Self-distilled BERT with Instance Weights for Denoised Distantly Supervised Relation Extraction

Anonymous ACL submission

Abstract

The widespread existence of wrongly-labeled instances is a challenge to distantly supervised relation extraction. Most of the previous works use the features from the final output of the encoder and are trained in a bag-level setting. However, intermediate layers of BERT encode a rich hierarchy of linguistic information which is helpful in identifying wrongly-labeled instances. Besides, sentence-level training better utilizes the information than bag-level training, as long as combined with effective noise alleviation. In this work, we design a novel instance weighting mechanism integrated with the self-distilled BERT backbone to enable denoised sentence-level training. Our method aims to alleviate noise and prevent overfitting through dynamic adjustment of learning priorities during self-distillation. Experiments on both held-out and manual datasets indicate that our method achieves state-of-the-art performance and consistent improvements over the baselines.

1 Introduction

Distantly Supervised Relation Extraction (DSRE) (Mintz et al., 2009) is designed to automatically annotate the sentences mentioning the entity pairs, which enables a significant way for constructing large-scale datasets. However, distant supervision (DS) works under an unrealistic assumption that all sentences mentioning the same entity pair express the same relation. This introduces many noisy (wrongly labeled) sentences into the dataset. To tackle this challenge, previous works mostly adopt the bag-level setting as shown at the top of Figure 1, where the vector representations of sentences are aggregated as the bag-level representation using multi-instance learning (MIL) (Riedel et al., 2010), and the prediction is thus produced from the bag representation. The optimization is conducted at the bag level to minimize the loss of bag prediction. Only a small subset of previous works leverage the sentence-level setting (Zhang et al., 2019b; Liu et al., 2020a) as in the bottom of Figure 1, where the sentence-level predictions are produced and then aggregated into the bag prediction. In fact, sentence-level training can directly optimize the loss from each sentence, enabling better information utilization than bag-level training. However, sentence-level training is vulnerable to the noisy sentences brought by DS, which limits its application. Therefore, sentence-level training should be combined with effective noise-alleviation mechanisms to improve its robustness.

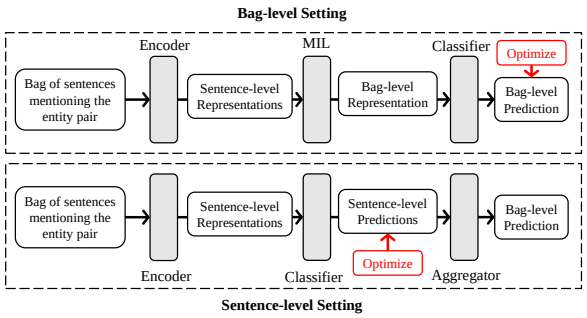

Figure 1: The bag-level and sentence-level pipelines of DSRE.

The encoders of DSRE models remain simple with Piecewise Convolutional Neural Network (PCNN) (Zeng et al., 2015) and Recurrent Neural Network (RNN) (Zhou et al., 2016; Liu et al., 2018) widely used for a long time. Most previous works take the encoder as a black box and only utilize its final output during training and inference. However, when employing BERT (Devlin et al., 2019) as the encoder, the intermediate layers can encode a rich hierarchy of linguistic information of the sentence (Jawahar et al., 2019). Inspired by the idea of probing tasks (Adi et al., 2016; Conneau et al., 2018), in this work we propose to append an auxiliary classifier to each BERT layer and use its output probabilities to probe whether this layer captures the key features indicating the relation between the entity pair. Furthermore, if the relation

indicated by the instance can be well captured by lower layers, i.e, the label relation has high probabilities in their predictions, this instance should be easy for the higher layers. Conversely, if the features captured by lower layers are not sufficient to predict the relation class, it may be a noisy instance with a wrong label, or a hard one requiring higher layers to handle. We can utilize these properties to design an instance weighting mechanism that can adjust the learning priorities of instances and improve the effectiveness of training.

To achieve denoised sentence-level training, we propose a novel **Transitive Instance Weighting (TIW)** mechanism for self-distilled BERT. The self-distilled BERT backbone is employe with one student classifier appended to each probed layer. Each student is trained using distillation and instance weights generated by TIW. The goal of TIW is to tackle noisy instances and alleviate overfitting. There are two types of noisy instances: false negative (*NA* relation) and false positive (non-*NA* relations). We filter false negatives with binary weights (0 or 1) based on the predictions of the previous student (peer). To tackle false positives and prevent overfitting to shallow features, the instance weight is determined by two factors: the uncertainty (Liu et al., 2020b) term and the soft confidence score, which are obtained from the output probabilities of the teacher and the previous peer. The uncertainty term is leveraged to prevent overfitting to easy instances, which mostly contain shallow features. The soft confidence score is used as the assessment of instance difficulty, where easy and hard instances usually have higher scores than noisy ones. During self-distillation, each student receives information and distillation from both the teacher and the previous peer, enabling the alleviation of noise from the teacher and knowledge transfer among students in a transitive way. According to the experiments on both held-out and manual datasets, our approach achieves state-of-the-art performance and consistent improvements over the teacher and the baselines. We also provide detailed ablation study to explore the effects of the modules. Finally, we analyse the errors that occurred and discuss the limitations of our method.

Our contributions are summarized as follows:

- We apply self-distilled BERT to utilize the intermediate outputs and are the first to denoise sentence-level DSRE with instance weights.
- We implement instance weights in a transitive way to enable knowledge transfer among students. The transitive instance weighting alleviates noise and overfitting effectively.
- Experiment and analysis show that our method achieves state-of-the-art performance with good generalization and robustness.

2 Related Work

2.1 Distantly Supervised Relation Extraction

Distant supervision (DS) for relation extraction (Mintz et al., 2009) enables automatic annotation of large-scale datasets, but its strong assumption introduces a large number of wrongly labeled instances. Following Riedel et al. (2010), various multi-instance learning methods are proposed to denoise from noisy instances, and they broadly fall into two categories: instance selection (Zeng et al., 2015; Qin et al., 2018; Feng et al., 2018) and instance attention (Lin et al., 2016; Yuan et al., 2019b,a; Ye and Ling, 2019). Apart from multi-instance learning, many of the previous works try to improve the effectiveness of training. Liu et al. (2017) and Shang et al. (2020) try to convert wrongly labeled instances to useful information through relabeling. Huang and Du (2019) proposes collaborative curriculum learning for denoising. Hao et al. (2021) adopts adversarial training to filter noisy instances in the dataset. Nevertheless, the above approaches are trained with bag-level loss, leading to lower utilization of syntactic and semantic information. We leverage instance weights for denoised sentence-level training to boost performance and robustness.

2.2 Knowledge Distillation

Knowledge distillation (Hinton et al., 2015) is an effective way to improve model generalization, though it has difficulty in transferring knowledge effectively (Stanton et al., 2021). By sharing some parameters between teacher and students, self-distillation (Zhang et al., 2019a) improves knowledge transfer from teacher to students. Liu et al. (2020b) applies self-distillation on BERT (Devlin et al., 2019) to improve inference efficiency. In our work, we employ self-distillation for extracting information within intermediate layers and extend self-distillation with transitive knowledge transfer among the students to further alleviate the noise from the teacher.

3 Methodology

Our model is illustrated in Figure 2. The backbone of our model is the self-distilled BERT on the left, with a teacher classifier on the top. Each student contains a subencoder and an auxiliary classifier. For example, the student 7 has a subencoder ending with the 7th BERT layer and a classifier appended to the 7th layer. The BERT encoder and teacher classifier are fine-tuned on the dataset before distillation. As discussed in Jawahar et al. (2019), the shallow layers may not be able to encode the information needed for the DSRE task. Therefore, TIW starts from layer L, which is empirically set and will be called **the head layer** in the rest of the paper.

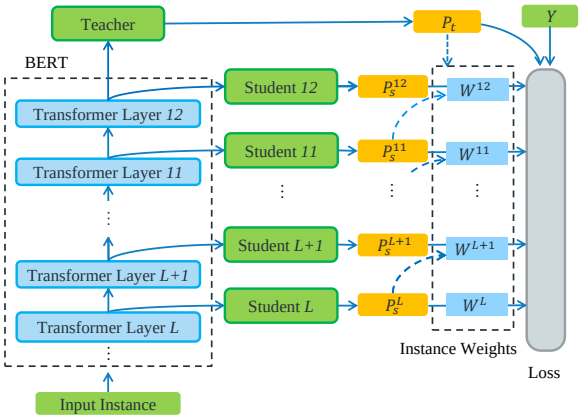

Figure 2: The overall framework of our model. Dotted arrows indicate the generation of instance weight.

3.1 Self-Distilled BERT

BERT (Devlin et al., 2019) is a powerful transformer-based pretrained network with broad applications in natural language processing. However, previous BERT applications in DSRE (Alt et al., 2019; Rao et al., 2022) fail to utilize the information encoded by the intermediate layers, which provides hierarchical views of the sentence features, ranging from surface features, syntactic features, to semantic features (Jawahar et al., 2019). We apply auxiliary classifiers as in Figure 2 to extract the hierarchical features and utilize the output probabilities to denoise from distant supervision. For example, if the student k is able to output a high probability for the label of the instance, it is the indication that the subencoder ending with layer k is able to capture the relation features of the instance. Since the students' subendocers share part of the parameters (knowledge), we can assume that the larger subencoders ending with higher layers also can capture the relation features, meaning that the instance is easy for student k and its higher peers. By sharing some transformer layers and parameters, the self-distillation backbone promotes knowledge sharing from teacher to student and from lower students to higher ones.

Before distillation, we fine-tune the BERT encoder on DSRE as in Gao et al. (2021). The structure of the embedding layer and BERT layers follow those in the previous works with the number of transformer layers $n = 12$ and hidden size $d_h = 768$.

Firstly, the input sentence is transformed to a sequence of vector representations $s = [w_1, w_2, ..., w_m]$ by the embedding layer, where m is the maximum length of the sentence. Then, BERT conducts layer-wise feature extraction with the input s, the output of i_{th} layer ($1 \leq i \leq n$) is described as:

$$h_i = BERT_i(s) \tag{1}$$

where $BERT_i$ refers to the subencoder containing transformer layers from the first to the i_{th}. The teacher model is fine-tuned with a simple feedforward classifier:

$$x_i = [h_i(p_1); h_i(p_2)] \tag{2}$$

$$FFN(h_i) = M_2(M_1 x_i + b_1) + b_2 \tag{3}$$

$$p^t = softmax(FFN_t(h_n)) \tag{4}$$

where $M_1 \in R^{d_h \times d_h}$ and $M_2 \in R^{n_c \times d_h}$ are weight matrices and $b_1 \in R^{d_h}$ and $b_2 \in R^{n_c}$ are bias terms. p_1 and p_2 are the positions of head entity and tail entity respectively. $[a : b]$ indicates the concatenation of vectors a and b. x_i is the entity-aware sentence representation generated by concatenating the hidden vectors of the entity pair. n_c is the number of classes and p^t is the output probability of the teacher.

The student i can be formulated as follows:

$$p_i^s = softmax(FFN_i(h_i)) \tag{5}$$

During distillation, the parameters of the teacher model including the BERT encoder stay fixed and only the student classifiers are updated.

3.2 Transitive Instance Weighting

The algorithm of TIW is shown in Algorithm 1, where $re2id(r)$ is a function that maps the relation

Algorithm 1 Transitive Instance Weighting

Input: DS label Y, teacher's output probability p^t and students' p^s for the instance.
Output: The soft target p^{tg} and the instance weight w of the instance from the students .

1: Initialize $w_l \leftarrow 1, p_l^{tg} \leftarrow p^t$
2: **for** $i = l+1 \rightarrow n$ **do**
3: Compute the soft confidences of i_{th} student: $c_i^t \leftarrow p_i^s \cdot p^t \quad c_i^s \leftarrow p_i^s \cdot p_{i-1}^s$
4: **if** $c_i^t > c_i^s$ **then** $p_i^{tg} \leftarrow p^t$ **else** $p_i^{tg} \leftarrow p_i^s$
5: **if** $Y = rel2id(NA)$ **then** ▷ False Negative Filtering
6: **if** $Y = argmax_j(p_{i-1}^s(j))$ **then** $w_i \leftarrow 1$ **else** $w_i \leftarrow 0$
7: **else** ▷ Positive Weighting
8: Compute the uncertainty of soft target: $u_i \leftarrow \sum_{j=1}^{n_c} \frac{p_i^{tg}(j) log p_i^{tg}(j)}{log \frac{1}{n_c}}$
9: Compute instance weight: $w_i \leftarrow max(c_i^t, c_i^s) u_i$
10: **end if**
11: **end for**

class r to its id for generating the one-hot label. We adopt different weighting strategies for negative instances and positive instances. We conduct false negative filtering (FNF) as in Lines 5-6 of the algorithm. Since we have sufficient negative instances in the dataset, it is acceptable to avoid more false negatives at the cost of slight information loss. Therefore, we assign 0 weight to all the possible false negatives and 1 weight to the true negatives. To correctly identify false negatives, we adopt a dynamic approach that if the previous peer agrees with distant supervision and also labels the instance as *NA*, then we classify the instance as a true negative. Otherwise, we assume it to be a false negative that the DS label is unreliable.

In order to preserve more information for the training of students, we use soft weights for the positive instances. In Positive Weighting (PW), the instance weight w_i of student i is determined by two factors: uncertainty (normalized entropy as in Liu et al. (2020b)) of the chosen soft target and the soft confidence score, which is the maximum between the probabilities of making the same prediction as the teacher c_i^t and the previous peer c_i^s, i.e, the maximum **Probability of Agreement (PoA)** with the two. PoA is computed as the dot product of two probability distributions and can be seen as the consistency between two models.

The uncertainty term is leveraged to prevent overfitting to shallow features in the dataset, especially in easy instances. The instances have low uncertainty values when well-fitted, so we discount their weights with uncertainty terms to prevent overfitting. The idea is that we hope the student tries to learn more hard features instead of shallow ones.

Most previous works in knowledge distillation directly use the teacher's output probability as the soft target. However, the teacher can constantly make mistakes if trained with noisy data, as in DSRE. Therefore, as in Line 4 of our algorithm, instead of blindly following the output from the teacher, each student except the first one chooses between the teacher or the previous peer and follows the one that has higher consistency with itself, i.e, the one that has higher PoA. This provides additional referential probability distributions for the students and helps them in alleviating the noise from the teacher.

The maximum between the PoAs from the teacher and the previous peer is the **Soft Confidence (SC)** score which evaluates the difficulty of the instance for the student. If the SC score is high, the student successfully follows the idea of the teacher or the peer, indicating that the instance is easy to understand for the student.

The instance weight for i_{th} student ($l < i \leq n$) is computed as the product of the SC score and the uncertainty term. Note that during distillation, the student is trained with both soft target distribution and DS labels, as shown in Equation 7. We present the discussions on the SC scores and losses of easy, noisy and hard instances in the following.

Easy instances have high SC scores since they are easy to fit. Easy instances are mostly well-fitted by the teacher or the peer, so the optimizations using soft labels and hard labels conform with each other.

Noisy instances are mostly underfitted and very hard to optimize because the wrong labels contradict the knowledge learned from clean instances.

They also have low SC scores.

Hard instances are underfitted clean instances with medium SC scores. However, they are easier to fit than noisy ones since the soft labels and hard labels also conform with each other.

Based on the above discussions, it is safe to say that both easy and hard instances have larger SC scores and faster optimizations than noisy ones. The uncertainty term only takes effect when easy instances are well-fitted and clean background knowledge is established, so it will not lead to overfitting to noisy instances.

From a global view, each student receives two information flows: the referential probabilities from the teacher and the peer probabilities passed along and updated transitively. Our model alleviates noise from the teacher and distant supervision by dynamically adjusting the learning priorities of the instances. Compared with previous bag-level denoising mechanisms, our method can be combined with sentence-level training, and thus can utilize more information for better performance. Compared with traditional knowledge distillation, our method further alleviates the noise from the teacher. In addition, we only need to add the auxiliary classifiers for distillation and don't need to retrain the model with high cost in time and effort. To sum up, our method is both effective and efficient.

3.3 Optimization

The teacher model may overfit noisy instances during fine-tuning. Therefore, we apply a dynamic temperature τ to the teacher in the following form:

$$\tau_i = 1 + \gamma(1 - u_i) \tag{6}$$

where γ is a hyperparameter empirically set as 3. The idea of τ is to further smooth the well-fitted instances to produce softer targets, thus can alleviate noise and overfitting.

The loss function of our model follows the general form of knowledge distillation with the instance weight w for denoising:

$$L = \sum_{i=l}^{n} w_i(\alpha KL_{\tau_i}(p_i^s, p_i^{tg}) + (1-\alpha)CE(p_i^s, Y)) \tag{7}$$

where α is a hyper-parameter empirically set as 0.5. $KL_\tau(p, q)$ computes the KL-divergence between distributions p and q with temperature τ for the teacher. Y is the label from distant supervision and $CE(p, Y)$ is the cross entropy loss with one-hot label obtained from Y.

4 Experiments

In this section, the datasets, settings and hyperparameters are specified first. Then, we present the performance of our model compared with previous baselines and the teacher model. We also conduct ablation study and error analysis to enable a deeper understanding of the mechanisms.

4.1 Datasets and Settings

We use two datasets for evaluation, the widely used **held-out** dataset NYT-10 (Riedel et al., 2010) and recent **manual** dataset NYT-10m (Gao et al., 2021). As a standard dataset for DSRE, NYT-10 is constructed by aligning the relations in Freebase (Bollacker et al., 2008) with the New York Times corpus (English). The training set includes sentences from 2005 to 2006, and the test set uses sentences from 2007. NYT-10m is a manual dataset constructed also from New York Times corpus, with a human-labeled test set and a new relation ontology. For NYT-10, we divide the dataset into five parts for cross-validation. For NYT-10m, we use the provided validation set. The details of the datasets are shown in Table 1.

Dataset	Train (k)		Test (k)		Rel.
	Sen.	Fac.	Sen.	Fac.	
held-out	522.6	18.4	172.4	2.0	53
manual	417.9	17.1	9.7	3.9	25

Table 1: The statistics of datasets. **Sen.**, **Fac.** and **Rel.** indicate the numbers of sentences, relation facts and relation types (including *NA*) respectively.

In the experiments, we use the *bert-base-uncased* checkpoint with about 110M parameters for initialization as in Han et al. (2019). We apply the AdamW (Loshchilov and Hutter, 2017) optimizer during distillation and fix the random seed as 42. Apart from the hyperparameters previously mentioned, the batch size is 32 and the learning rate is $2e-5$. The maximum length of sentences m is 128. The head layer L is set as layer 7 in our experiments.

During the evaluation, we compare the Area Under precision-recall Curve (AUC), the F1 score and the mean of P@N (N=100, 200, 300), which is denoted as P@M. Following the *at-least-one* assumption (Riedel et al., 2010), we adopt **ONE** strategy (Zeng et al., 2015) for bag-level evaluation, which takes the maximum score for each relation to

generate bag-level predictions. We use the output probabilities of the last student as the output of our model during evaluation. In the appendix, we also display the results from other students and results using different settings of L.

4.2 Overall Performance

We compare the performance of our model against that of the following baselines:

PCNN+ATT (Lin et al., 2016) proposes PCNN with selective attention mechanism.

RESIDE (Vashishth et al., 2018) integrates side information into Graph Convolution Networks to improve relation extraction.

DISTRE (Alt et al., 2019) extends and fine-tunes GPT on DSRE.

Intra+inter (Ye and Ling, 2019) combines intra-bag attention with inter-bag attention to tackle the noisy bags.

CIL (Chen et al., 2021) applies contrastive instance learning to reduce noise from DS.

Teacher follows the implementation in Gao et al. (2021), containing a BERT encoder and a linear classifier.

Among the baselines, DISTRE and CIL use pretrained language models for initialization. CIL adopts the BERT pretrained encoder with the same setting as ours. The held-out dataset is the mainstream for DSRE evaluation, but it contains wrongly-labeled test instances leading to inaccurate evaluation. The manual dataset provides an accurate test set but is limited by its scale in generalization. Therefore, we use both of the datasets for better evaluation.

4.2.1 Evaluation on Held-out Dataset

Model	AUC	F1	P@M
PCNN+ATT	33.8	40.7	71.1
RESIDE	41.5	45.7	79.4
DISTRE	42.2	48.6	66.8
Intra+inter	42.3	46.5	84.8
CIL	<u>50.8</u>	<u>52.2</u>	**86.0**
Teacher	50.6	<u>52.2</u>	83.6
Student 12	**53.9**	**55.3**	<u>84.9</u>

Table 2: The performance (%) of our model and the baselines on the held-out dataset. The best scores are marked as **bold** and the second best scores are <u>underlined</u>.

Table 2 shows the experimental results on the held-out dataset. We use the results reported in the papers of previous work. We also plot the precision-recall curves as in Figure 3.

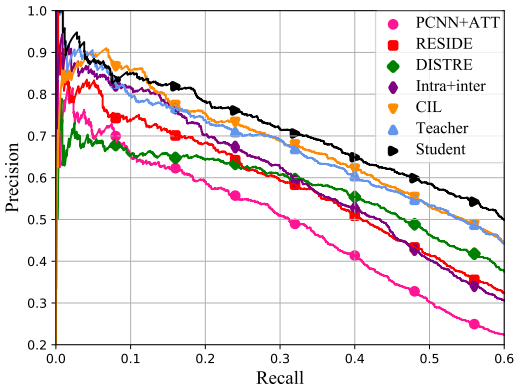

Figure 3: PR curves of the models on the held-out dataset.

As shown in the results, our model achieves the best AUC and F1 score among all the compared methods. The P@M of our model is relatively lower than bag-level methods, but still significantly higher than the teacher model. We can see that sentence-level training leads to a slight decline in the precision of top predictions due to the existence of noisy sentences but achieves better overall performance on the test set because of its advantage in information utilization. Our method further alleviates noise and overfitting, achieving state-of-the-art performance by only retraining the classifier with self-distillation and instance weights.

4.2.2 Evaluation on Manual Dataset

Model	AUC	F1	P@M
PCNN+ATT	57.7	57.0	89.2
Intra+inter	53.6	53.5	**91.8**
CIL	60.2	58.8	<u>91.7</u>
Teacher	<u>61.3</u>	<u>62.4</u>	84.3
Student 12	**63.9**	**63.8**	90.8

Table 3: The performance (%) of our model and the baselines on the manual dataset. The best scores are marked as **bold** and the second bests are <u>underlined</u>.

Table 3 shows the experimental results on the manual dataset. We use the original implementations of the methods to train and evaluate using the manual dataset. The precision-recall curves are plotted in Figure 4.

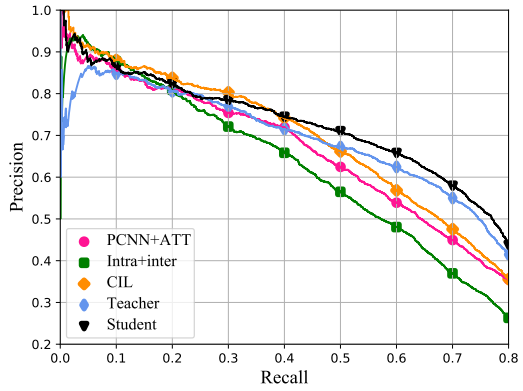

Figure 4: PR curves of the models on manual dataset.

In the results, the bag-level methods still perform better on top predictions, however, our method outperforms them in AUC and F1 score by large margins. It shows that previous bag-level methods overfit easy instances during training, leading to the loss of overall generalization. Compared with the teacher model, our method also achieves significant improvements, especially in P@M. It shows that our method effectively alleviates the DS noise in sentence-level training.

According to Gao et al. (2021), the performance of the model may be inconsistent if evaluated in both the held-out and manual datasets. Good performance on the held-out set may indicate overfitting to the bias from distant supervision. However, our method achieves state-of-the-art performance in both datasets, which further demonstrates its effectiveness and robustness.

4.3 Ablation Study

Model	AUC	F1	P@M
Our method	**53.9**	**55.3**	84.9
a: - Un	52.5	53.2	**86.1**
b: - PW	51.9	52.5	84.8
c: - FNF	53.3	54.9	82.5
d: - TIW	52.1	52.6	84.6
e: Probe	50.6	52.5	80.0

Table 4: Ablation study of our method. The best results are marked as **bold** and the second best are underlined.

As shown in Table 4, all the modules improve the overall performance of our model. Detailed discussions are given below:

a: removes uncertainty term and directly uses SC score as positive weights. In this case, the easy instances always have the largest weights during distillation even though they are already well-fitted. The model thus overfits shallow features, which is verified by the high precision on top predictions and the decline of overall performance.

b: removes PW and all the positive instances are treated equally, including the noisy ones. In this case, the model is heavily affected by noise and the false negative filtering may be inaccurate, leading to further declines in performance.

c: removes FNF. The false negative instances only make up a small part of the dataset, so the effect of removing FNF is relatively small. However, the noise from false negatives significantly reduces P@M. We suspect that the fitting of false negatives affects that of true positives. If a false negative fn has similar syntactic and semantic features to a true positive tp, fitting fn is similar to fitting tp using an incorrect label.

d: removes TIW totally and all the instances are weighted as 1. The label smoothness of knowledge distillation is able to alleviate the noise from distant supervision, so there is improvement in performance over *e*. However, the teacher model is trained with DS label and overfits noisy instances. Without TIW, the noise from the teacher cannot be effectively tackled.

e: is the probing result of 12th layer using the DS label. It shows that without effective denoising mechanisms, simply retraining the classifier does not help in performance.

As shown in the ablation results and analysis, all the modules play important roles in denoising from distant supervision and combining them together leads to the best performance.

4.4 Error Analysis

For accurate analysis of the errors of the model, we use the test set of the manual dataset for statistical discussions. Each positive label is considered an **item**. The instances with multiple positive labels are considered to have multiple items. We classify the items based on the predictions of the teacher and student, then count the number and percentage of each class as in Table 5. The goal is to explore where the errors of the student come from: a) **from the teacher**, meaning that the knowledge from the teacher is noisy and leads to the student's errors, or b) **from the student itself**, meaning that the teacher gives correct knowledge but the student fails to follow.

Sentence	Teacher	Student
Carl Friedrich von Weizsäcker was born in *Kiel*, Germany, on June 28, 1912.	/people/person/place_of_birth	/people/person/place_lived
Presented by *Brooklyn College* and the office of Borough President *Marty Markowitz*.	/business/person/company	/people/person/place_lived
Furthermore, the relationship between the central government, dominated by three small *A rab* tribes living along the Nile, and Darfur's Arabs, who claim a heritage going back to th e Prophet *Muhammad*, is often antagonistic.	/people/person/ethnicity	/people/person/place_of_birth

Figure 5: *TCSI* examples. The entities are underlined.

Class	Num. of items	Percentage (%)
BC	3,044	78.07
BI	742	19.03
TISC	94	2.41
TCSI	19	0.49

Table 5: Numbers and percentages of different classes of items. *BC* stands for *both correct*, *BI* stands for *both incorrect*, *TISC* stands for *teacher incorrect, student correct* and *TCSI* stands for *teacher correct, student incorrect*.

In the results, the student achieves slightly higher (about 2%) accuracy than the teacher and shows high fidelity with 97.1% of all predictions being the same as the teacher. *BI* represents the student's errors caused by the errors from the teacher. *TISC* indicates the student's corrections on the errors from the teacher and *TCSI* represents the errors from the student itself. From the results, we can conclude that almost all (about 97.5%) of the errors come from the teacher, and the corrections made by the student are much more than the errors made by the student itself. This demonstrates the effectiveness of our method in reducing the occurrence of errors and the limitation that it requires a good teacher for good performance.

For further analysis of the student's errors, we inspect the *TCSI* items and select some representative ones for discussions as in Figure 5. Most of the instances with *place_of_birth* relation are correctly classified and the first example should be an easy instance in the form, yet misclassified by the student as *place_lived*. We observe several similar items and suspect that long and uncommon names like *Carl Friedrich von Weizsäcker* sometimes confuse the student to make conservative predictions, which is the more common relation *place_lived*. The second example, however, confuses the student with a compound noun *Brooklyn College*. *Brooklyn* appears very often in the dataset in the form of location, making the student believe that *Brooklyn College* is a location rather than an organization. The third example is mostly related to ambiguity, where the word *Arab* may refer to the Arab people (ethnic group) or the Arab world (location). The latter two examples indicate that the lack of entity-related information may lead to inconsistency between the student and the teacher. The first example shows that the student may be confused to lose focus on key phrases like *was born in*, which may be solved by combining with word-level attention in the future.

5 Conclusions and Limitations

In this paper, we propose a novel transitive instance weighting mechanism integrated with a self-distilled BERT structure to denoise from sentence-level training of DSRE. We employ the self-distilled backbone to utilize more information and achieve better efficiency. We use the instance weights generated through careful utilization of the knowledge from the teacher and the peers to tackle the challenges of noisy instances, overfitting to shallow features and noise from the teacher. The experiment results show that our method improves the general resistance to DS noise and prevents overfitting from harming its generalization, thus can achieve state-of-the-art performance and consistent improvements over the baselines on both the held-out and manual datasets.

However, our work still has some limitations. Firstly, Since our model is built on the basis of the teacher-student network, the performance of the student is highly affected by the teacher. If the teacher provides too much noisy information, our instance weighting mechanism might not work. Secondly, in some cases, the student fails to follow the correct predictions from the teacher due to ambiguity, lack of information or word-level noise, which indicates that further extension of our method is plausible. Finally, we haven't explored other instance weighting methods in this paper. There might be better solutions yet to be discovered.

References

Yossi Adi, Einat Kermany, Yonatan Belinkov, Ofer Lavi, and Yoav Goldberg. 2016. Fine-grained analysis of sentence embeddings using auxiliary prediction tasks. *arXiv preprint arXiv:1608.04207*.

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Fine-tuning pre-trained transformer language models to distantly supervised relation extraction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1388–1398, Florence, Italy. Association for Computational Linguistics.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Tao Chen, Haizhou Shi, Siliang Tang, Zhigang Chen, Fei Wu, and Yueting Zhuang. 2021. CIL: Contrastive instance learning framework for distantly supervised relation extraction. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6191–6200, Online. Association for Computational Linguistics.

Alexis Conneau, German Kruszewski, Guillaume Lample, Loïc Barrault, and Marco Baroni. 2018. What you can cram into a single $&!#* vector: Probing sentence embeddings for linguistic properties. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2126–2136, Melbourne, Australia. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. 2018. Reinforcement learning for relation classification from noisy data. In *Proceedings of the aaai conference on artificial intelligence*, volume 32.

Tianyu Gao, Xu Han, Yuzhuo Bai, Keyue Qiu, Zhiyu Xie, Yankai Lin, Zhiyuan Liu, Peng Li, Maosong Sun, and Jie Zhou. 2021. Manual evaluation matters: Reviewing test protocols of distantly supervised relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1306–1318, Online. Association for Computational Linguistics.

Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. 2019. OpenNRE: An open and extensible toolkit for neural relation extraction. 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): System Demonstrations*, pages 169–174, Hong Kong, China. Association for Computational Linguistics.

Kailong Hao, Botao Yu, and Wei Hu. 2021. Knowing false negatives: An adversarial training method for distantly supervised relation extraction. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9661–9672, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7).

Yuyun Huang and Jinhua Du. 2019. Self-attention enhanced CNNs and collaborative curriculum learning for distantly supervised relation extraction. 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 389–398, Hong Kong, China. Association for Computational Linguistics.

Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. 2019. What does bert learn about the structure of language? In *ACL 2019-57th Annual Meeting of the Association for Computational Linguistics*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133.

Tianyi Liu, Xiangyu Lin, Weijia Jia, Mingliang Zhou, and Wei Zhao. 2020a. Regularized attentive capsule network for overlapped relation extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6388–6398, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Tianyi Liu, Xinsong Zhang, Wanhao Zhou, and Weijia Jia. 2018. Neural relation extraction via inner-sentence noise reduction and transfer learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2195–2204.

Tianyu Liu, Kexiang Wang, Baobao Chang, and Zhifang Sui. 2017. A soft-label method for noise-tolerant distantly supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1790–1795.

Weijie Liu, Peng Zhou, Zhiruo Wang, Zhe Zhao, Haotang Deng, and Qi Ju. 2020b. FastBERT: a self-distilling BERT with adaptive inference time. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6035–6044, Online. Association for Computational Linguistics.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 2009. 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*, pages 1003–1011.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2137–2147.

Ziqin Rao, Fangxiang Feng, Ruifan Li, and Xiaojie Wang. 2022. A simple model for distantly supervised relation extraction. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 2651–2657, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer.

Yuming Shang, He-Yan Huang, Xian-Ling Mao, Xin Sun, and Wei Wei. 2020. Are noisy sentences useless for distant supervised relation extraction? In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8799–8806.

Samuel Stanton, Pavel Izmailov, Polina Kirichenko, Alexander A Alemi, and Andrew G Wilson. 2021. Does knowledge distillation really work? *Advances in Neural Information Processing Systems*, 34:6906–6919.

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. Reside: Improving distantly-supervised neural relation extraction using side information. *arXiv preprint arXiv:1812.04361*.

Zhi-Xiu Ye and Zhen-Hua Ling. 2019. Distant supervision relation extraction with intra-bag and inter-bag attentions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2810–2819.

Changsen Yuan, Heyan Huang, Chong Feng, Xiao Liu, and Xiaochi Wei. 2019a. Distant supervision for relation extraction with linear attenuation simulation and non-iid relevance embedding. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7418–7425.

Yujin Yuan, Liyuan Liu, Siliang Tang, Zhongfei Zhang, Yueting Zhuang, Shiliang Pu, Fei Wu, and Xiang Ren. 2019b. Cross-relation cross-bag attention for distantly-supervised relation extraction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 419–426.

Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019a. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3713–3722.

Xinsong Zhang, Pengshuai Li, Weijia Jia, and Hai Zhao. 2019b. Multi-labeled relation extraction with attentive capsule network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7484–7491.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. 2016. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 207–212.

A Hyperparameter Analysis

There are two key hyperparameters in our experiments, the student selected and the head layer L. In our best model, we select the last student (12th) for evaluation and set layer 7 as the head layer.

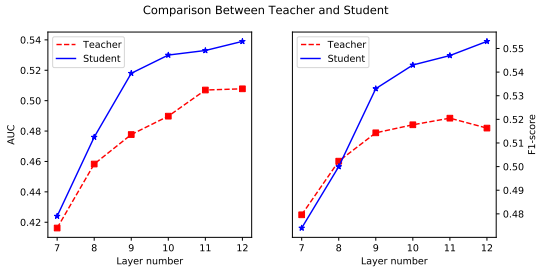

Figure 6: PR curves of the students and auxiliary classifiers of the teacher on the held-out dataset.

As shown in Figure 6, the higher students($\geq$ 9) improve significantly over the teacher. The last student performs the best and the students from 9th to

11th also achieve comparable performances. Lower layers of BERT encode shallower features and the instance weighting in lower students is more affected by noise, so the performances of 7th and 8th students show little advantage over the teacher. With knowledge passed and noise alleviated student by student, the performance gradually improves.

Setting	AUC	F1	P@M
$L = 11$	53.4	55.1	82.8
$L = 10$	53.5	54.9	83.6
$L = 9$	53.6	55.0	84.0
$L = 8$	53.7	55.1	84.7
$L = 7$	**53.9**	**55.3**	**84.9**
$L = 6$	53.8	**55.3**	84.8
$L = 5$	53.7	55.1	84.6
$L = 3$	53.5	55.0	84.7
$L = 2$	53.5	54.9	84.6
$L = 1$	53.4	54.9	84.5

Table 6: Results of using different head layer L settings. The best results are marked as **bold**.

To study the effect of head layer L, we run experiments with L from 1 to n. In Table 6, we present the results where $L = 7$ achieves the best performance. For $L > 7$, the head layer is too close to the top, and TIW filters fewer false negatives. So the P@M declines quickly, which is similar to the effect of removing FNF as in Table 4. For $L < 7$, the lower layers of BERT are not able to encode sufficient information for accurate relation extraction, so the lower students are not able to provide reliable instance weights, leading to the transfer of some noise among students. Though other settings are less effective than the best, their performances still dominate the baselines. The above results show that our method is not dependent on the empirical settings of hyperparameters and further demonstrate the effectiveness and robustness of our method.