and calculate the uncertainty for unlabeled data. The meta-weighter is jointly trained with the student model to allocate sub-task-specific weights for gold and pseudo data. Guided by the teacher model and the meta-weighter, the student model is trained with gold and pseudo labels, and then performs the final interference.

MSM utilizes two independent teacher model and student model, so that the teacher can only be trained by gold data to avoid noise impacts and generate high-quality pseudo labels. The student model can adopt more training data without extra manual annotations to mitigate imbalanced and insufficient data influences. Considering that automatically labeled data are noisy, necessary measures can be used to prevent gradual drift in the student model. To this end, we design the meta-weighter to provide sub-task-specific weights. Different from using predefined weight functions by human experts [23], [45], the meta-weighter considers feedback from the student model and can be dynamically adjusted during training. Intuitively, this is similar to curriculum learning [46], which instructs the student model to be trained with a learnable input order and also recognizes pseudo and gold data.

TABLE 1
The comparison of different existing related work. AOPE means Aspect-Opinion Pair Extraction [47]. ACSA means Aspect-Category Sentiment Analysis [48]. TASD means Target Aspect Sentiment Detection [49]. ABSA and ABSA[1] [50] both means Aspect-based Sentiment Analysis. Compared with ABSA[1], our ABSA task also extracts opinion. The symbol [a] refers to aspect, [c] refers to category, [o] refers to opinion, and [s] refers to sentiment.

| Input: An expensive laptop but offers many advanced features. | |
|---|---|
| Task Name | Output |
| AOPE | (laptop [a], expensive [o]), (features [a], advanced[o]) |
| ACSA | (Product [c], positive [s]), (Attribute [c], negative[s]) |
| TASD | (laptop [a], product [c], positive [s]) |
| ABSA[1] | (laptop [a], positive [s]), (features [a], negative [s]) |
| ABSA | (laptop [a], expensive [o], positive [s]), (features [a], advanced [o], negative [s]) |

### 3.1 Task Definition

ABSA is a standard MTL issue, which consists of aspect extraction (AE), opinion extraction (OE), and sentiment classification (SC) sub-tasks. Each subtask is formulated as a sequence tagging task, following the baseline studies [5], [9], [24], [25]. Table 1 shows the comparison of different existing related work. Specifically, given an input sentence with $L$ tokens, MSM aims to predict a label sequence for

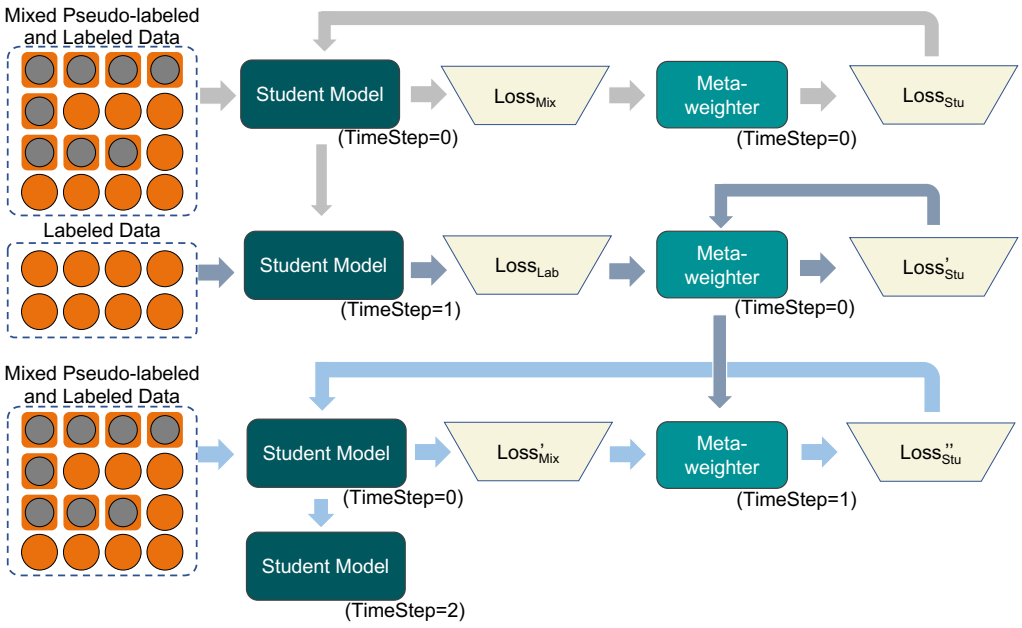



Fig. 3. Joint training meta-weighter with the student model, based on a three-step meta updating (TMU). This figure illustrates how MSM updates the student model and meta-weighter from a time step $t$=0 to the next step $t$=1. This figure corresponds to the TMU training in Fig. 2.

TABLE 2
The Example of the BIO tagging scheme in ABSA, where "B" means the beginning, "I" means the inner, and "O" means the other.

| Sentence: | The | restaurant | is | very | crowded | but | with | efficient | and | accurate | service | . |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| AE: | O | B | O | O | O | O | O | O | O | O | B | O |
| OE: | O | O | O | B | I | O | O | B | O | O | O | O |
| SC: | O | neg | O | O | O | O | O | O | O | O | pos | O |

each sub-task. The BIO schema is employed for AE and OE subtasks, where "B" means the beginning, "I" stands the inner, and "O" represents the other. For the SC subtask, MSM employs the label set $\{pos, neg, neu, O\}$, representing positive, negative, neutral sentiment polarities and others, respectively. The label "O" in the SC subtask means this token is not an aspect term and does not have a sentiment polarity. Table 2 shows the example of the utilized tagging scheme in this paper.

### 3.2 Learning to self-train

This section aims to introduce the self-training method in MSM in detail. The self-training in MSM consists of four steps, namely initializing the teacher model, generating pseudo labels, initializing the student model, and self-training the student model with a meta-weighter (see Fig. 2).

**Initializing the teacher model.** In the first step, the teacher model is supervised with a labeled data set $D_{lab}$ for initialization (see Eq. 1 and Eq. 2). There is no unlabeled data taking part in this training process to minimize the impact of data noise from unlabeled data on MSM.

$$H_S = BERT_{Tea}(w_1, ..., w_L) \tag{1}$$

$$\hat{Y}_S^t = Argmax(Softmax(FNN_{Tea}^t(H_S))), \tag{2}$$

where $w_1, ..., w_L$ is an input sentence $S$ with $L$ tokens. Task $t \in \{AE, OE, SC\}$. $H_S$ is the hidden representation of $S \in D_{lab}$. FNN is a two-layer feedforward neural network with ReLU activation function [51]. $BERT_{Tea}$ is the encoder of the teacher model. For the fair comparison with studies [5], [9], [24], [25], the teacher model and the student model both utilize BERT-large [52] as encoders.

$Loss_{Tea}$ is used to update the teacher model, which is computed as averaged cross entropy $CE$ on the three tasks.

$$Loss_{Tea} = \frac{1}{||tasks||} \sum_{t \in tasks} \sum_{w_i \in S} CE\left(\hat{Y}_{w_i}^t, Y_{w_i}^t\right). \tag{3}$$

$\hat{Y}_S^t$ is a predicted label sequence, where $\hat{Y}_{w_i}^t \in \hat{Y}_S^t$; $Y_S^t$ is a gold label sequence, where $Y_{w_i}^t \in Y_S^t$.

**Generating pseudo labels.** In the second step, MSM utilizes the initialized teacher model to generate pseudo labels for unlabeled data $D_{unl}$, calculating data uncertainty and modeling utility to cherry-pick these generated labels. $D_{unl}$ are taken from the original dataset without annotated labels.

Specifically, for each unlabeled sentence $S_{unl}$, given $N_{aug}$ forward passes through the teacher model with dropout (stochasticity can be introduced by dropout), and each pass with the same model parameters, MSM generates the averaged representations of these unlabeled data as data augmentation. Then, an averaged representation is mapped into a soft pseudo label for each token in a sub-task $t$.

$$logits_{S,pseu}^t = \frac{1}{N_{aug}} \sum_{i \in N_{aug}} FNN_{Tea}^t(H_{S,i}) \tag{4}$$

$$\hat{Y}^t_{S,pseu} = Softmax(logits^t_{S,pseu})), \quad (5)$$

where $H_{S,i}$ denotes the Bert hidden states of a sentence $S$ from the $i$th forward pass in the teacher model.

MSM utilize Monte-Carlo dropout [53] to calculate data uncertainty $unc$ in Eq. 6, based on soft pseudo labels $\hat{Y}^t_{pseu}$

$$unc^t_{pseu} = \frac{\hat{Y}^t_{S,pseu} * \log \hat{Y}^t_{S,pseu}}{\sum_{S' \in D_{unl}} \hat{Y}^t_{S',pseu} * \log \hat{Y}^t_{S',pseu}}. \quad (6)$$

The motivation of Eq. 6 is that if we can use the same model to predict the same sample multiple times and the predicted values are different, the entropy from these multiple tries can be calculated to estimate the uncertainty for this sample.

The final data uncertainty $unc_{pseu}$ for an instance is averaged over different tasks $t$. Higher data uncertainty indicates that the instances are hard samples. If MSM delivers too many easy samples for the self-training student model, it may not contribute helpful information. However, very high uncertainty may suggest these data instances are noisy. For such a reason, MSM only utilizes pseudo-labeled data with uncertainty in a range from $\mathcal{U}\%$ to $\mathcal{L}\%$, where $\mathcal{U}$ and $\mathcal{L}$ are hyperparameters (see Table 4). The pseudo-label data filtered by uncertainties is denoted as $D^{unc}_{pseu}$.

Additionally, we define $util_{pseu}$ to measure the prediction utility of the teacher model for unlabeled data. The difference between $util_{pseu}$ and $unc_{pseu}$ is that $unc_{pseu}$ is an instance-level measure and $util_{pseu}$ is token-level. $unc_{pseu}$ aims to select more reliable and confident sentences, while $util_{pseu}$ is for sampling the sentences whose label sequences have more non-"other" tags. $util_{pseu}$ is given by

$$util_{pseu} = \frac{1}{||tags'||} \sum_{tag \in tags'} Softmax(logits^{tag}_{S,pseu}), \quad (7)$$

where $tags'$ is the set of all non "other" tags from AE, OE, and SC. Considering the imbalanced label distributions of ABSA, MSM selects instances of which $util > \beta^{tag}$ to obtain more non "other" tags, forming pseudo-labeled data set $D^{util}_{pseu}$. $\beta^{tag}$ is a threshold hyperparameter (see Table 4). The final selected pseudo-labeled data set $D_{pseu} = D^{util}_{pseu} \cap D^{unc}_{pseu}$. With appropriate uncertainty and utility, MSM can cherry-pick pseudo-labeled data to maximize the information gain for the student model without introducing too much noise.

**Initializing the student model.** In the third step, the student model is initialized with $D_{lab}$, which is similar to step 1. An initialized student model helps avoid gradual drift when the student model conducts self-training with pseudo-labeled data.

**Self-training the student model with a meta-weighter** In the fourth step, MSM first combines the selected pseudo-labeled data set $D_{pseu}$ and $D_{lab}$ to form $D_{mix}$. $D_{mix}$ is employed to perform self-training for the student model.

The sentence $S' \in D_{mix}$ inputs to the Bert encoder of the student model to obtain the representation $H_{S'}$ as the same way as Eq. 1, and then get prediction labels $\hat{Y}^t_{S'}$ as Eq. 2. Then, $Loss^t_{Mix}$ is calculated by

$$Loss^t_{Mix} = CE(\hat{Y}^t_{S'_{Mix}}, Y^t_{S'_{Mix}}), \quad (8)$$

where $\hat{Y}^t_{S'_{Mix}}$ is a predicted label sequence. $Y^t_{S'_{Mix}}$ is a gold or pseudo label sequence. Noticeably, $Loss^t_{Mix}$ is a loss sequence with the length $L$ (same to the length of input). MSM reserves this loss sequence as the input of the meta-weighter for computing the sub-task-specific weights.

**Algorithm 1** The algorithm for self-learning with MSM.

**Input:**
A teacher model; a student model; a meta-weighter; max training iterations $Epoch$; a labeled data set $D_{lab}$; a unlabeled data set $D_{unl}$;

**Output:**
The predictions on a validation set.

**for** i= 0 to $Epoch$ **do**
  initializing the teacher model with $D_{lab}$ as Eq. 3.
  generating pseudo labels for $D_{unl}$ with the initialized teacher model.
  mixing the $D_{lab}$ and pseudo-labeled $D_{unl}$ to form $D_{mix}$.
  initializing the student model with $D_{lab}$.
  training the student model and meta-weighter with $D_{mix}$ by TMU method (see Algorithm 2).
  inference on the validation set.
**end for**

Next, $Loss^t_{Mix}$ are regarded as parts of input for meta-weighter to calculate the sub-task-specific and time-varying weights $W^t$.

$$W^t = Meta\text{-}weighter([Loss^t_{Mix}; H_{lab}; H_{epo}; H_{S'}]), \quad (9)$$

where $Meta - weighter$ is a two-layer forward neural network. [;] denotes concatenation.

$H_{lab}$ and $H_{epo}$ are corresponding label and epoch embeddings, which are encoded by two separate embedding layers (fully connected layers). Each label is allocated with a one-hot vector, and the one-hot vector is inputted into the related embedding layer to obtain $H_{lab}$, which represents the related class. $H_{epo}$ indicates the learning progress of the student model, which is normalized as an integer between 1 and 100. Similarly, this integer also is mapping into a one-hot vector and input into related embedding layer to obtain $H_{epo}$. For clarity, we omit $H_{lab}$, $H_{epo}$, and $H_{S'}$ in the inputs of meta-weighter in Fig. 2 and Fig. 3.

Then, $Loss^t_{Mix}$ of each sub-task will be weighted by $W^t$ to obtain final $Loss_{Stu}$.

$$Loss_{Stu} = \frac{1}{L} \sum_{L} W^t \cdot Loss^t_{Mix} \quad (10)$$

The details of updating the student model with $Loss_{Stu}$ are described in the next section. Summarily, the whole training of MSM is illustrated in Algorithm 1.

### 3.3 A three-step meta updating for jointly training the meta-weighter and student model

Inspired by the studies from [17], [22], [55], this section describes a three-step meta updating (TMU) method for jointly training the meta-weighter and the student model. The workflow can be seen in Fig. 3. This method allows MSM to take advantage of current feedback ($Loss_{Stu}$) from the student model to weight corresponding data instances. The generated weights can both alleviate the impacts of noise in $D_{mix}$ and coordinate three sub-tasks in ABSA.

**Algorithm 2** A three-step meta updating method for the meta-weighter and the student model (TMU)

**Input:**

An initialized student model; a meta-weighter; a mixed label and pseudo-labeled data set $D_{mix}$; a labeled data set $D_{lab}$;

**Output:**

The updated parameters of the student model (in time step 2) and meta-weighter (in time step 1).

1: computing weighted $Loss_{Stu}$ with $D_{mix}$ by Eqs. 9 and 10.
2: updating the student model from time step 0 to time step 1 with $Loss_{Stu}$ by Eq. 11.
3: updating the meta-weighter from time step 0 to time step 1 with $D_{lab}$, based on the student model in time step 1.
4: updating the student model from time step 0 to time step 2 with $D_{mix}$, based on the meta-weighter in time step 1.

TABLE 3
The statistics of datasets. # indicates the number; % indicates the percentage; ap. represents aspect labels; op. represents opinion labels; Tags ratio means the percentage of different tags in each sub-task. For sub-task AE and OE, it represents the percentage of tag "B", "I" and "O" for all tags. For sub-task SC, it represents the ratio of "positive", "negative", "neural" and "O" for all tags, where only integers are reserved for clear.

| Dataset | #number | Train | Valid | Test |
|---|---|---|---|---|
| Res14 | # of sent. | 2,436 | 608 | 800 |
| | # of tok. | 35,545 | 8,779 | 11,825 |
| | % of ap. | 11.82 | 10.97 | 14.02 |
| | % of op. | 8.15 | 8.36 | 9.21 |
| | % Tags ratio (AE) | 8/3/88 | 8/2/89 | 9/4/85 |
| | % Tags ratio (OE) | 8/1/91 | 8/1/9 | 9/1/90 |
| | % Tags ratio (SC) | 7/2/2/88 | 7/2/2/89 | 9/2/3/86 |
| Lap14 | # of sent. | 2,439 | 609 | 800 |
| | # of tok. | 38,675 | 9,670 | 11,007 |
| | % of ap. | 7.30 | 7.34 | 9.78 |
| | % of op. | 5.95 | 6.12 | 6.57 |
| | % Tags ratio (AE) | 5/2/93 | 5/3/93 | 6/4/90 |
| | % Tags ratio (OE) | 5/1/94 | 5/1/94 | 6/1/93 |
| | % Tags ratio (SC) | 3/3/2/92 | 3/3/1/93 | 5/2/3/90 |
| Res15 | # of sent. | 1052 | 263 | 685 |
| | # of tok. | 13875 | 3385 | 9831 |
| | % of ap. | 10.04 | 10.04 | 8.18 |
| | % of op. | 7.67 | 7.56 | 5.56 |
| | % Tags ratio (AE) | 7/3/90 | 7/3/90 | 6/3/92 |
| | % Tags ratio (OE) | 7/1/92 | 7/1/92 | 5/1/94 |
| | % Tags ratio (SC) | 9/2/1/88 | 9/3/0/88 | 6/4/0/90 |

According to loss sequence $Loss_{Stu}$ in Eq. 10, the student model will be updated once from $ts$=0 to $ts$=1 following

$$\Theta^1_{Stu} = \Theta^0_{Stu} - \lambda_{Stu}\nabla_{\Theta_{Stu}}\alpha_m Loss_{stu}, \quad (11)$$

where $ts$ corresponds to $TimeStep$ in Fig. 3. $\alpha_m$ is a hyper-parameter. $\lambda_{Stu}$ is the learning rate for the student model. Noticeably, the meta-weighter is not updated at this step.

Next, the student model utilizes data with gold-labels to calculate $Loss_{Lab}$ in Fig. 3. It helps the meta-weighter lead the student model to the appropriate gradient direction. With the updated student model parameter $\Theta^1_{Stu}$ and gold labeled data, $Loss_{Lab}$ is calculated following Eq. 1, Eq. 2 and Eq. 3. Then, the meta-weighter is updated as

$$\Theta^1_{Mix} = \Theta^0_{Mix} - \lambda_{Mix}\nabla_{\Theta_{Mix}}(\alpha_m Loss_{Lab}|\Theta^1_{Stu}), \quad (12)$$

where $\lambda_{Mix}$ is a learning rate for the meta-weighter. $Loss_{Lab}$ is only used to train meta-weighter at this step, without any effects to other modules.

Next, MSM obtains an updated meta-weighter ($ts$=1). Thus, meta-weighter can lead the student model to the more appropriate gradient direction, because it is already updated by the current batch data. With the meta-weighter taking the feedback of the current batch from the student model ($Loss_{Mix}$ in Fig. 3) as input, the original student model ($ts$=0) is formally updated with the supervision of the updated meta-weighter following

$$\Theta^2_{Stu} = \Theta^0_{Stu} - \lambda_{Stu}\nabla_{\Theta_{Stu}}(\alpha_m Loss^2_{stu}|\Theta^1_{Mix}), \quad (13)$$

where $Loss^2_{stu}$ is similarly calculating with $Loss_{Lab}$-based $\Theta^0_{Stu}$ and $\Theta^1_{Mix}$. The whole process of the three-step meta updating is summarized in Algorithm 2. We calculate the time costs with/without Algorithm 2, the results show we need about extra 0.35 times run cost for performing TMU.

TABLE 4
Hyperparameters for uncertainty range and utility threshold.

| | Res14 | Lap14 | Res15 |
|---|---|---|---|
| Uncertainty upper bound ($\mathcal{U}$%) | 80% | 75% | 65% |
| Uncertainty lower bound ($\mathcal{L}$%) | 20% | 30% | 35% |
| Utility threshold ($\beta^{tag}$) | 0.25 | 0.3 | 0.4 |

## 4 EXPERIMENT

### 4.1 Data

We utilized three ABSA datasets (Res14, Lap14, Res15) from SemEval-2014 [1] and SemEval-2015 [10] to evaluate the proposed MSM. These three datasets are widely employed by many researchers [9], [24], [25], [54] before. All the utilized datasets have fixed segmentation for training and testing. In this paper, MSM follows the study [9], which further randomly sample 20% of training data as validation sets for tuning hyper-parameters. The remaining 80% of data are used for training MSM. The details of used data are shown in Table 3. To illustrate the imbalanced label distributions in ABSA, we list the ratios of different tags in AE, OE, SC sub-tasks. The ratios of "O" tags exceed 90% in all three sub-tasks. On the contrary, all ratios of "I" tags are no more than 4%.

Following all baselines [9], [24], [25], [54], we regard the first predicted sentiment label of an aspect term as an SC result and filter out the conflict sentiment labels in SC sub-task. This filter may cause the sum of ratios in SC to be less than 100%. Besides, to further explore the data utilization of MSM, we allocate an augmented dataset for each dataset above. Specially, for Res14, Lap14, and Res15, we match Res15, Lap15, and Res14 as their augmented dataset, respectively. Lap15 dataset also comes from SemEval-2015 [10] challenge.

TABLE 5
The main results of MSM compared with full data supervised baselines. The numbers in the brackets are the percentages of labeled data used for supervised training and the remaining for self-training. + extra means MSM utilizes unlabeled data from other datasets. † indicates the results are quoted from the repetition experiments in the study [9].

| Model | Res14 | | | | Lap14 | | | | Res15 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | AE-F1 | OE-F1 | SC-F1 | ABSA-F1 | AE-F1 | OE-F1 | SC-F1 | ABSA-F1 | AE-F1 | OE-F1 | SC-F1 | ABSA-F1 |
| IMN-GloVe [25] | 84.01 | 85.64 | 71.90 | 68.32 | 78.46 | 78.14 | 69.92 | 57.66 | 69.80 | 72.11 | 60.65 | 57.91 |
| IMN-BERT [25]† | 84.06 | 85.10 | 75.67 | 70.72 | 77.55 | **81.00** | 75.56 | 61.73 | 69.90 | 73.29 | 70.10 | 60.22 |
| SPAN-BERT [54]† | 86.71 | - | 71.75 | 73.68 | 82.34 | - | 62.50 | 61.25 | 74.63 | - | 50.28 | 62.29 |
| RACL-GloVe [9] | 85.37 | 85.32 | 74.46 | 70.67 | 81.99 | 79.76 | 71.09 | 60.63 | 72.82 | **78.06** | 68.69 | 60.31 |
| RACL-BERT [9] | 86.38 | **87.18** | 81.61 | 75.42 | 81.79 | 79.72 | 73.91 | 63.40 | 73.99 | 76.00 | 74.91 | 66.05 |
| GBM-BERT [24] | 87.10 | 87.16 | 82.24 | 75.73 | **83.34** | 77.93 | **77.52** | 65.61 | - | - | - | - |
| MSM (10%) | 78.32 | 79.50 | 75.79 | 65.92 | 70.64 | 74.65 | 67.46 | 50.19 | 62.10 | 65.80 | 55.94 | 50.28 |
| MSM (30%) | 84.09 | 84.20 | 77.77 | 71.60 | 77.70 | 77.99 | 71.92 | 58.20 | 69.87 | 68.99 | 72.56 | 60.40 |
| MSM (50%) | 87.36 | 83.96 | 80.02 | 75.90 | 81.76 | 77.76 | 76.43 | 65.53 | **76.77** | 74.00 | **76.78** | 64.16 |
| MSM (70%+extra) | 88.18 | 84.65 | 80.83 | 76.21 | 80.37 | 78.09 | 77.42 | 66.26 | 74.59 | 77.66 | 74.36 | 65.19 |
| MSM (100%+extra) | **88.95** | 84.78 | **82.36** | **76.89** | 83.16 | 77.91 | 77.50 | **67.02** | 74.29 | 77.84 | 76.78 | **66.89** |

TABLE 6
Ablation analysis of MSM on testing sets. w/o means without.

| Dataset | | AE-F1 | OE-F1 | SC-F1 | ABSA-F1 |
|---|---|---|---|---|---|
| Res14 | MSM(50%) | **87.36** | 83.96 | 80.02 | **75.90** |
| | w/o meta-weighter | 86.34 | **85.94** | **81.06** | 74.93 |
| | w/o self-training | 85.31 | 83.69 | 78.47 | 72.72 |
| Lap14 | MSM(50%) | **81.76** | **77.76** | **76.43** | **65.53** |
| | w/o meta-weighter | 78.76 | 75.65 | 76.41 | 65.03 |
| | w/o self-training | 77.46 | 76.94 | 75.72 | 63.51 |
| Res15 | MSM(50%) | **76.77** | 74.00 | **76.78** | **64.16** |
| | w/o meta-weighter | 74.89 | **75.72** | 74.46 | 63.58 |
| | w/o self-training | 70.73 | 72.87 | 67.78 | 59.85 |

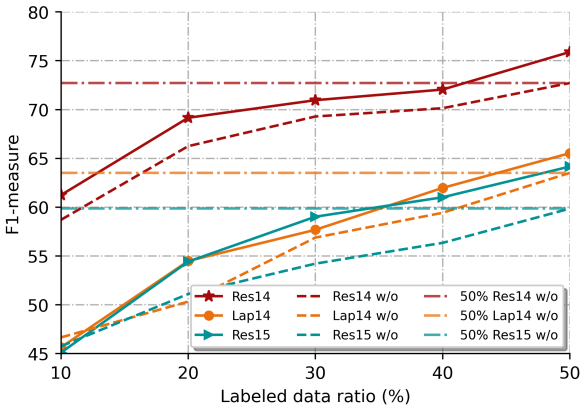



Fig. 4. Performance comparison with different input data setups. The performance is measured by ABSA-F1 on testing sets. The solid line denotes the results, given by MSM with incremental labeled data and fixed 50% unlabeled data. The dash line "w/o" means that the model only uses incremental labeled data without using unlabeled data for self-training. 50% dataset w/o means that the model uses fixed 50% labeled data without using unlabeled data.

### 4.2 Baselines

We introduce four benchmarking baselines:

**IMN** [25] is an interactive MTL model for AE, OE, and SC, which can be employed in both the token level and document level. IMN introduced a message passing mechanism that can transform from different tasks through shared latent variables. For a fair comparison, MSM compares with their results which using BERT-large as the pretrained language model. The cited results was reported by the study [9].

**SPAN** [54] is a pipelined ABSA method. SPAN extracted multi-targets by a heuristic algorithm at first, and then the related span representations were used to classify their sentiment polarities. We benchmark the results of this work that was reported in the work of [9].

**RACL** [9] proposed a relation-aware collaborative learning model for learning AE, OE, and SC, simultaneously. They explored how to establish interactions among three sub-tasks with an attention mechanism.

**GBM** [24] proposed a novel gating mechanism for the bridging of different sub-task in MTL. GBM allowed a task-specific tower to filter out useless information from a neighbour tower and fuse the useful information of the neighbour tower to obtain performance gains.

### 4.3 Evaluation Metrics and Hyperparameters

For fair comparisons, MSM employs the same evaluation metrics with all baselines mentioned above. Specifically, MSM is measured by AE-F1, OE-F1, SC-F1, and ABSA-F1. The first three F1 scores are computed by normal precision and recall measures. For SC-F1, if an aspect contains multiple words, the predicted sentiment label of the first token is considered as the final result. For ABSA-F1, an aspect label is regarded as true positive only when related AE and SC labels are both correct. Besides, conflict labels are ignored when calculating SC-F1 and ABSA-F1. MSM utilizes BERT-large with 1024 hidden dimensions. The maximum length of input sentences is equal to 100. Adam optimizer [56] is used to update MSM. Batch size is 64. In Eq. 11, the learning rate $\lambda_{Stu}$ for BERT-large and for the linear layers on the top of BERT-large is equal to 4e-5 and 8e-5, respectively. $\lambda_{Mix}$ is set to 8e-5 in Eq. 12. $\alpha_m$ in Eqs 11, 12 and 13 is 0.5, 0.5, and 0.3 for Res14, Lap14 and Res15, respectively. The model utility threshold $\beta^{tag}$, upper ($\mathcal{U}$%), and lower ($\mathcal{L}$%) bounds of uncertainties for the entities that are delivered to the student model by MSM are shown in Table 4. The total training $Epoch$ is 500 in Algorithm 1. $N_{aug}$ is set to 2 in Eq 4.

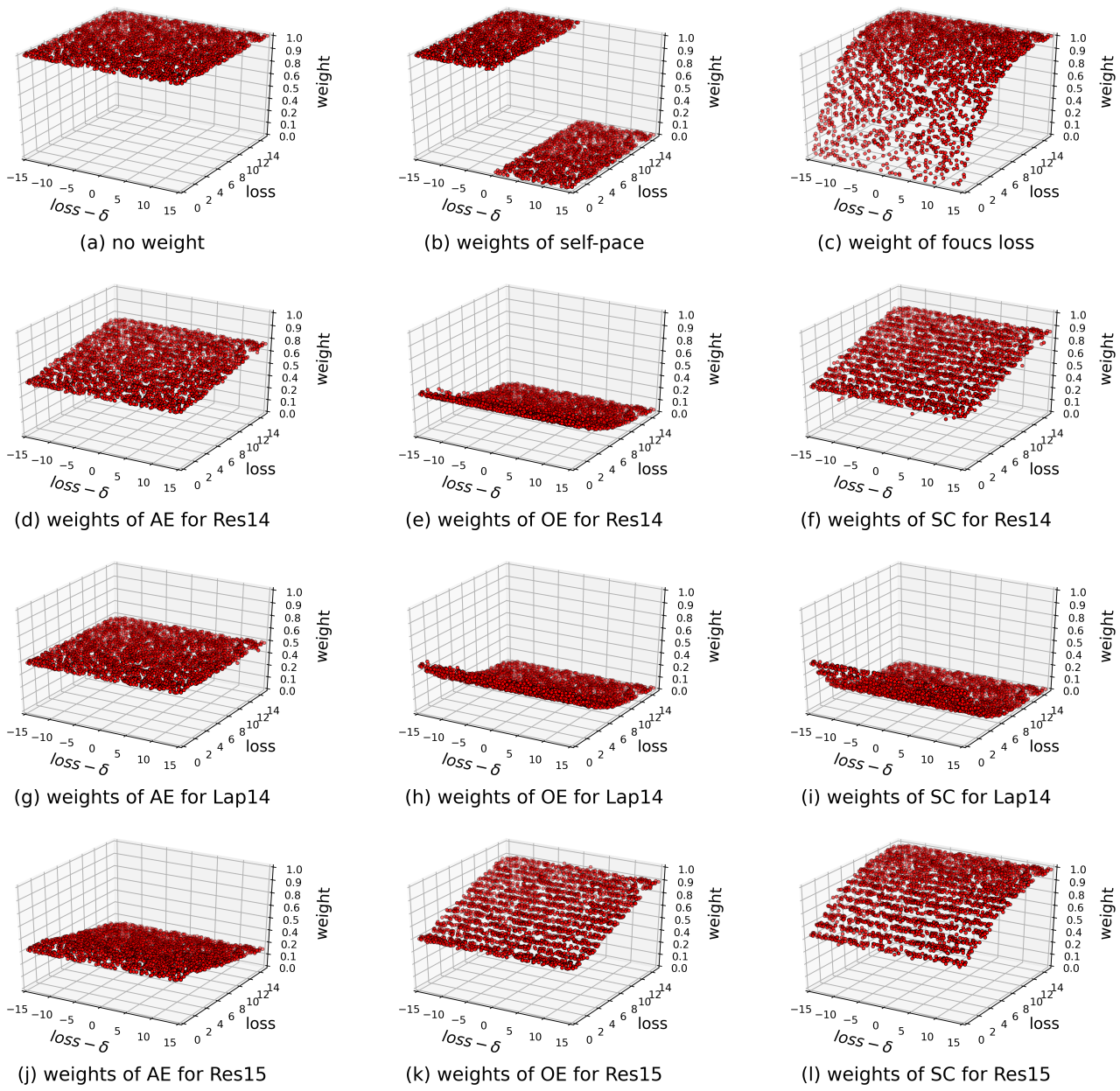



Fig. 5. The visualization of weights that are learned by various methods. (a) means no weights for data instances. (b) means using the self-pace function [57] to calculate weights. (c) means using focus loss [58] to calculate weights. (d)-(f), (e)-(i), and (j)-(l) show the weights learned by the meta-weighter of MSM at on AE, OE, and SC of Res14, Lap14, and Res15 dataset, respectively. The z-axes is the learned weight values. The x-axes denotes inputted losses for the meta-weighter. The y-axes indicates the difference between the current loss with its moving average, where $\delta$ is the averaged loss moving.

## 5 Results

Table 5 illustrates the main results of the proposed MSM. Since MSM is a self-training method, we explore its performances with different amounts of manually annotated data. It should notice that the benchmarking baselines adopt all labeled data of the corresponding dataset in Table 5, while we report MSM performance with the datasets in 10%, 30%, and 50%. Additionally, we also test MSM with 70% and 100% of data with extra unlabeled dataset augmentation. The purpose of introducing the unlabeled dataset augmentation is that the unlabeled data can help the learning of the student model and the meta-weighter with more pseudo-labeled data, since 70% and 100% of labeled data have been employed in these setups. Generally, all MSM F1 measures of different sub-tasks are rising with more labeled data for MSM. Considering ABSA-F1, with only 30% labeled data, our proposed MSM outperforms IMN-GloVe [25] in all sub-tasks. In the same conditions, MSM is also comparable to the performance of RACL-GloVe [9] in Res14 and Res15 datasets and obtains better results in Lap14. By using the same BERT encoder and 50% labeled data, MSM can achieve the improvements of 5.18% ABSA-F1 for Res14, 3.80% ABSA-F1 for Lap14, and 3.94% ABSA-F1 for Res15, compared with IMN-BERT [25].

In addition, MSM sightly outperforms the state-of-art GBM-BERT [24] in the Res14 and Lap14 dataset, with only 50% labeled data. For adopting 50% labeled data of Res15, ABSA-F1 of MSM is lower than results of full-data supervised RACL-BERT. This is because the size of Res15 is less than half of the sizes of Res14 and Lap14. This is insufficient to train a robust teacher model to guide a student model, because the generated pseudo data may contain too much noise from an unserviceable teacher model. Next, MSM adopts 70% and 100% annotated data in Res14, Lap14, and Res15 as labeled data, with extra unlabeled data from other datasets for self-training. The employed extra unlabeled data are introduced in section 4.1. With more labeled data, the improvements of Res15 are apparent, while the improvements of Res14 and Lap14 start to be limited. For example, ABSA-F1 measures are increased by 1.03% (50%+extra vs. 70%+extra) and 1.70% (70%+extra vs. 100%+extra) in Res15. The corresponding increments are 0.73% (50%+extra vs. 70%+extra) and 0.76% (70%+extra vs. 100%+extra) in Lap14, while these increments drop to 0.31% (50%+extra vs. 70%+extra) and 0.68% (70%+extra vs. 100%+extra) in Res14. These results indicate that MSM can reduce the requirements of labeled data to some extent. To sum up, compared with the strongest baselines in each dataset, MSM can achieve comparable results with only 50% labeled data. When using 70% and 100% labeled data, MSM takes more strict data selections to avoid the noise impacts. In such condition, the problem of insufficient data is not prominent, so MSM generates lesser pseudo-labels and tends to focus on re-weighting data instances in $D_{mix}$. Then, the final improvements of MSM are 1.16% and 1.41% compared with GBM-BERT [24] in Res14 and Lap14, and 0.84% with RACL-BERT [9] in Res15.

### 5.1 Ablation study

As shown in Table 6, we explore the utilities of different components in MSM. Ablation analysis is conducted to investigate the effects of the meta-weighter and self-training process. Without the meta-weighter, the student model allocates all gold-labeled and pseudo-labeled data with the same weights, and assumes that the three sub-tasks are also with equal complexity and importance. There are 0.97%, 0.5%, and 0.58% ABSA-F1 drops occupied in Res14, Lap14, and Res15, respectively. Comparing the rows without meta-weighter and the rows without self-training, the differences illustrate the effects of self-training with unlabeled data by excluding the influences of the meta-weighter. With pseudo-labeled data, we can obtain 2.21%, 1.52%, and 3.73% ABSA-F1 improvements for three utilized datasets. Finally, in the comparison between full MSM and MSM without self-training, we omit the weighted self-training process. The model without self-training only trains a student model with 50% training data, excluding the teacher model and meta-weighter. The results show that the full MSM model improves 3.18%, 2.02%, and 4.31% ABSA-F1 in Res14, Lap14, and Res15, respectively.

### 5.2 Influence of labeled and unlabeled data

This section explores the influences of different numbers of annotated data on MSM. First, we compare the effects of including and excluding unlabeled data on model performance (see the solid line vs. the dash line in Fig. 4). Generally, all models show the increasing F1 measures by employing more labeled data. MSM (including unlabeled data) can stably improve the ABSA-F1 measures across all the employed labeled data ratios from 10% to 50%.

The biggest gap between MSM (the solid line) and MSM without unlabeled data (the dash line) appears in using 30% labeled data in Res15, accounting for 4.81%. For Res14 and Lap14, the most prominent improvements are given by 3.18% with 50% of labeled data, and 4.15% with 20% of labeled data, respectively. MSM with 10% labeled data displays a limited enhancement for Lap14 and Res15 dataset. By using more than 20% labeled data, the improvement becomes apparent. This is because only utilizing 10% labeled data is hard to supervise a usable teacher model. The teacher model needs an effective initialization to help MSM avoid gradual drift.

Next, we compare the performance of MSM with a smaller fraction of the labeled data and the unlabeled data (the solid line), and the model with 50% of the labeled data and without unlabeled data (the vertical dash-dotted line). In Fig. 4, the intersection of the solid line "Res14" and the dash-dotted line "50% Res14 w/o" at the point of 41% indicates that MSM can achieve the same performance with less labeled data, saving 9% of labeled data. Similarly, by using MSM, about 4.5% and 15% labeled data can be saved for Lap14 and Res15 datasets.

### 5.3 Loss visualization

Considering that the meta-weighter of MSM aims to automatically learn different weighting strategies under different conditions, we visualize the generated weights and compare them with three typical weighting methods (weighters). We use virtual losses ranging from 0 to 15 as inputs of weighters. The virtual losses are automatically generated numbers with uniform distribution. These losses can simulate all possible inputs in a suitable range to our weighters, which are helpful to visualize the learned distribution of weighters. Then, the losses are fed to different weighters. In Fig. 5 (a)-(c), the weighters are manually designed with explicit mathematics functions. In Fig. 5 (d)-(l), the weighter is our proposed meta-weighter that is trained with different ABSA sub-tasks and datasets.

The distributions of calculated weights in Fig. 5 (a) and (b) are not changed over different losses. Fig. 5 (a) represents non-weight scenarios, namely all data instances with equal weights under any conditions. Fig. 5 (b) is self-pace-based methods [57], which chooses easy samples to learn and ignores the hard ones. Fig. 5 (c) is focus loss [58] which was designed for imbalanced data. The distribution in a waterfall shape shows that instances from classes which have insufficient labels usually have a larger loss. Focus loss allocates these instances with larger weights to help the model focus on hard negative examples. The methods in Fig. 5 (a), (b), and (c) are based on a predefined weight function. They cannot adaptively respond to the feedback from the student model in different condition.

In contrast to all the aforementioned predefined methods, the distributions of generated weights from the meta-weighter of MSM are dynamically changed over different

TABLE 7
Correctly and incorrectly classified examples. The gold labels are the same with the predicted labels in the correct example.

| An Example of Correct ABSA | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Sentence: | The | price | is | reasonable | , | although | the | service | is | poor | . | |
| Predicted AE: | O | B | O | O | O | O | O | B | O | O | O | |
| Predicted OE: | O | O | O | B | O | O | O | B | O | B | O | |
| Predicted SC: | O | pos | O | O | O | O | O | neg | O | O | O | |
| Two Typical Examples of Incorrect ABSA | | | | | | | | | | | | |
| Sentence: | students | love | this | place | so | it | makes | for | a | fun | young | atmosphere | . |
| Gold AE: | O | O | O | O | O | O | O | O | O | O | O | B | O |
| Predicted AE: | O | O | O | B | O | O | O | O | O | O | O | B | O |
| Gold OE: | O | B | O | O | O | O | O | O | O | B | B | O | O |
| Predicted OE: | O | B | O | O | O | O | O | O | O | B | I | O | O |
| Gold SC: | O | O | O | O | O | O | O | O | O | O | O | pos | O |
| Predicted SC: | O | O | O | pos | O | O | O | O | O | O | O | pos | O |
| Sentence: | Web | access | through | the | 3G | is | so | slow | , | it 's | very | frustrating | . |
| Gold AE: | B | I | O | O | B | O | O | O | O | O | O | O | O |
| Predicted AE: | B | I | O | O | B | O | O | O | O | O | O | O | O |
| Gold OE: | O | O | O | O | O | O | O | B | O | O | O | B | O |
| Predicted OE: | O | O | O | O | O | O | O | B | O | O | O | B | O |
| Gold SC: | neg | neg | O | O | neg | O | O | O | O | O | O | O | O |
| Predicted SC: | neg | neg | O | O | O | O | O | O | O | O | O | O | O |

instances with different feedback signals. Fig. 5 (d)-(f), (g)-(i), and (j)-(l) show the final weight distributions for AE, OE, and SC sub-tasks of the three utilized datasets, respectively. Two main observations can be found in these figures. First, MSM can automatically generate task-specific weights for the three sub-tasks. For different sub-tasks, MSM can get rid of human experts to design task-specific weighting functions. For example, given Res14 dataset, the generated weights approach a linear increase for AE, and an exponential decrease for OE. The distribution pattern differences between tasks can be observed in other datasets as well.

Second, the generated weight distributions can be different for the same tasks in different datasets. Specifically, comparing Fig. 5 (f), (i), and (l), these figures show that the generated weight distributions of SC for Res14 and Res15 are similar. MSM allocates complex samples with higher weights, selecting hard samples for training models. On the contrary, for SC of Lap14, MSM prefers to employ easy samples to learn, because the weights decrease when the losses increase. It may suggest that as the generated pseudo labels of SC in Lap14 are noisier, MSM tries to filter out those very pseudo-labeled data.

### 5.4 Case study

We list correct and incorrect ABSA examples in Table 7. As shown in the correct example, the sentence is a typical ABSA instance with multiple aspects with different sentiment polarities. MSM correctly recognizes the aspects, opinions, and related sentiment polarities in the sentence.

Besides, we display the two incorrect examples, which represent three kinds of most common errors in MSM. In the first incorrect example, there are two types of errors. The first one is MSM recognize "place" as an aspect and predict related positive sentiment. However, the word "place" is not regarded as an aspect and has no sentiment in gold annotations. The related reason may come from the tokens "love" and "this" before "place". These three tokens compose a common pattern to recognize aspects, which may mislead our model. The second error occurred with the token "young". Because the token "fun" and "young" are contiguous, MSM allocate "I" label for the token "young". After reviewing corpus, We find that gold annotations also assign "I" label for the second token in such condition, which may confuse our model.

In the second incorrect example, although all aspects are appropriately recognized, some related sentiment labels are missed. Considering MSM focuses on self-training and re-weighting from the view of data distribution, this phenomenon indicates that the interactions of the three sub-tasks are not enhanced enough in MSM. Related works [9], [24] may be helpful to mitigate this problem by establishing more information fusion among different sub-tasks.

## 6 Conclusion

This paper proposes a meta-based self-training method MSM to alleviate the problem of insufficient and imbalanced data in the ABSA task. MSM consists of a teacher model, a student model, and a meta-weighter, where the teacher model and the student model have the exact same structure. The teacher model is utilized to generate pseudo labels with extra unlabeled data for training the student model. In addition, a three-step meta updating method for jointly training the meta-weighter and the student model is proposed, aiming to avoid gradual drift caused by noise in the automatically labeled data. In the future, we will learn to deal with multiple sequence labeling tasks with more severe low-data scenarios, such as N-way-K-shot setting.

## References

[1] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar, "SemEval-2014 task 4: Aspect based sentiment analysis," in *SemEval*, 2014, pp. 27–35.
[2] N. Jinjie, P. Vlad, Y. Tom, Z. Haicang, and C. Erik, "Hitkg: Towards goal-oriented conversations via multi-hierarchy learning," in *AAAI*, 2022.

[3] Y. Ma, K. L. Nguyen, F. Z. Xing, and E. Cambria, "A survey on empathetic dialogue systems," *Information Fusion*, vol. 64, pp. 50–70, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253520303092

[4] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *AAAI*, vol. 32, no. 1, 2018.

[5] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in *ACL*, 2019, pp. 537–546.

[6] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An interactive multi-task learning network for end-to-end aspect-based sentiment analysis," in *ACL*, 2019, pp. 504–515.

[7] X. Li, L. Bing, P. Li, W. Lam, and Z. Yang, "Aspect term extraction with history attention and selective transformation," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*. International Joint Conferences on Artificial Intelligence Organization, 7 2018, pp. 4194–4200. [Online]. Available: https://doi.org/10.24963/ijcai.2018/583

[8] W. Wang, S. J. Pan, D. Dahlmeier, and X. Xiao, "Coupled multi-layer attentions for co-extraction of aspect and opinion terms," in *AAAI*, vol. 31, no. 1, 2017.

[9] Z. Chen and T. Qian, "Relation-aware collaborative learning for unified aspect-based sentiment analysis," in *ACL*, 2020, pp. 3685–3694.

[10] M. Pontiki, D. Galanis, H. Papageorgiou, S. Manandhar, and I. Androutsopoulos, "SemEval-2015 task 12: Aspect based sentiment analysis," in *SemEval*, 2015, pp. 486–495.

[11] Z. Gao, P. Puttapirat, J. Shi, and C. Li, "Renal cell carcinoma detection and subtyping with minimal point-based annotation in whole-slide images," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2020, pp. 439–448.

[12] S. Mukherjee and A. H. Awadallah, "Uncertainty-aware self-training for text classification with few labels," *arXiv preprint arXiv:2006.15315*, 2020.

[13] C. Wei, K. Sohn, C. Mellina, A. Yuille, and F. Yang, "Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 10 857–10 866.

[14] D. Berthelot, N. Carlini, I. Goodfellow, N. Papernot, A. Oliver, and C. Raffel, "Mixmatch: A holistic approach to semi-supervised learning," *arXiv preprint arXiv:1905.02249*, 2019.

[15] K. Sohn, D. Berthelot, N. Carlini, Z. Zhang, H. Zhang, C. A. Raffel, E. D. Cubuk, A. Kurakin, and C.-L. Li, "Fixmatch: Simplifying semi-supervised learning with consistency and confidence," *Advances in Neural Information Processing Systems*, vol. 33, pp. 596–608, 2020.

[16] Z. Chiyuan, B. Samy, H. Moritz, R. Benjamin, and O. Vinyals, "Understanding deep learning requires rethinking generalization (2016)," in *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009*. ACM International Conference Proceeding Series, 2009, pp. 41–48.

[17] H. Bao, K. He, X. Yin, X. Li, X. Bao, H. Zhang, J. Wu, and Z. Gao, "Bert-based meta-learning approach with looking back for sentiment analysis of literary book reviews," in *Natural Language Processing and Chinese Computing*, L. Wang, Y. Feng, Y. Hong, and R. He, Eds. Cham: Springer International Publishing, 2021, pp. 235–247.

[18] Y. Zou, Z. Yu, X. Liu, B. Kumar, and J. Wang, "Confidence regularized self-training," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5982–5991.

[19] X. Hu, C. Zhang, F. Ma, C. Liu, L. Wen, and P. S. Yu, "Semi-supervised relation extraction via incremental meta self-training," *arXiv preprint arXiv:2010.16410*, 2020.

[20] R. Mao, C. Lin, and F. Guerin, "End-to-end sequential metaphor identification inspired by linguistic theories," in *ACL*, 2019, pp. 3888–3898.

[21] M. Ge, R. Mao, and E. Cambria, "Explainable metaphor identification inspired by conceptual metaphor theory," in *AAAI*, 2022.

[22] J. Shu, Q. Xie, L. Yi, Q. Zhao, S. Zhou, Z. Xu, and D. Meng, "Meta-weight-net: Learning an explicit mapping for sample weighting," *Advances in Neural Information Processing Systems*, vol. 32, pp. 1919–1930, 2019.

[23] J. S. Supancic and D. Ramanan, "Self-paced learning for long-term tracking," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2379–2386.

[24] R. Mao and X. Li, "Bridging towers of multi-task learning with a gating mechanism for aspect-based sentiment analysis and sequential metaphor identification," *AAAI*, vol. 35, pp. 13 534–13 542, 2021.

[25] R. He, W. S. Lee, H. T. Ng, and D. Dahlmeier, "An interactive multi-task learning network for end-to-end aspect-based sentiment analysis," in *ACL*, 2019, pp. 504–515.

[26] Y. Mao, Y. Shen, C. Yu, and L. Cai, "A joint training dual-mrc framework for aspect based sentiment analysis," *arXiv preprint arXiv:2101.00816*, 2021.

[27] Z. Gao, B. Hong, X. Zhang, Y. Li, C. Jia, J. Wu, C. Wang, D. Meng, and C. Li, "Instance-based vision transformer for subtyping of papillary renal cell carcinoma in histopathological image," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2021, pp. 299–308.

[28] Q. Xie, Z. Dai, E. Hovy, T. Luong, and Q. Le, "Unsupervised data augmentation for consistency training," in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 6256–6268. [Online]. Available: https://proceedings.neurips.cc/paper/2020/file/44feb0096faa8326192570788b38c1d1-Paper.pdf

[29] Y. Yu, S. Zuo, H. Jiang, W. Ren, T. Zhao, and C. Zhang, "Fine-tuning pre-trained language model with weak supervision: A contrastive-regularized self-training approach," *arXiv preprint arXiv:2010.07835*, 2020.

[30] R. Mao, C. Lin, and F. Guerin, "Word embedding and WordNet based metaphor identification and interpretation," in *ACL*, 2018, pp. 1222–1231.

[31] R. Mao, X. Li, M. Ge, and E. Cambria, "MetaPro: A computational metaphor processing model for text pre-processing," *Information Fusion*, vol. 86-87, pp. 30–43, 2022. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1566253522000550

[32] J. Wei and K. Zou, "EDA: Easy data augmentation techniques for boosting performance on text classification tasks," in *EMNLP*, 2019, pp. 6382–6388.

[33] Y. Chen, Y. Zhang, C. Zhang, G. Lee, R. Cheng, and H. Li, "Revisiting self-training for few-shot learning of language model," in *EMNLP*, 2021, pp. 9125–9135.

[34] J. Du, E. Grave, B. Gunel, V. Chaudhary, O. Celebi, M. Auli, V. Stoyanov, and A. Conneau, "Self-training improves pre-training for natural language understanding," in *NAACL*, 2021, pp. 5408–5418.

[35] Y. Wang, S. Mukherjee, H. Chu, Y. Tu, M. Wu, J. Gao, and A. H. Awadallah, *Meta Self-Training for Few-Shot Neural Sequence Labeling*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 1737–1747. [Online]. Available: https://doi.org/10.1145/3447548.3467235

[36] R. Mihalcea, "Co-training and self-training for word sense disambiguation," in *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, 2004, pp. 33–40.

[37] S. Clark, J. R. Curran, and M. Osborne, "Bootstrapping pos-taggers using unlabelled data," in *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 2003, pp. 49–55.

[38] H.-S. Chang, E. Learned-Miller, and A. McCallum, "Active bias: Training more accurate neural networks by emphasizing high variance samples," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper/2017/file/2f37d10131f2a483a8dd005b3d14b0d9-Paper.pdf

[39] P. Ren, Y. Xiao, X. Chang, P.-Y. Huang, Z. Li, B. B. Gupta, X. Chen, and X. Wang, "A survey of deep active learning," 2021.

[40] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proceedings of The 33rd International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, M. F. Balcan and K. Q. Weinberger, Eds., vol. 48. New York, New York, USA: PMLR, 20–22 Jun 2016, pp. 1050–1059. [Online]. Available: https://proceedings.mlr.press/v48/gal16.html

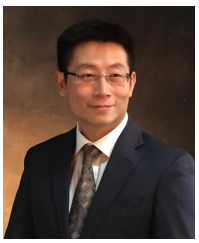
[41] L. Jiang, Z. Zhou, T. Leung, L.-J. Li, and L. Fei-Fei, "MentorNet: Learning data-driven curriculum for very deep neural networks on corrupted labels," in *Proceedings of the 35th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, J. Dy and A. Krause, Eds., vol. 80.

PMLR, 10–15 Jul 2018, pp. 2304–2313. [Online]. Available: https://proceedings.mlr.press/v80/jiang18c.html
[42] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09. New York, NY, USA: Association for Computing Machinery, 2009, pp. 41–48. [Online]. Available: https://doi.org/10.1145/1553374.1553380
[43] M. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*, J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta, Eds., vol. 23. Curran Associates, Inc., 2010. [Online]. Available: https://proceedings.neurips.cc/paper/2010/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf
[44] J. S. Andersen, T. Schöner, and W. Maalej, "Word-level uncertainty estimation for black-box text classifiers using RNNs," in *Proceedings of the 28th International Conference on Computational Linguistics*. Barcelona, Spain (Online): International Committee on Computational Linguistics, Dec. 2020, pp. 5541–5546. [Online]. Available: https://aclanthology.org/2020.coling-main.484
[45] Y. Zhou, B. Yang, D. F. Wong, Y. Wan, and L. S. Chao, "Uncertainty-aware curriculum learning for neural machine translation," in *ACL*, 2020, pp. 6934–6944.
[46] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, 2009, pp. 41–48.
[47] Z. Wu, C. Ying, F. Zhao, Z. Fan, X. Dai, and R. Xia, "Grid tagging scheme for aspect-oriented fine-grained opinion extraction," in *EMNLPs*, 2020, pp. 2576–2585.
[48] Z. Dai, C. Peng, H. Chen, and Y. Ding, "A multi-task incremental learning framework with category name embedding for aspect-category sentiment analysis," in *EMNLP*, 2020, pp. 6955–6965.
[49] H. Wan, Y. Yang, J. Du, Y. Liu, K. Qi, and J. Z. Pan, "Target-aspect-sentiment joint detection for aspect-based sentiment analysis," *AAAI*, vol. 34, no. 05, pp. 9122–9129, 2020.
[50] Z. Li, X. Li, Y. Wei, L. Bing, Y. Zhang, and Q. Yang, "Transferable end-to-end aspect-based sentiment analysis with selective adversarial learning," in *EMNLP*, 2019, pp. 4590–4600.
[51] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, G. Gordon, D. Dunson, and M. Dudík, Eds., vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, pp. 315–323. [Online]. Available: https://proceedings.mlr.press/v15/glorot11a.html
[52] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *NAACL*, 2019, pp. 4171–4186.
[53] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*. PMLR, 2016, pp. 1050–1059.
[54] M. Hu, Y. Peng, Z. Huang, D. Li, and Y. Lv, "Open-domain targeted sentiment analysis via span-based extraction and classification," in *ACL*, 2019, pp. 537–546.
[55] J. Shi, C. Jia, Z. Gao, T. Gong, C. Wang, and C. Li, "Meta mask correction for nuclei segmentation in histopathological image," in *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. IEEE, 2021, pp. 2059–2063.
[56] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
[57] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models." in *NIPS*, vol. 1, 2010, p. 2.
[58] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

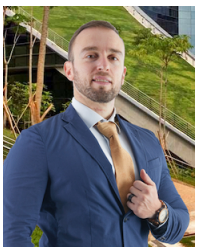

**Kai He** is a PhD candidate at the school of Computer Science and Technology, Xi'an Jiaotong University, under the supervision of Chen Li. Currently, he is on an academic visit to the school of Computer Science and Engineering of Nanyang Technological University under the supervision of Erik Cambria. His research interests include information extraction and sentiment analysis in NLP field. In his free time, Kai also likes to play badminton, hiking and try different food.

**Rui Mao** received his Ph.D. in Computing Science from the University of Aberdeen. Currently, he is a Research Fellow in Nanyang Technological University, studying conversational AI. His research interests include computational metaphor processing, token-level and sequence-level semantic understanding, and affective computing. He has published several papers as the first author in top-tier international conferences and journals, e,g., ACL, AAAI, and Information Fusion.

**Tieliang Gong** graduated from Xi'an Jiaotong University in 2018. He conducted postdoctoral research at the school of mathematics and statistics of the University of Ottawa from October 2018 to September 2020. Research interests include statistical learning theory, robust statistical inference, machine learning and so on. The research results are mainly published in NeurIPS, AAAI, IEEE Trans On Cybernetics, Neural Computing and other top international conferences and journals.



**Chen Li** is Doctor of Cambridge University, UK, postdoctoral fellow of Massachusetts Institute of technology, overseas fellowship winner of Cambridge University and fellowship winner of European Molecular Biology Laboratory. At present, he works as a professor Xi'an Jiaotong University. He has been engaged in the research and development of biomedical text mining in EMBL-EBI. His Biomodels data standard system has been rated the most important resource in the field of systems biology.



**Erik Cambria** (Fellow, IEEE) is the Founder of SenticNet, a Singapore-based company offering B2B sentiment analysis services, and a Professor at NTU, where he also holds the appointment of Provost Chair in Computer Science and Engineering. His research focuses on neurosymbolic AI for explainable natural language processing in domains like sentiment analysis, dialogue systems, and financial forecasting. He is Associate Editor of many top-tier journals, e.g., Information Fusion and IEEE Trans on Affective Computing.