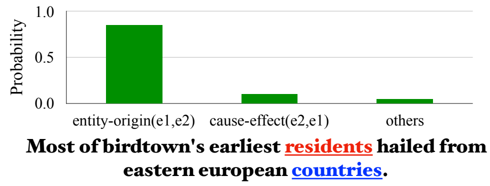
Self-training with Modeling Ambiguous Data for Low-Resource Relation Extraction

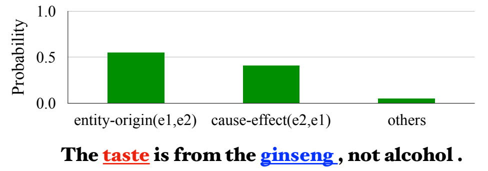
Anonymous ACL submission

Abstract

We present a simple yet effective approach to improve the performance of self-training relation extraction in a low-resource scenario. The approach first classifies the auto-annotated instances into two groups: confident instances and uncertain instances, according to the probabilities predicted by a teacher model. In contrast to most previous studies, which mainly only use the confident instances for self-training, we make use of the uncertain instances. We propose a method to identify some ambiguous but useful instances from the uncertain instances. Then, we propose to utilize negative training for the ambiguous instances and positive training for the confident instances. Finally, they are combined in a joint-training manner to build a relation extraction system. Experimental results on two widely used datasets with low-resource settings demonstrate that this new approach indeed achieves significant and consistent improvements when compared to several competitive self-training systems. [1]



(a) Confident instance



(b) Uncertain instance

Figure 1: Two examples of auto-annotated instances. For simplicity, we list two detailed relations and use "others" to represent the other relations.

1 Introduction

Relation Extraction (RE) is a fundamental task in Information Extraction, which aims to obtain a predefined semantic relation between two entities in a given sentence (Zhou et al., 2005). In recent years, fine-tuning the downstream RE tasks with pre-trained models (Soares et al., 2019; Wang et al., 2019; Li and Tian, 2020) has achieved significant progress with the rapid development of the "Pre-train and Fine-tune" Paradigm (Devlin et al., 2018; Liu et al., 2019; Lewis et al., 2020) which leverages large-scale unlabeled data. However, RE still suffers from the data scarcity problem. For most RE tasks, due to the task-specific definition of relations, the lack of customized annotation data poses great challenge for the supervised RE (Hendrickx et al., 2010; Zhang et al., 2017). Meanwhile, manual labeling a large-scale RE data is extremely time-consuming, expensive, and labor-intensive. As an alternative, automatically building annotated data for RE attracts a lot of attention in the research community (Mintz et al., 2010; Luo et al., 2019; Yu et al., 2020).

Self-training is a simple and effective approach to build auto-annotated data (Lee et al., 2013; Zhang and Zong, 2016; Xie et al., 2020; Vu et al., 2021). The idea is to use a teacher model trained on human-annotated data to automatically annotate the additional unlabeled data. Then we can combine the human-annotated data with some instances selected from the auto-annotated data to train a student model. In this paper, we follow the self-training framework to improve **Low-Resource RE**, which is closer to practical situations where the task starts with a small seed set of human-annotated data.

In the previous studies, the researchers often select the auto-annotated instances with high confidence, named as *confident instance*, and have achieved a certain success (Qian et al., 2009; Oliver et al., 2018). Figure 1(a) shows an example of con-

[1] Code, data and models will be made publicly available.

fident instance, where the teacher model can easily classify it as relation "entity-origin(e1,e2)" with the clue offered by "hailed from". Therefore, we first follow this kind of solutions to conduct self-training in our task. However, in the preliminary experiments, we find that *some relations might use similar expressions in instances* which makes the teacher model confused. As a result, for the uncertain instances, the teacher model gives similar high probabilities to some relations or assigns low probabilities to all the relations. An example of uncertain instance is shown in Figure 1(b), where the teacher model predicts the instance as relation "entity-origin(e1,e2)" with a probability 56%, as "cause-effect(e2,e1)" (the ground truth label) with 42%, and as other relations with only 2%. It is hard to distinguish between the first two relations as the expression "... is from ..." is often used for both. In most previous studies, such as (Sohn et al., 2020; Du et al., 2021), the uncertain instances are often discarded due to the confusion. However, we argue that ignoring all the uncertain instances might not be appropriate since they might contain useful information. For example, it is a good clue that the answer is one of the first two relations with a probability 98% for the instance in Figure 1(b).

Ideally, we would wish to fully use of all the auto-annotated instances to improve the RE system. But it is very hard due to the confusion problem. We split the uncertain instances into two groups: *ambiguous set* and *hard set*. The ambiguous set includes the instances for which the teacher model gives similar high probabilities to some relations (not so many), while the hard set includes the ones that the teacher model assigns low probabilities to all the relations. In this paper, we focus on the ambiguous set and propose an approach to use the ambiguous instances and the confident instances to improve Low-Resource RE.

In our approach, we tackle two main issues when exploiting the ambiguous instances: 1) how to identify the ambiguous instances from the auto-annotated instances; 2) how to train a new model with the ambiguous instances. As for the first issue, we adopt a probability accumulation method (Holtzman et al., 2019) to obtain a set of relations containing the great majority of the probability, and then identify the ambiguous instances based on this set. To deal with the second issue, we make an assumption: *For the ambiguous instances, the teacher model does not know which relation is the exact answer, but it does know that #1) the answer is (with high probability) in a set of candidates (likes the first two relations in Figure 1(b)) and #2) the answer is not the relations which are with very low probabilities (likes "others" in Figure 1(b)).* Under this assumption, we treat the ambiguous instances as partially-labeled training instances (Cour et al., 2011), where the answer is in a candidate set of labels, but only one of which is correct. Based on Assumption #1, it is naturally that we propose a training method which applies positive training on the partially-labeled training instances, POSPARTIALLABEL. However, since only one relation is correct among the candidates, POSPARTIALLABEL might go the wrong way when it supposes all of the candidates are correct. Therefore, based on Assumption #2, we propose another training method based on negative training to learn from partially-labeled training instances, NEGPARTIALLABEL. Finally, we use joint training to combine the ambiguous instances and the confident instances.

Our main contributions are as follows:

- We propose a method to classify the auto-annotated instances into three groups: confident set, ambiguous set, and hard set. The ambiguous instances are then treated as partially-labeled, which can reduce the effect of confused expressions. To our best knowledge, it is the first time that partial labeling is used to tag the auto-annotated instances in RE.

- We propose a simple yet effective approach to train with ambiguous instances under the self-training framework. In order to exploit the auto-annotated instances properly, we propose to apply negative training with the ambiguous instances and positive training with the confident instances. Negative training can utilize the information that the answer is not the relations which have very low probabilities predicted by the teacher model. Then, they are combined in a joint-training manner to obtain the final RE system.

To verify the effectiveness of our approach, we conduct experiments on two widely used datasets with low-resource settings. Results show that our proposed system significantly outperforms the conventional self-training system which only samples confident instances and other compared systems.

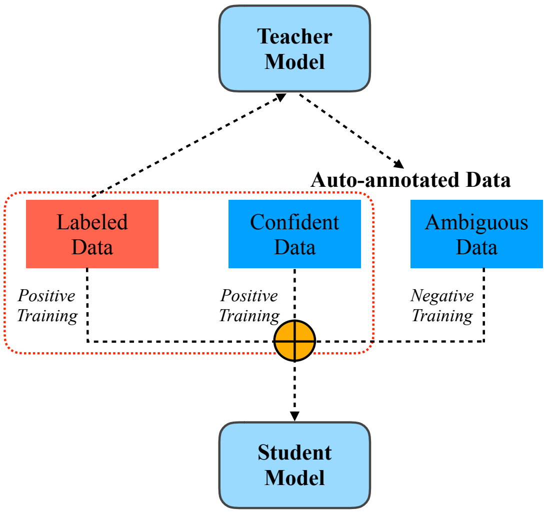


Figure 2: Framework of our approach.

2 Our Approach

We first briefly introduce the relation extraction as well as the self-training framework frequently used in the previous studies (Zhou et al., 2008; He et al., 2019). Then, we propose an algorithm to classify the auto-annotated instances into different groups: confident data, ambiguous data and hard data. Finally, we present a training method to use the confident and ambiguous data. The framework of the proposed approach is shown in Figure 2.

2.1 Self-Training for Relation Extraction

2.1.1 Relation Extraction

Fine-tuning on a pre-trained model, e.g., BERT, with a task-specific classifier is a common practice for downstream NLP tasks (Devlin et al., 2018). Following Soares et al. (2019), the relation extraction model is composed of a BERT encoder and a relation classification layer. Entity markers ([E1] for the head entity and [E2] for the tail entity) are inserted into input tokens to learn the entity representations. Concretely, the output representation of two entity markers are concatenated as input for the relation classification layer.

Formally, the output representation of an instance x after BERT is $\mathbf{h} = \mathbf{h}_{[E1]} \oplus \mathbf{h}_{[E2]}$. Then, the output probability distribution for M relations $p = [p_1, p_2, ..., p_M]$ is computed by the relation classification layer:

$$p = f(x) = Softmax(\mathbf{W}\mathbf{h} + \mathbf{b}), \quad (1)$$

where $\mathbf{W}$ and $\mathbf{b}$ are model parameters.

During training, each instance x from the human-annotated data is labeled with a one-hot label vector y: a single 1 value for the ground-truth label and 0 values for other labels. Then, the positive training is performed to calculate the cross entropy loss:

$$\mathcal{L}_{PT}(f(x), y) = -\sum_{i=1}^{M} y_i \log p_i, \quad (2)$$

where M is the number of relations, and y_i and p_i are the label and prediction probability of ith relation, respectively.

2.1.2 Self-Training

Generally, in the self-training framework, the unlabeled instances are labeled by the teacher model to form the auto-annotated data. As shown in Figure 2, the general flow of self-training is performed in the following steps: (1) use the human-annotated data to train a teacher model; (2) use the teacher model to conduct label prediction for unlabeled data; (3) select confident auto-annotated instances via a pre-defined probability threshold (described in Sec. 2.2); (4) combine confident auto-annotated data and human-annotated data to train a student model (the red dotted rectangle in Figure 2).

And in step (3) the remaining uncertain instances are considered to be useless. However, as described in Sec. 1, uncertain instances (e.g., the example in Figure 1(b)) might contain useful information.

2.2 Instance Classification

To make full use of the auto-annotated instances to improve the RE system, we use Algorithm 1 to classify the auto-annotated data into the confident data (line 14), ambiguous data (line 16) and hard data (line 18).

Confident instance. A probability threshold T is set to identify the confident instances. We set the instance whose highest prediction probability exceeds the probability threshold T to be confident instance, as the teacher makes a certain prediction about the instance.

Ambiguous instance. We adopt the probability accumulation method to identify the ambiguous instances, according to our observations that the teacher model gives similar high probabilities to some relation labels. Specifically, we sort the probabilities in prediction probability distribution and dynamically accumulate the probability of top relations (Line 4-10 in Algorithm 1) until the cumulative probability is larger than T. A hyper-parameter N is set to control the maximum size of candidate

Algorithm 1 Instance Classification

Input: auto-annotated data $\mathbf{D_{auto}} = \{x, P, Y\}$ containing sentence x, prediction distribution P and its corresponding relations Y

Parameter: probability threshold T, partial label size threshold N

Output: confident data $\mathbf{D_{con}}$, ambiguous data $\mathbf{D_{amb}}$, and hard data $\mathbf{D_{hard}}$

1: **for** $(x, P) \in \mathbf{D_{auto}}$ **do**
2: Let $score = 0.0$.
3: Let candidate label set $C = \{\}$
4: Arrange P from largest to smallest
5: Arrange Y by the order in P
6: **for** $(p, y) \in P, Y$ **do**
7: $score \leftarrow score + p$.
8: Append y to C.
9: **if** $score > T$ **then**
10: **break**
11: **end if**
12: **end for**
13: **if** $score > T$ **and** len(C) == 1 **then**
14: Append (x, C) to $\mathbf{D_{con}}$
15: **else if** $score > T$ **and** len(C) $\leqslant N$ **then**
16: Append (x, C) to $\mathbf{D_{amb}}$
17: **else**
18: Append (x, C) to $\mathbf{D_{hard}}$
19: **end if**
20: **end for**
21: **return** $\mathbf{D_{con}}, \mathbf{D_{amb}}, \mathbf{D_{hard}}$

relations[2], and the instance with no more than N candidate labels is considered as an ambiguous instance (line 15 in Algorithm 1).

2.3 Instance Label Tagging Mode

After identifying confident and ambiguous data from the auto-annotated data, we now have three training sets: a small seed set of human-annotated data $\mathbf{D_{hum}}$, a confident auto-annotated data $\mathbf{D_{con}}$, and an ambiguous auto-annotated data $\mathbf{D_{amb}}$. For the human-annotated data and confident data, we take the original one-hot label vector format as described in Sec. 2.1.1 to label the data.

For the ambiguous data, a variety of methods can be used to tag the instance. As shown in Table 1, given the probability distributions predicted by the teacher model, hard label mode assigns an exact label (the label with highest prediction probability) with a one-hot vector (Lee et al., 2013; Sohn et al., 2020) while soft label mode (Mey and Loog, 2016; Najafi et al., 2019; Xie et al., 2020) adopts probability distributions over the labels to cover all possible label choices.

[2]We will discuss the effect of the hyper-parameter N in Sec. 4.1.

Mode	Ent-Ori	Cau-Eff	Others
Probability	0.56	0.42	0.02
Hard Label	1	0	0
Soft Label	0.56	0.42	0.02
Partial Label	1	1	0

Table 1: An example of three tagging modes with given predicted probability distribution.

In this work, we propose to use the partial label to tag ambiguous instances. As the ambiguous example discussed in Sec. 1, the teacher gives similar high probabilities to several relations, partial label mode assigns each ambiguous instance with a set of candidate labels (the candidate label set C is described in Algorithm 1 line 3) and treats each candidate label equally to form the multi-hot label vector, as the example shown in Table 1.

2.4 Training with Partial Labels

Training strategies of the hard label and soft label have been explored in previous work (Lee et al., 2013; Xie et al., 2020). In this work, we focus on training on ambiguous data with partial labels.

2.4.1 Positive Training for Partial Labeling

We first propose to use positive training for partially labeled ambiguous data. This solution can deal with the instance with multiple positive labels and is an extended version of traditional positive training (Eqn. 2). Formally, an input instance x from ambiguous data is labeled with a multi-hot label vector y. Then, we calculate scores for relations with label 1 in y, individually. Finally, an averaged score for positive labels is used to update the model. The loss function is:

$$\mathcal{L}_{PTPL}(f(x), y) = -\frac{\sum_{i=1}^{M} y_i \log p_i}{\sum_{i=1}^{M} y_i}. \quad (3)$$

2.4.2 Negative Training for Partial Labeling

Inspired by negative training (Kim et al., 2019; Ma et al., 2021) which trains noisy data by selecting a random label as negative label, we propose to train ambiguous data in a negative manner. With the Assumption #2 described in Sec. 1, we are confident that the answer is not in the labels with low probabilities. Therefore, negative training is a feasible method to train ambiguous data by treating the relations out of candidate set C as negative labels.

In detail, we first randomly select a negative label. Then, the multi-hot label for ambiguous data is converted into a one-hot label which contains a 1 value for the selected negative label and others are 0 values. Finally, the loss function for ambiguous instances under negative training is:

$$\mathcal{L}_{NTPL}(f(x), y) = -\sum_{i=1}^{M} y_i \log(1 - p_i), \quad (4)$$

where the one-hot label y is dynamically changed by randomly selecting a negative label during training.

2.4.3 Joint Training

During training, another challenge is how to combine three data sets ($\mathbf{D_{hum}}$, $\mathbf{D_{con}}$, and $\mathbf{D_{amb}}$) under both positive and negative training. For simplicity, we split this challenge into two issues: (1) how to keep the importance of human-annotated data and (2) how to train instances by a mixed positive and negative training method.

For the first issue, the quality of human-annotated data $\mathbf{D_{hum}}$ is higher than the auto-annotated data while the size of $\mathbf{D_{hum}}$ is usually much smaller than $\mathbf{D_{con}}$ and $\mathbf{D_{amb}}$. Therefore, it is likely that a small amount of human-annotated data may be overwhelmed by the large amount of auto-annotated data (Li et al., 2014). To relieve the problem, we propose to use a two-stage fine-tune based solution which trains human-annotated data and auto-annotated data separately. In detail, we first train a preliminary model $\mathcal{M}_{auto}$ by fine-tuning on BERT with $\mathbf{D_{con}}$ and $\mathbf{D_{amb}}$. And then we go on to train a final model $\mathcal{M}_{hum}$ by fine-tuning on $\mathcal{M}_{auto}$ with $\mathbf{D_{hum}}$.

As for the second issue, the positive training method for partially labeled ambiguous data (described in Eqn. 3) is an extended version of the standard positive training method (Eqn. 2). Therefore, we can directly use this solution to train the mixed data which contains $\mathbf{D_{hum}}$, $\mathbf{D_{con}}$ and partially labeled data $\mathbf{D_{amb}}$.

In order to combine positive training (Eqn. 2) and negative training (Eqn. 4), we first introduce a flag variable z to represent whether current input instance is partially labeled or not:

$$z = \begin{cases} 1 & \text{if partially labeled,} \\ 0 & \text{others.} \end{cases} \quad (5)$$

Then, a unified loss function is:

$$\mathcal{L}(f(x), y) = -\sum_{i=1}^{M} y_i \log |z - p_i|, \quad (6)$$

where $|*|$ is the absolute value.

3 Experiments

In this section, we describe our experimental results and present detailed analysis.

3.1 Datasets and Metrics

Datasets. We conduct our experiments on two widely used relation extraction datasets: SemEval 2010 Task 8 (SemEval) and Re-TACRED, which are built for supervised training. The brief information of two datasets are as follows:

- SemEval: A classical dataset in relation extraction which contains 10,717 annotated sentences covering 9 relations with two directions and one special relation "no_relation" (Hendrickx et al., 2010).
- Re-TACRED: A repaired version of TACRED (Zhang et al., 2017) proposed by (Stoica et al., 2021) who re-annotated part of examples in training set and refined relation definition. In total, it contains 91,467 sentences covering 40 relations (also including a "no_relation" class).

Data	Rel	Train	Dev	Test	Unlabel
SemEval	10	100	976	1,829	4,212
Re-TACRED	10	100	5,863	4,153	15,635

Table 2: Statistics of SemEval and Re-TACRED under low-resource settings.

Low-Resource Setting. In this work, we focus on addressing the relation extraction task under a low-resource scenario. In order to avoid the interference of data imbalance problem (Li et al., 2011), we select top 10 relations (excluding "no_relation") by sorting the relations on the number of instances they have in the original training set. To simulate

Method	Micro-F1	Macro-F1
$\mathbf{D_{hum}}$	73.5	72.3
$\mathbf{D_{hum}}$+$\mathbf{D_{con}}$	81.5	80.9
$\mathbf{D_{hum}}$+$\mathbf{D_{con}}$+$\mathbf{D_{amb}}$	78.9	78.5

Table 3: Results of different data combinations on the development set of SemEval.

the low-resource scenario, we randomly sample 10 instances for each relation as a seed set of human-annotated training data and the rest instances are used as unlabeled data. As for the development and test sets, we keep all the instances for the top 10 relations. The statistics of two datasets are shown in Table 2.

Metrics. In order to give an overall evaluation, we follow previous studies (Hendrickx et al., 2010; Zhang et al., 2017; Stoica et al., 2021) to report both averaged micro F1 scores (Micro-F1) and averaged macro F1 scores (Macro-F1).

3.2 Hyper-Parameters

In this paper, we use $\texttt{BERT}_{\texttt{base}}$ (Devlin et al., 2018) as pre-trained model for all the systems. We choose the settings of hyper-parameters according to the performance on the development set of SemEval. As a result, we use a batch size of 32 and a learning rate of 5e-5 with Adam (Kingma and Ba, 2014) and train the model in 20 epochs on one GPU. We set probability threshold T as 0.95, and partial label size N as 5. The partial label size N controls the sampling of ambiguous instances where a larger number collects more instances as it looses the condition. We run 5 seeds to get an averaged value as the final result for each system.

3.3 Preliminary Experiments

In order to verify the effective of confident data and ambiguous data in the hard label mode (as shown in Table 1), we conduct the preliminary experiments by using different data combinations in a conventional self-training system.

Table 3 shows the results of three data combinations on the development set of SemEval. From the table, we find that self-training with confident data ($\mathbf{D_{hum}}$+$\mathbf{D_{con}}$) gets a significant performance improvement (+8.0 on Micro-F1 and +8.6 on Macro-F1), compared with the supervised model ($\mathbf{D_{hum}}$) which only uses human-annotated data. However, if we continually add the ambiguous data ($\mathbf{D_{hum}}$+$\mathbf{D_{con}}$+$\mathbf{D_{amb}}$), the performance declines. These facts indicate that the confident data is pretty useful for self-training, while the ambiguous data can not be used directly with self-training.

3.4 Comparison Systems

We use the RE model proposed by Soares et al. (2019) as base model to build all the systems compared in our experiments. We implement the systems by ourselves based on the previous studies. The comparison systems are listed as follows:

SUPERVISED. We follow Soares et al. (2019) to fine-tune the pre-trained BERT on the downstream relation extraction task with human-annotated data in a supervised manner.

SELF-TRAINING. Our implementation of the representative self-training method (Lee et al., 2013), which only uses confident data from the auto-annotated data.

HARD PSEUDO-LABEL. Our implementation of self-training method with the confident and ambiguous data in hard label mode. The training strategy is the same as SELF-TRAINING (Lee et al., 2013), while the difference is the input data.

SOFT PSEUDO-LABEL. Our implementation of self-training method with the confident and ambiguous data in soft labels mode (Xie et al., 2020).

NEGHARDLABEL. Our implementation of the negative training method from Kim et al. (2019) which is originally used for tackling noisy label problems in image classification. We use negative training for ambiguous data with hard labels.

3.5 Main Results

In this section, we show the model performances of our proposed systems (POSPARTIALLABEL and NEGPARTIALLABEL), and meanwhile compare them with the other systems mentioned above. POSPARTIALLABEL refers to the system with positive training for ambiguous data in partial-labeled mode (described in Sec. 2.4.1), while NEGPARTIALLABEL refers to the one with negative training for ambiguous data in partial-labeled mode (described in Sec. 2.4.2).

Table 4 shows the main results on SemEval and Re-TACRED. For simplicity, we report the average performance (average Micro-F1 and average Macro-F1) on two set sets to evaluate the model. Our observations are:

#	Method	SemEval				Re-TACRED				Avg.	
		Micro-F1		Macro-F1		Micro-F1		Macro-F1		Micro-F1	Macro-F1
		Dev	Test	Dev	Test	Dev	Test	Dev	Test	Test	Test
1	SUPERVISED	73.5	76.0	72.3	75.5	80.7	84.7	73.9	74.3	80.4	74.9
2	SELF-TRAINING	81.5	81.7	80.9	81.4	85.4	89.0	77.2	76.8	85.5	79.1
3	HARD PSEUDO-LABEL	78.9	80.0	78.5	79.6	87.3	90.5	79.6	77.8	85.3	78.7
4	SOFT PSEUDO-LABEL	80.0	80.8	79.0	80.8	86.5	89.2	78.7	77.3	85.0	79.1
5	NEGHARDLABEL	80.2	80.7	79.9	80.9	87.1	89.6	78.3	77.3	85.2	79.1
6	POSPARTIALLABEL	79.0	80.2	78.0	79.8	72.7	73.2	71.5	68.3	76.7	74.1
7	NEGPARTIALLABEL	83.7	84.1	83.4	83.9	90.6	92.8	80.9	79.7	**88.5**	**81.8**

Table 4: Main results on SemEval and Re-TACRED.

- System SELF-TRAINING significantly and consistently outperforms SUPERVISED with +5.1 on Micro-F1 (85.5 vs. 80.4) and +4.2 on Macro-F1 (79.1 vs. 74.9) in average. The results indicate that self-training with sampling confident data is effective for relation extraction in low-resource scenarios.

- Systems HARD PSEUDO-LABEL, SOFT PSEUDO-LABEL and NEGHARDLABEL which employ the ambiguous data can not achieve consistent improvement over the SELF-TRAINING, demonstrating that it is challenging to achieve the consistent improvement with the ambiguous data.

- Our final system NEGPARTIALLABEL significantly outperforms the SELF-TRAINING on both datasets with +3.0 on Micro-F1 (88.5 vs. 85.5) and +2.7 on Macro-F1 (81.8 vs. 79.1) in average, demonstrating the the effectiveness and the versatility of the proposed approach.

- Unfortunately, the proposed POSPARTIALLABEL suffers from the performance degradation. Ideally, the effect of the positive training should be equivalent to negative training for fully annotated instances. With our partial label setting, each ambiguous instance in the positive training contains only one ground truth label and other candidates are false positive labels. Noises induced by the false positive labels makes the positive training difficult to coverage during model training.

4 Discussion

In this section, we further analyze the results of our final system NEGPARTIALLABEL.

4.1 Effect of Different Partial Label Size N

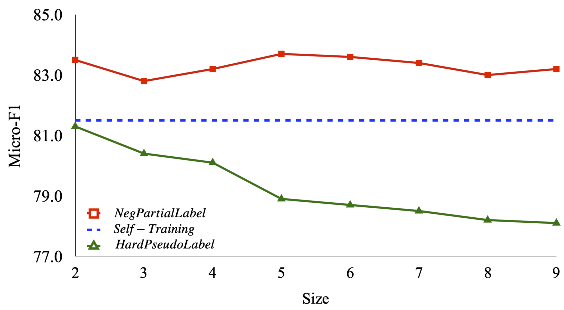


Figure 3: Effect of different partial label sizes.

To analyse the effect of partial label size N in sampling ambiguous instances, we conduct experiments on the development set of SemEval with N from 2 to 9.

The results of averaged micro F1 scores are shown in Figure 3. As a comparison, the experiments of SELF-TRAINING and HARD PSEUDO-LABEL are also conducted. It is clear that the performance of HARD PSEUDO-LABEL decreases as N becomes larger because it introduces more unconfident instances. As for NEGPARTIALLABEL, we can find that the performance is not sensitive to N. A slight advantage is achieved when $N = 5$ which is exactly half of the total number of relations. This indicates that our partial labeling method for sampling ambiguous instances can mostly ensure the assumption that the correct label is in the candidate set.

4.2 The Amount of Ambiguous Data

In order to figure out the relevance between the number of ambiguous instances and performance changes of each relation, we analyse the results on the test set of SemEval. The number of am-

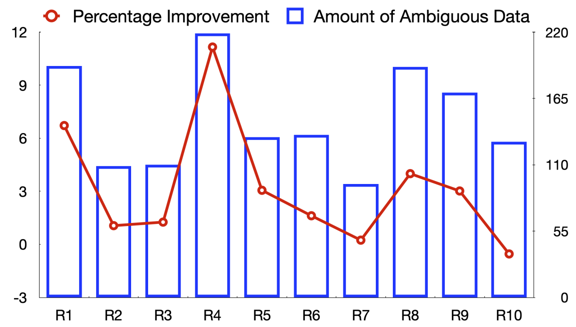


Figure 4: Correlation between amount of ambiguous data and percentage improvement for each relation.

biguous instances for each relation is calculated by a weighted sum. For example, an ambiguous instance with a candidate set of 3 relations gives a contribution of 1/3 for each relation in candidates.

Figure 4 shows the results where the histogram means the number of ambiguous instances for relations and the line chart represents the percentage performance improvement. We can find that the amount of ambiguous data and the improvements have a positive correlation. This indicates that sampling ambiguous data in self-training is useful.

5 Related Work

Self-Training. Self-training is one of the most commonly used approaches for exploiting unlabeled data and has a long history (Scudder, 1965; Yarowsky, 1995; McClosky et al., 2006; Lee et al., 2013). With the development of neural network models and the growth of demand for labeled data, self-training becomes more popular. In neural machine translation, self-training is used to obtain synthetic parallel data (Zhang and Zong, 2016; Wu et al., 2019; Jiao et al., 2021). In computer vision, Xie et al. (2020) proposes noise student training in a teacher-student self-training framework. Zoph et al. (2020) studies self-training in object detection and segmentation and the results show that self-training can often help training a better model. Moreover, self-training works well with other data augmentation methods (Sohn et al., 2020; Du et al., 2021; Vu et al., 2021). In this work, we apply self-training to exploit unlabeled data for low-resource relation extraction. The main difference is that we propose to take a partially labeling strategy for low-confident instances, while they are often discarded in the previous studies.

Partial Label Learning. The definition of partial label in this work is a candidate set of labels is provided for an instance in a multi-class classification task (Cour et al., 2011). This is different from that in sequence labeling tasks (Li et al., 2014; Yang et al., 2018) and multi-label multi-class classification tasks (Xie and Huang, 2018; Huynh and Elhamifar, 2020). In order to learn from partially labeled instances, many researchers have proposed various methods to deal with this problem (Hüllermeier and Beringer, 2006; Nguyen and Caruana, 2008; Cour et al., 2011). Recently, with the help of self-training, Feng and An (2019) proposes a self-guided retraining method to learn from partially labeled data. Besides, Yan and Guo (2020) also proposes to recalculate the confidence of labels in a candidate set by taking the current model as a teacher. However, the partial label problem in this work comes from the assumption of ambiguous instances in self-training. Inspired by the idea of negative training proposed by Kim et al. (2019) to learn from noisy labels in image classification, this paper proposes to use a partial labeling strategy for the ambiguous data, and then apply negative training with them for self-training relation extraction in a low-resource scenario.

6 Conclusion

In this paper, we propose a novel self-training approach for Low-Resource Relation Extraction which makes full use of the auto-annotated data. According to the probabilities predicted by the teacher model, we classify the auto-annotated data into three sets: confident set, ambiguous set, and hard set. During training, we consider the ambiguous set which is often discarded by the previous studies. Since the annotation of ambiguous instances contains noise, we propose a new negative training method to fit the situation well. With proper joint training, the confident set and the ambiguous set are combined to improve the system. Finally, the experimental results show that our proposed system consistently outperforms the baseline systems.

References

Timothee Cour, Ben Sapp, and Ben Taskar. 2011. Learning from partial labels. *The Journal of Machine Learning Research*, 12:1501–1536.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Jingfei Du, Édouard Grave, Beliz Gunel, Vishrav Chaudhary, Onur Celebi, Michael Auli, Veselin Stoyanov, and Alexis Conneau. 2021. Self-training improves pre-training for natural language understanding. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5408–5418.

Lei Feng and Bo An. 2019. Partial label learning with self-guided retraining. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3542–3549.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid Ó Séaghdha, Sebastian Padó, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2010. Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 33–38.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2019. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*.

Eyke Hüllermeier and Jürgen Beringer. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439.

Dat Huynh and Ehsan Elhamifar. 2020. Interactive multi-label cnn learning with partial labels. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9423–9432.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, Michael R. Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *ACL/IJCNLP*.

Youngdong Kim, Junho Yim, Juseung Yun, and Junmo Kim. 2019. Nlnl: Negative learning for noisy labels. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 101–110.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Dong-Hyun Lee et al. 2013. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, page 896.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880.

Cheng Li and Ye Tian. 2020. Downstream model design of pre-trained language model for relation extraction task. *arXiv preprint arXiv:2004.03786*.

Shoushan Li, Zhongqing Wang, Guodong Zhou, and Sophia Yat Mei Lee. 2011. Semi-supervised learning for imbalanced sentiment classification. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Zhenghua Li, Min Zhang, and Wenliang Chen. 2014. Ambiguity-aware ensemble training for semi-supervised dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 457–467.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Fan Luo, Ajay Nagesh, Rebecca Sharp, and Mihai Surdeanu. 2019. Semi-supervised teacher-student architecture for relation extraction. In *Proceedings of the Third Workshop on Structured Prediction for NLP*, pages 29–37.

Ruotian Ma, Tao Gui, Linyang Li, Qi Zhang, Yaqian Zhou, and Xuanjing Huang. 2021. Sent: Sentence-level distant relation extraction via negative training. *arXiv preprint arXiv:2106.11566*.

David McClosky, Eugene Charniak, and Mark Johnson. 2006. Effective self-training for parsing. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 152–159.

Alexander Mey and Marco Loog. 2016. A soft-labeled self-training approach. In *2016 23rd International Conference on Pattern Recognition (ICPR)*, pages 2604–2609. IEEE.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2010. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - ACL-IJCNLP '09*, page 1003.

Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. 2019. Robustness to adversarial perturbations in learning from incomplete data. *Advances in Neural Information Processing Systems*, 32:5541–5551.

Nam Nguyen and Rich Caruana. 2008. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 551–559.

Avital Oliver, Augustus Odena, Colin A Raffel, Ekin Dogus Cubuk, and Ian Goodfellow. 2018. Realistic evaluation of deep semi-supervised learning algorithms.

Advances in Neural Information Processing Systems, 31:3235–3246.

Longhua Qian, Guodong Zhou, Fang Kong, and Qiaoming Zhu. 2009. Semi-supervised learning for semantic relation classification using stratified sampling strategy. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 1437–1445.

Henry Scudder. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3):363–371.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*.

Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. 2020. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in Neural Information Processing Systems*, 33.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the tacred dataset. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13843–13850.

Tu Vu, Minh-Thang Luong, Quoc V Le, Grady Simon, and Mohit Iyyer. 2021. Strata: Self-training with task augmentation for better few-shot learning. *arXiv preprint arXiv:2109.06270*.

Hong Wang, Christfried Focke, Rob Sylvester, Nilesh Mishra, and William Wang. 2019. Fine-tune bert for docred with two-step process. *arXiv preprint arXiv:1909.11898*.

Lijun Wu, Yiren Wang, Yingce Xia, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2019. Exploiting monolingual data at scale for neural machine translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4207–4216.

Ming-Kun Xie and Sheng-Jun Huang. 2018. Partial multi-label learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. 2020. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698.

Yan Yan and Yuhong Guo. 2020. Partial label learning with batch label correction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 6575–6582.

Yaosheng Yang, Wenliang Chen, Zhenghua Li, Zhengqiu He, and Min Zhang. 2018. Distantly supervised ner with partial annotation learning and reinforcement learning. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2159–2169.

David Yarowsky. 1995. Unsupervised word sense disambiguation rivaling supervised methods. In *33rd annual meeting of the association for computational linguistics*, pages 189–196.

Junjie Yu, Tong Zhu, Wenliang Chen, Wei Zhang, and Min Zhang. 2020. Improving relation extraction with relational paraphrase sentences. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1687–1698, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45.

GuoDong Zhou, JunHui Li, LongHua Qian, and QiaoMing Zhu. 2008. Semi-supervised learning for relation extraction. In *Proceedings of the Third International Joint Conference on Natural Language Processing: Volume-I*.

GuoDong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting of the association for computational linguistics (acl'05)*, pages 427–434.

Barret Zoph, Golnaz Ghiasi, Tsung-Yi Lin, Yin Cui, Hanxiao Liu, Ekin Dogus Cubuk, and Quoc Le. 2020. Rethinking pre-training and self-training. *Advances in Neural Information Processing Systems*, 33.