Semi-supervised Relation Extraction via Incremental Meta Self-Training

Xuming Hu[1], **Chenwei Zhang**[2†], **Fukun Ma**[1], **Chenyao Liu**[1], **Lijie Wen**[1†], **Philip S. Yu**[1,3]
[1]Tsinghua University, [2]Amazon, [3]University of Illinois at Chicago
[1]{hxm19,mafk19,liucy19}@mails.tsinghua.edu.cn
[1]wenlj@tsinghua.edu.cn [2]cwzhang@amazon.com [3]psyu@uic.edu

Abstract

To alleviate human efforts from obtaining large-scale annotations, Semi-Supervised Relation Extraction methods aim to leverage unlabeled data in addition to learning from limited samples. Existing self-training methods suffer from the gradual drift problem, where noisy pseudo labels on unlabeled data are incorporated during training. To alleviate the noise in pseudo labels, we propose a method called MetaSRE, where a Relation Label Generation Network generates quality assessment on pseudo labels by (meta) learning from the successful and failed attempts on Relation Classification Network as an additional meta-objective. To reduce the influence of noisy pseudo labels, MetaSRE adopts a pseudo label selection and exploitation scheme which assesses pseudo label quality on unlabeled samples and only exploits high-quality pseudo labels in a self-training fashion to incrementally augment labeled samples for both robustness and accuracy. Experimental results on two public datasets demonstrate the effectiveness of the proposed approach. Source code is available[1].

1 Introduction

Relation extraction plays a key role in transforming massive corpus into structured triplets (subject, relation, object). For example, given a sentence: *The song was composed for a famous Brazilian musician*, we can extract a relation PRODUCT-PRODUCER between two entities *song* and *musician*. These triples can be used in various downstream applications such as web search, sentiment analysis and question answering. Current relation extraction methods (Zeng et al., 2016; Zhang et al., 2017) can discover the semantic relation that holds between two entities under supervised learning. However, these methods typically require lots of manually labeled data for model training. While in practice, these labeled data would be labor-intensive to obtain and error-prone due to human subjective judgments.

A lot of work is being explored to alleviate the human supervision in relation extraction. Distant Supervision methods (Mintz et al., 2009; Zeng et al., 2015) leverage external knowledge bases to obtain annotated triplets as the supervision. These methods make a strong assumption that the relation between entity pairs should not depend on the context, which leads to context-agnostic label noises and sparse matching results. Semi-Supervised Relation Extraction method aims to leverage large amounts of unlabeled data to augment limited labeled data. There are two major ways to make fully use of unlabeled data, i.e., *self-ensembling* method and *self-training* method. Self-ensembling methods (Miyato et al., 2018) assume that predictions on the unlabeled data by the model should remain unchanged, even if there are perturbations in the model parameters or training data. Self-ensembling methods usually suffer from insufficient supervision – when labeled data is limited, the model is

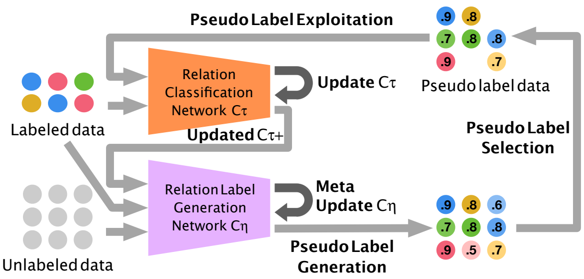

Figure 1: Semi-supervised Relation Extraction via Incremental Meta Self-Training. Relation Classification Network (RCN) uses both labeled data and unlabeled data with pseudo labels generated by the Relation Label Generation Network (RLGN). RLGN will be meta optimized by RCN using the labeled data and leverages pseudo label selection/exploitation scheme to obtain high-quality pseudo labels for RCN.

[1] https://github.com/THU-BPM/MetaSRE
†Corresponding Authors.

Findings of the Association for Computational Linguistics: EMNLP 2021, pages 487–496
November 7–11, 2021. ©2021 Association for Computational Linguistics

reluctant to acquire new relation knowledge that could be learned from the unlabeled data and thus impede further improvements. On the contrary, self-training methods (Rosenberg et al., 2005) improve the predictive ability of the model by obtaining high-confidence labels from unlabeled data incrementally, and retraining the model on the updated labeled data. However, using self-training directly may introduce noisy pseudo labels inevitably, which hurts the model performance, known as the gradual drift problem (Liu et al., 2019; Li and Qian, 2020).

To alleviate the noise in pseudo labels, we design a method that can generate high-quality pseudo labels from the unlabeled dataset as additional supervision. We propose a semi-supervised learning framework which adopts meta-learning during pseudo label generation and automatically learns to reduce the influence of noisy pseudo labels. We also adopt a pseudo label selection and exploitation scheme during each self-training step to take full advantage of high-quality pseudo labels. As shown in Figure 1, the proposed framework has two networks, Relation Classification Network (RCN) and Relation Label Generation Network (RLGN). RCN encodes entity pair representations based on the context in which they are mentioned, and train a classifier with both labeled data and pseudo-labeled data generated by RLGN to classify the representations into relation categories. RLGN leverages pseudo label selection/exploitation scheme to calibrate confidence scores and obtain high-quality pseudo labels from unlabeled data for RCN. RLGN is meta optimized by RCN using the labeled data.

The main contributions of this work are as follows:

- We propose a novel semi-supervised relation extraction framework `MetaSRE`, which adopts a meta learner network to prevent the model from drifting due to label noise and enables robust iterative self-training.

- We develop a label selection and exploitation scheme that explores all unlabeled samples and exploits high-confidence ones to get pseudo labels, for effective and efficient self-training.

- We show that the semi-supervised model outperforms strong baselines and extensive experiments validate the effectiveness of the proposed model.

2 Proposed Model

The proposed Incremental Meta Self-training Relation Extraction (`MetaSRE`) consists of two networks: Relation Classification Network (RCN) and Relation Label Generation Network (RLGN). As shown in Figure 1, the input of the RCN is labeled data, which consists of sentences and relation mentions: $[Sentence, Entity_1, Entity_2, Relation]$. The goal of the network is to conduct relation classification using both labeled data and unlabeled ones with high-quality pseudo labels generated by the RLGN. In order to alleviate gradual drifts in traditional self-training scheme, the RLGN will be meta-optimized by RCN using the labeled data to insure pseudo label quality. RLGN leverages pseudo label selection scheme to obtain high-confidence pseudo labels with less noises. RCN exploits these pseudo labels in an iterative fashion and further improves its classification performance.

2.1 Relation Classification Network (RCN)

The main purpose of RCN is to extract the relational representation of context from sentences, and classify these features to get the corresponding relations. We assume that the named entities in the sentences have been recognized and marked in advance, and we need to focus on the binary relations which involve two entities.

Contextualized Relation Encoder

Since the relation between two entities in a sentence is often contained in the context of the two entities, the relation is carried through the contexts in which entities are expressed. In this work, we use pretrained deep bi-directional transformers networks: BERT (Devlin et al., 2019) to extract contextualized entity features.

For an input sentence containing a relation mention, two entities $Entity_1$ and $Entity_2$ are marked in advance. We follow the labeling mechanism adopted by Zhang and Wang (2015); Soares et al. (2019) to enhance the position information of entities. For each sentence $X = [x_1, .., x_T]$, four reserved tokens $[E1_{start}]$, $[E1_{end}]$, $[E2_{start}]$, $[E2_{end}]$ are inserted to indicate entities' start and end positions:

$$\begin{aligned} X = \big[&x_1, ..., [E1_{start}], x_i, ..., x_{j-1}, [E1_{end}], \\ &..., [E2_{start}], x_k, ..., x_{l-1}, [E2_{end}], ..., x_T\big]. \end{aligned} \tag{1}$$

The updated sentence is used as the input of the Contextualized Relation Encoder. Instead of using the sentence-level output $[CLS]$ from BERT, we use the entity-level output $[E1_{start}]$, $[E2_{start}]$ and concatenate them and obtain a fixed-length vector $\mathbf{h} \in \mathbb{R}^{2 \cdot h_R}$ as the contextualized entity pair representation:

$$\mathbf{h} = [\mathbf{h}_{[E1_{start}]}, \mathbf{h}_{[E2_{start}]}]. \tag{2}$$

Relation Classification Once we obtain contextualized entity pair representations, the contextualized relation encoder is used to classify these representations. When the labeled data that we used to train the classifier is insufficient, the classifier may have limited generalization ability. High-quality pseudo labels generated on unlabeled data can help improve the classification performance. However, not all unlabeled data are helpful. In our model, in order to mitigate such noises, we construct the RLGN specifically to associate unlabeled sentences with high-quality pseudo labels. This network will be introduced in detail in Section 2.2.

RCN learns to predict the right golden labels and pseudo labels. More specifically, we have:

$$\mathbf{l}_n = C_\tau(X_{n,E1,E2}), \tag{3}$$

where $\mathbf{l}_n$ is a probability distribution over the number of relations. C_τ consists of the Contextualized Relation Encoder module which converts $X_{n,E1,E2}$ into $\mathbf{h} = [\mathbf{h}_{[E1_{start}]}, \mathbf{h}_{[E2_{start}]}]$ and a fully connected dense layer that uses $\mathbf{h}$ for classification. There are a total of N golden labels and M pseudo labels: $S_{all} = \{g_1, g_2, ..., g_N, p_1, p_2, ..., p_M\}$. To distinguish pseudo labels from golden labels when optimizing τ, we adopt the following classification loss:

$$\begin{aligned} \mathcal{L}_{C_\tau} = \sum_{n=1}^{N} loss(\mathbf{l}_n, \text{one\_hot}(g_n)) + \\ \sum_{m=1}^{M} w_m \cdot loss(\mathbf{l}_m, \text{one\_hot}(p_m)), \end{aligned} \tag{4}$$

where $loss$ is the cross entropy loss function. $\mathbf{l}_n$ is the inferred probability distribution of the labeled data. $\mathbf{l}_m$ is the inferred probability distribution of the unlabeled data via pseudo labeling. $\text{one\_hot}(\cdot)$ returns an one-hot vector indicating the label assignment. Since we wish the module to confide less in pseudo labels than golden labels, therefore when optimizing parameter τ in classifier C_τ, the loss terms on pseudo labels are used with a confidence coefficient $0 \leq w_m \leq 1$, which is generated by the RLGN. Note that in the first iteration, the second term for pseudo labels in Eq. 4 is not applicable until subsequent iterations where pseudo labels are generated.

2.2 Relation Label Generation Network (RLGN)

We aim to make full use of unlabeled data to improve the classification quality of RCN. In this section, RLGN is introduced to generate pseudo labels for unlabeled data with a simple yet effective meta-update scheme.

Unlike the traditional self-supervised scheme, `MetaSRE` does not directly use the RCN to classify unlabeled sentences with pseudo labels. As illustrated in Figure 1, we construct a RLGN to generate pseudo labels, which has the same architecture as the RCN but is trained separately as a meta learner to label the unlabeled sentences and learn more about the distribution of relation mentions on pseudo labels through the RCN.

The reason why we discard the classification network and retrain another network is to prevent the noise contained in the generated pseudo labels which would lead the sentence feature distribution to drift gradually (Zhang et al., 2016; Liu et al., 2019). For example, when the classifier incorrectly gives the unlabeled sentence: *The* ***song*** *was composed for a famous Brazilian* ***musician*** a false pseudo label: CONTENT-CONTAINER instead of PRODUCT-PRODUCER, this falsely-labeled sentence will be added into pseudo label mentions, which will accumulate errors in the subsequent training.

To address this issue, we adopt a simple yet effective schema: we let the RLGN learn to effectively assess the quality of pseudo labels by (meta) learning from the successful and failed attempts using the most updated RCN as an additional meta-objective. This schema can be seen as a form of meta-learning. In other words, the meta objective of the RLGN is to perform a derivative over the parameters on RCN. In order to prevent noises in the generated pseudo labels from contaminating the overall objective, RLGN is tuned to generate pseudo labels using only the labeled data. These two networks have the same network structure but are initialized separately and trained completely independently. To distinguish, we denote the parameters of the RLGN as η. The meta objective is defined as follows:

$$\underset{\eta}{\text{argmin}}\ loss\big(C_{\tau^+}(X_{n,E1,E2}), \text{one\_hot}(g_n)\big), \tag{5}$$

where τ^+ represents the parameters of the RCN after one gradient update using the loss in Equation 4, n comes from N golden labels: $S_G = \{g_1, g_2, ..., g_N\}$:

$$\tau^+ \leftarrow \tau - \alpha \nabla_\tau \mathcal{L}_{C_\tau}, \tag{6}$$

where α is the learning rate. The trick in meta objective is to use the updated parameters τ^+ to calculate the derivative and update η. This can make the η learn more about the procedure of τ to learn pseudo labels. This trick was also adopted in other meta-learning frameworks such as Finn et al. (2017), Zhang et al. (2018) and Liu et al. (2019).

Therefore, the final optimization function is:

$$\mathcal{L}_{C_\eta} = \min_\eta \sum_{n=1}^{N} \big(loss((C_{\tau^+}(X_{n,E1,E2}), \\ \text{one\_hot}(g_n))\big) \tag{7}$$

After we optimize RLGN after an epoch, there are broadly two methods to make fully use of unlabeled data, i.e., *self-ensembling* method and *self-training* method. As we illustrate in Section 1, we adopt the self-training method and use RLGN to incrementally augment labeled data via iterative pseudo label generation.

2.3 Pseudo label Selection and Exploitation

Selection

In the selection stage, RLGN selects the potential pseudo label for each unlabeled sample. We denote m' for each unlabeled sample and $X_{m'}$ is one unlabeled data we use to obtain the pseudo label for. We consider the relation that corresponds to the maximum probability after softmax $\text{argmax}(C_\eta(X_{m',E1,E2}))$ as the pseudo label and use the corresponding probability as the confidence score. Since not all generated pseudo labels are equally informative, we sort all pseudo labels according to the confidence score in a descending order and select the top $Z\%$ high-confidence pseudo labels as the final M pseudo labels, denoted as $S_P = \{p_1, p_2, ..., p_M\}$.

Exploitation

When we complete pseudo label selection and obtain high-confidence pseudo labels, we need to exploit these pseudo labels. Although the obtained pseudo labels are of high confidence, simply treating them as golden labels may introduce noises and deteriorate the RCN robustness. When adding pseudo labels to the labeled data S_G which will be used to optimize the RCN parameters, we use the maximum probability value from the output of RLGN as the weight:

$$w_m = \max_m(C_\eta(X_{m,E1,E2})), \tag{8}$$

where m comes from the final M exploited pseudo labels. w_m is not updated during the optimization of RCN.

Incremental Self-Training

Pseudo labeling all unlabeled data at once using limited labeled data is not ideal. In this work, we explore an incremental way to select high-quality pseudo labels and utilize them in batches to let the RCN gradually improve as we obtain more high-quality pseudo labels from the RLGN. The whole Incremental Self-Training workflow is illustrated in Algorithm 1.

Algorithm 1 Incremental Self-Training in `MetaSRE`

```
Require: Labeled data and unlabeled data. Unlabeled data are divided into 10 batches.
1: Train the Classification Network $C_\tau$ using labeled data (Eq. 4).
2: for each batch of $10\%$ unlabeled data do
3:     Meta update the Generation Network $C_\eta$ using labeled data and the updated Classification Network $C_{\tau^+}$ (Eq. 7).
4:     Generate pseudo labels using the Generation Network $C_\eta$ for unlabeled data in this batch.
5:     Use Pseudo Label Selection to select top $Z\%$ high-confidence pseudo labels.
6:     Update the Classification Network $C_\tau$ using labeled data and high-confidence pseudo labels (Eq. 4 & 6).
7: end for
```

3 Experiments

We first introduce datasets, experimental settings and evaluation metrics, and then present the performance comparison with baseline models. A detailed analysis is presented to show the advantages of each module.

3.1 Datasets

We use two public relation extraction datasets with different characteristics in our experiments: (1) **SemEval** (Hendrickx et al., 2010): SemEval 2010 Task 8 provides a standard benchmark dataset and is widely used in the evaluation of relation extraction models. Its training, validation, test set contain 7199, 800, 1864 relation mentions respectively, with 19 relation types in total (including *no_relation*), of which *no_relation* percentage is 17.4%. (2) **TACRED** (Zhang et al., 2017): TAC Relation Extraction Dataset is a large-scale crowd-sourced relation extraction dataset. The corpus is collected from all the prior TAC KBP shared tasks, and follows TAC KBP relation scheme. Its training, validation, test set contain 75049, 25763, 18659 relation mentions respectively, with 42 relation types in total (including *no_relation*), of which *no_relation* percentage is 78.7%.

For the two datasets above, their named entities in the sentences have been recognized and marked in advance. In terms of data characteristics, TACRED is far more complicated than SemEval because it has more relation types and more skewed distribution on negative mentions.

3.2 Baselines and Evaluation metrics

As the proposed framework `MetaSRE` is general to integrate different contextualized relation encoders. We first compare several widely used supervised relation encoders such as **LSTM** (Hochreiter and Schmidhuber, 1997), **PCNN** (Zeng et al., 2015), **PRNN** (Zhang et al., 2017), **BERT** (Devlin et al., 2019) and train them only on the labeled dataset, then adopt the best performing one as the base encoder for `MetaSRE` as well as for other representative semi-supervised approaches for a fair comparison.

We select four representative approaches from different categories of semi-supervised learning methods as our baselines:

- **Self-Training** (Rosenberg et al., 2005) uses a model to predict on unlabeled data recursively, and adds the pseudo labels to the labeled data. The model improves itself by retraining on the updated labeled dataset.
- **Mean-Teacher** (Tarvainen and Valpola, 2017) encourages different variants of the model to make consistent predictions on similar inputs. The model is optimized by perturbation-based loss and training loss jointly.
- **DualRE** (Lin et al., 2019) leverages sentence retrieval as a dual task for relation extraction. The model combines the loss of a prediction module and a retrieval module to obtain the information corresponding to sentences and relations in unlabeled data.
- **MRefG** (Li and Qian, 2020) is the state-of-the-art method that constructs reference graphs, including entity reference, verb reference, and semantics reference to semantically or lexically connect the unlabeled samples to the labeled ones.

Finally, we present another model: **BERT w. gold labels** to demonstrate the upper bound of the performance for all the semi-supervised methods. This model trains the BERT base model using both the different proportion of labeled data and the gold labels of the unlabeled data, which means that all unlabeled data are correctly labeled.

For the evaluation metrics, we consider F1 score as the main metric while precision and recall serve as auxiliary metrics. Note that following Li and Qian (2020), the correct prediction of *no_relation* is ignored.

3.3 Implementation Details

For all datasets, strictly following the settings used in Li and Qian (2020), we divide the training set into labeled and unlabeled sets of various sizes according to the stratified sampling which could ensure the distribution of the label will not change. We sampled 5%, 10%, and 30% of the training set as labeled sets for the SemEval dataset, and the sampled 3%, 10%, and 15% of the training set as labeled set for the TACRED dataset. For both datasets, we sampled 50% of the training set as the unlabeled set. For all models that use unlabeled data in an incremental way, we follow the setting by Li and Qian (2020) and fix the increased amount of unlabeled data as 10% per iteration for a fair comparison, which means the unlabeled dataset will be exhausted after 10 iterations.

For the RCN, in the Contextualized Relation Encoder module, we use the pretrained BERT-Base + Cased as the initial parameter. We use the BERT default tokenizer and set max-length as 128 to preprocess dataset. We use BertAdam with $1e{-}4$ learning rate to optimize the loss. In the Relation Classification module, we use a fully connected layer with the following dimensions: $2 \cdot h_R$-h_R-label_size,

Labeled Data	5%			10%			30%		
Methods	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LSTM	25.34 ± 2.17	20.68 ± 4.01	22.65 ± 3.35	36.93 ± 6.23	29.65 ± 7.13	32.87 ± 6.79	64.80 ± 0.67	62.98 ± 0.66	63.87 ± 0.65
PCNN	42.87 ± 4.56	40.71 ± 4.36	41.82 ± 4.48	53.67 ± 1.52	49.23 ± 2.34	51.34 ± 1.87	64.52 ± 0.59	62.87 ± 0.53	63.72 ± 0.51
PRNN	56.13 ± 1.29	54.52 ± 1.43	55.34 ± 1.08	61.70 ± 1.16	63.61 ± 2.07	62.63 ± 1.42	69.66 ± 2.12	68.72 ± 2.57	69.02 ± 1.01
BERT	73.12 ± 1.34	72.21 ± 1.21	72.71 ± 1.24	74.71 ± 1.02	72.81 ± 0.98	73.93 ± 0.99	79.10 ± 0.89	81.05 ± 0.82	80.55 ± 0.87
Mean-Teacher$_{BERT}$	70.33 ± 3.45	68.55 ± 4.23	69.05 ± 3.89	74.01 ± 1.89	72.08 ± 1.32	73.37 ± 1.42	79.09 ± 0.89	82.23 ± 0.78	80.61 ± 0.81
Self-Training$_{BERT}$	73.10 ± 1.21	70.01 ± 1.83	71.34 ± 1.68	75.54 ± 1.07	73.00 ± 1.18	74.25 ± 1.10	80.92 ± 0.99	82.39 ± 0.67	81.71 ± 0.79
DualRE$_{BERT}$	73.32 ± 1.87	77.01 ± 1.67	74.35 ± 1.76	75.51 ± 1.21	78.81 ± 1.01	77.13 ± 1.10	81.30 ± 0.81	84.55 ± 0.52	82.88 ± 0.67
MRefG$_{BERT}$	73.04 ± 1.43	78.29 ± 1.21	75.48 ± 1.34	76.32 ± 1.00	79.76 ± 0.81	77.96 ± 0.90	81.75 ± 0.78	84.91 ± 0.63	83.24 ± 0.71
MetaSRE$_{BERT}$	**75.59 ± 0.92**	**81.40 ± 0.91**	**78.33 ± 0.92**	**78.05 ± 0.87**	**82.29 ± 0.72**	**80.09 ± 0.78**	**82.01 ± 0.36**	**87.95 ± 0.56**	**84.81 ± 0.44**
BERT w. gold labels	82.43 ± 0.38	84.94 ± 0.21	83.64 ± 0.28	82.87 ± 0.36	86.16 ± 0.32	84.40 ± 0.34	86.28 ± 0.21	87.93 ± 0.27	87.08 ± 0.23

Table 1: Performance on SemEval with various amounts of labeled data and 50% unlabeled data

Labeled Data	3%			10%			15%		
Methods	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
LSTM	40.63 ± 6.42	23.12 ± 4.58	28.68 ± 4.29	50.43 ± 0.97	43.18 ± 1.38	46.79 ± 0.99	55.69 ± 0.57	44.23 ± 0.62	49.42 ± 0.59
PCNN	58.30 ± 7.43	37.78 ± 6.93	44.02 ± 5.23	64.64 ± 7.51	42.10 ± 4.94	50.35 ± 3.28	67.92 ± 0.51	42.09 ± 0.46	52.50 ± 0.39
PRNN	49.12 ± 5.23	33.21 ± 2.39	39.11 ± 1.92	53.71 ± 2.73	51.81 ± 1.72	52.23 ± 1.20	58.10 ± 2.78	51.05 ± 2.00	54.55 ± 1.92
BERT	49.82 ± 4.98	34.21 ± 3.19	41.11 ± 3.88	55.71 ± 2.17	52.81 ± 1.83	54.23 ± 1.67	59.10 ± 1.29	53.05 ± 1.02	56.55 ± 0.82
Mean-Teacher$_{BERT}$	50.04 ± 3.21	**38.45 ± 2.73**	44.34 ± 1.78	56.02 ± 2.17	49.52 ± 1.09	53.08 ± 1.01	57.01 ± 1.02	51.24 ± 1.49	53.79 ± 1.38
Self-Training$_{BERT}$	48.01 ± 1.29	36.99 ± 1.56	42.11 ± 1.04	56.23 ± 0.67	52.02 ± 0.78	54.17 ± 0.53	59.25 ± 0.79	53.91 ± 0.33	56.52 ± 0.40
DualRE$_{BERT}$	57.99 ± 1.77	35.21 ± 1.32	43.06 ± 1.73	60.00 ± 0.92	52.52 ± 0.37	56.03 ± 0.55	61.03 ± 0.54	55.24 ± 0.84	57.99 ± 0.67
MRefG$_{BERT}$	56.31 ± 1.72	36.25 ± 1.22	43.81 ± 1.44	59.25 ± 1.27	51.93 ± 1.88	55.42 ± 1.40	61.02 ± 0.82	**55.61 ± 0.69**	58.21 ± 0.71
MetaSRE$_{BERT}$	**58.96 ± 1.00**	37.66 ± 1.26	**46.16 ± 1.02**	**60.49 ± 0.92**	**53.69 ± 0.58**	**56.95 ± 0.34**	**65.03 ± 0.43**	54.02 ± 0.45	**58.94 ± 0.36**
BERT w. gold labels	65.92 ± 0.37	60.13 ± 0.48	62.93 ± 0.41	67.26 ± 0.29	60.42 ± 0.21	63.66 ± 0.23	68.01 ± 0.28	61.76 ± 0.26	64.69 ± 0.29

Table 2: Performance on TACRED with various amounts of labeled data and 50% unlabeled data.

where $h_R = 768$. Learning rate is set as $1e{-}4$ and warmup to 0.1.

RLGN has the same structure and learning rate as the RCN. We also use the pretrained BERT-Base + Cased as the initial parameter in the Contextualized Relation Encoder module. In the Pseudo label Selection module, each increment will generate 10% of unlabeled data as pseudo labels. $Z\% = 90\%$ is used.

3.4 Main Results

Table 1 and 2 show the experimental results on SemEval and TACRED dataset when adopting various labeled data and 50% unlabeled data. We conduct 5 runs of training and testing then report the mean and standard deviation results. Considering that BERT (Devlin et al., 2019) has achieved the SOTA performance in all base encoders, we adopt BERT as the base encoder for all methods. We can also observe that MRefG achieves the best performance among all the baselines, which is considered as the previous SOTA method. The proposed model MetaSRE outperforms all baseline models consistently on F1. MetaSRE on average achieves 2.18% higher F1 on SemEval dataset and 1.54% higher F1 on TACRED dataset across various labeled data when comparing with MRefG. When considering standard deviation, MetaSRE performs more robust than all the baselines.

From Table 1 and 2, when we fix unlabeled data to 50% of the training set, we can find that with the increasing of labeled data, the performance of the model is effectively improved. In term of less labeled data, MetaSRE performs much better than the previous state-of-the-art model: MRefG. For example, MetaSRE achieves 2.85% higher F1 performance with 5% labeled data on SemEval and 2.35% higher F1 performance with 3% labeled data on TACRED, which indicate the effectiveness of our model under low-resource situations.

Model/F_1 Performance	5%SemEval	30%SemEval	3%TACRED	15%TACRED
MetaSRE	**78.33**	**84.81**	**46.16**	**58.94**
w/o Meta Learning	75.01	82.25	43.12	57.33
w/o Pseudo Label Selection	75.02	83.29	43.96	58.16
w/o Pseudo Label Exploitation	77.52	83.88	44.98	58.28

Table 3: Ablation study of MetaSRE on two datasets.

To investigate how models leverage different amounts of unlabeled data for performance improvement, we fix the amount of labeled data and compare the performance of the model with different amounts of unlabeled data. We report the F1 performance for SemEval and TACRED with 10% labeled data and adopt unlabeled data as 10%, 30%, 50%, 70%. Since the labeled and unlabeled data are from the training set, we can provide up to 70% unlabeled data. From Figure 2, we could see all semi-supervised methods have performance gains by using unlabeled data and MetaSRE achieves consistent and better F1 performance than other baselines under different ratios of unlabeled data.

Ablation Study

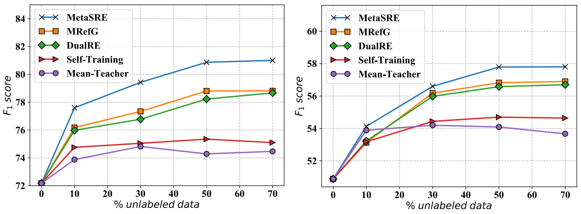

Figure 2: F_1 Performance with various unlabeled data and 10% labeled data on SemEval (left) and TACRED (right).

We conduct ablation study to show the effectiveness of different modules in MetaSRE. MetaSRE w/o Pseudo Label Selection is the proposed model without selecting Top $Z\%$ samples among all pseudo labels according to their confidences and adds all pseudo labels with weightings to the labeled data. MetaSRE w/o Pseudo Label Exploitation treats pseudo labels as having the same confidence score as golden labels, and does not distinguish between pseudo labels and golden labels when optimizing the RLGN. MetaSRE w/o Meta Learning uses the same RCN for both label generation and classification. This is also equivalent to the Self-Training method with the addition of Pseudo Label Selection and Exploitation modules.

From ablation results in Table 3, we found that all modules contribute positively to the improved performance. More specifically, Meta Learning and Pseudo Label Selection modules do impact the performance: performances on MetaSRE w/o Meta Learning and MetaSRE w/o Pseudo Label Selection are deteriorated by 2.63% and 1.88% on F1 averaged over two datasets. Pseudo Label Exploitation gives 0.90% performance boost in average over F1 on two datasets when comparing with the hard-weighting alternative.

3.5 Analysis and Discussion

Effectiveness of Meta Learning

The main purpose of the Meta Learning module is to generate pseudo labels with less noise and higher accuracy, which could prevent the RCN from drifting gradually. From the Ablation Study we can conclude the Meta Learning module can effectively improve the results. In order to explore how and why this module is effective, we use MetaSRE and MetaSRE w/o Meta Learning to explore the quality of the generated pseudo labels on the two datasets respectively. We sample the SemEval dataset with 30% labeled data and the TACRED dataset with 15% labeled data, and both with 50% unlabeled data.

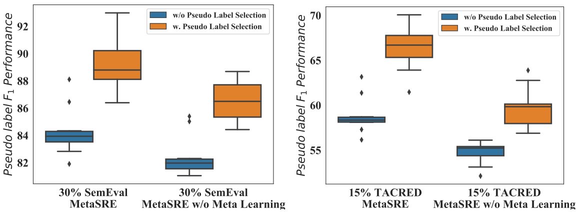

Figure 3: Pseudo label F_1 Performance with different modules based on SemEval (left) and TACRED (right).

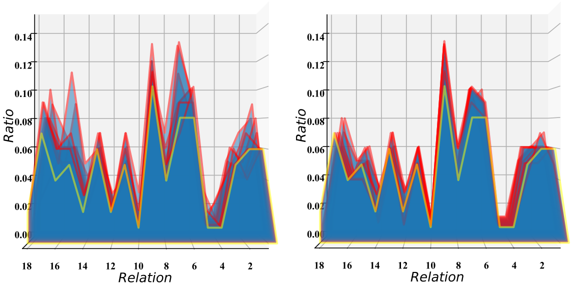

Figure 4: Pseudo label distributions generated by different iterations (red) of MetaSRE w/o Meta Learning (left) and MetaSRE (right). Yellow line is the gold label distribution.

From Figure 3, we can find that: (1) No matter whether it leverages Pseudo Label Selection or not, the Meta Learning module can continuously generate pseudo labels with higher quality. (2) After the Pseudo Label Selection, the quality of the pseudo labels generated by the Meta Learning module will be further improved. (3) On the TACRED dataset which has more skewed label distributions, Meta Learning module improves the quality of pseudo labels even more, which means the Meta Learning module could help MetaSRE learn more accurate relational labels.

To validate meta learner network could prevent MetaSRE from drifting due to label noise and enables robust iterative self-training, in Figure 4, we use yellow line to represent the gold label distributions and red lines to represent the pseudo label distributions which are generated by MetaSRE and MetaSRE w/o Meta Learning on the unlabeled data in all 10 iterations. Both models are evaluated with 10% labeled data and 50% unlabeled data on the SemEval dataset. Note that following Li and Qian (2020), we use stratified sampling to obtain unlabeled data for each iteration, so they all share the same gold label distribution. From Figure 4, we observe that with Meta Learning, the pseudo label distribution is closer to the gold label distribution, with less drift, and thus contributes to the robust iterative self-training schema.

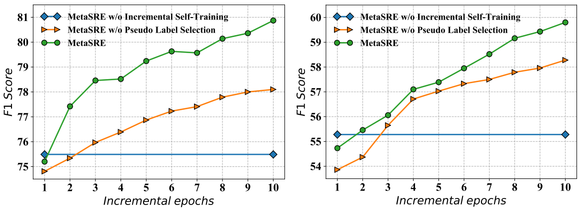

Figure 5: Effectiveness of Incremental Self-Training module on SemEval (left) and TACRED (right).

Dataset \ $Z\%$	60	70	80	90	100
SemEval	74.89	75.26	78.36	**80.09**	77.28
TACRED	53.38	55.12	56.60	**56.95**	55.56

Table 4: F_1 Performance with different $Z\%$ on 10% labeled data of SemEval and TACRED.

Effectiveness of Incremental Self-Training

During each iteration of the Incremental Self-Training, the Pseudo Label Selection module selects the top 90% high-confidence pseudo labels which contain less noise and abandon the rest. From Figure 3, we could observe that on both SemEval and TACRED dataset, the selection mechanism achieves nearly 5% - 10% F1 performance improvement compared to the full, unfiltered pseudo labels. The improvement is higher on the TACRED dataset, which has a more skewed label distribution.

To demonstrate that high-confidence pseudo labels selected by Pseudo Label Selection could be helpful for robust self-training, we compare Incremental Self-Training that uses 10% unlabeled data with high-confidence pseudo labels in each iteration, with the same ratio unfiltered pseudo labels on unlabeled data. We sample the SemEval with 10% and the TACRED with 15% labeled data, both with 50% unlabeled data. From Figure 5, when comparing with `MetaSRE` w/o Pseudo Label Selection (brown), the Incremental Self-Training handles noises in pseudo label selection & exploitation more effectively, leading to robust and consistently better performance.

To explore the effectiveness of the incremental scheme, we report the F1 on `MetaSRE` w/o Incremental Self-Training (blue), which adds all the pseudo labels to the labeled data at once to optimize RCN. From Figure 5, `MetaSRE` (green) could obtain more competitive results compared with blue line, which is mainly because `MetaSRE` leverages the most up-to-date RLGN to generate higher quality pseudo labels and incrementally improves the classification performance.

Hyperparameter Analysis

We study the ratio of top $Z\%$ in Pseudo Label Selection module. We sample $Z\%$ from 60% to 100% and report the F_1 Performance of SemEval and TACRED with the same 10% labeled data. Note that our hyperparameter $Z\%$ is decided based on the validation set and we report the $Z\%$ analysis on the test set here for better demonstration. From Table 4, the fluctuation results indicate that both quality and coverage of pseudo labels will impact performance. Using a high $Z\%$ will introduce low-quality pseudo labels that are noisy-prone, causing the gradual drift problem. Low $Z\%$ will cause the low coverage of pseudo labels on some relations, affecting the recall.

4 Related Work

Relation extraction focuses on predicting the relation between two entities in a sentence. Recent literature leverage deep neural networks to extract features about two entities from sentences, and classify them into specific relations. Relation extraction methods are often formulated in a supervised setting (Zeng et al., 2015; Guo et al., 2020; Nan et al., 2020) and require manual annotations on large amounts of corpora, which is labor-intensive to obtain.

Semi-supervised learning methods have received attention recently, since these methods require fewer labeled data and generate pseudo labels by re-training model and improve the performance iteratively. Two major categories of semi-supervised learning methods are related to our problem. One major category is the self-ensembling method (French et al., 2017; Xu et al., 2019), which is based on the assumption that pseudo label distributions should remain unchanged even if the model parameters and instances have small perturbations. In this case, various models based on consistent data could be used to improve the performance by co-training each other (Tarvainen and Valpola, 2017). However, this method relies heavily on the quality and quantity of labeled data. In our task, the improvement of the model by this method is limited.

Another category is the self-training method, the work proposed by Rosenberg et al. (2005) incrementally generates pseudo labels from unlabeled data, and uses these pseudo labels to enhance the classification ability of the model. However, this method often needs to endure semantic drift (Curran et al., 2007; Zhang et al., 2016; Hu et al., 2020; Li and Qian, 2020), which means the noise generated by pseudo labels in each iteration will be continuously strengthened, causing the model to deviate from the global minimum point. Li et al. (2019) tries to fine-tune the model on the labeled data in each iteration to avoid noise and construct a prototypical network to exploit pseudo labels. However, the model is still influenced by the noises with the unchanged pseudo labels and spreads these noises to the generated pseudo labels. In our work, we adopt a meta learner network to prevent the model from drifting due to label noise and enables robust iterative self-training.

5 Conclusion

In this paper, we propose a semi-supervised learning model MetaSRE for relation extraction. Different from conventional semi-supervised models which directly adopt the classification network to classify unlabeled data into pseudo labels, our model proposes a novel meta-learning-based self-training network to reduce the noises contained in pseudo labels and avoid the model gradual drift. Comparing with using pseudo labels directly, our model leverages pseudo label selection and exploitation scheme to further select high-confidence pseudo labels with low noises. Experiments on two popular benchmarks show the effectiveness and consistent improvements over baselines.

Acknowledgments

We thank the reviewers for their valuable comments. The work was supported by the National Key Research and Development Program of China (No. 2019YFB1704003), the National Nature Science Foundation of China (No. 71690231 and No. 62021002), NSF under grants III-1763325, III-1909323, III-2106758, SaTC-1930941, Tsinghua BNRist and Beijing Key Laboratory of Industrial Bigdata System and Application.

References

James R Curran, Tara Murphy, and Bernhard Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proc. of PACL*, volume 6, pages 172–180. Bali.

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

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proc. of ICML*, pages 1126–1135. JMLR. org.

Geoffrey French, Michal Mackiewicz, and Mark Fisher. 2017. Self-ensembling for visual domain adaptation. *arXiv preprint arXiv:1706.05208*.

Zhijiang Guo, Guoshun Nan, Wei Lu, and Shay B. Cohen. 2020. Learning latent forests for medical relation extraction. In *Proc. of IJCAI*, pages 3651–3657. ijcai.org.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O 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 *Proc. of the 5th International Workshop on Semantic Evaluation*, pages 33–38.

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

Xuming Hu, Lijie Wen, Yusong Xu, Chenwei Zhang, and S Yu Philip. 2020. Selfore: Self-supervised relational feature learning for open relation extraction. In *Proc. of EMNLP*, pages 3673–3682.

Wanli Li and Tieyun Qian. 2020. Exploit multiple reference graphs for semi-supervised relation extraction. *arXiv preprint arXiv:2010.11383*.

Xinzhe Li, Qianru Sun, Yaoyao Liu, Qin Zhou, Shibao Zheng, Tat-Seng Chua, and Bernt Schiele. 2019. Learning to self-train for semi-supervised few-shot classification. In *NeurIPS*, pages 10276–10286.

Hongtao Lin, Jun Yan, Meng Qu, and Xiang Ren. 2019. Learning dual retrieval module for semi-supervised relation extraction. In *Proc. of WWW*, pages 1073–1083.

Shikun Liu, Andrew Davison, and Edward Johns. 2019. Self-supervised generalisation with meta auxiliary learning. In *NeurIPS*, pages 1677–1687.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. Distant supervision for relation extraction without labeled data. In *Proc. of ACL*, pages 1003–1011.

Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, and Shin Ishii. 2018. Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE TPAMI*, 41(8):1979–1993.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *Proc. of ACL*, pages 1546–1557.

Chuck Rosenberg, Martial Hebert, and Henry Schneiderman. 2005. Semi-supervised self-training of object detection models. *WACV/MOTION*, 2.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. 2019. Matching the blanks: Distributional similarity for relation learning. *arXiv preprint arXiv:1906.03158*.

Antti Tarvainen and Harri Valpola. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *NeurIPS*, pages 1195–1204.

Yonghao Xu, Bo Du, Lefei Zhang, Qian Zhang, Guoli Wang, and Liangpei Zhang. 2019. Self-ensembling attention networks: Addressing domain shift for semantic segmentation. In *Proc. of AAAI*, volume 33, pages 5581–5588.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proc. of EMNLP*, pages 1753–1762.

Wenyuan Zeng, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2016. Incorporating relation paths in neural relation extraction. *arXiv preprint arXiv:1609.07479*.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Dongxu Zhang and Dong Wang. 2015. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*.

Yabin Zhang, Hui Tang, and Kui Jia. 2018. Fine-grained visual categorization using meta-learning optimization with sample selection of auxiliary data. In *Proc. of ECCV*, pages 233–248.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proc. of EMNLP*, pages 35–45.