H-FND: Hierarchical False-Negative Denoising for Distant Supervision Relation Extraction

Jhih-Wei Chen[*1], **Tsu-Jui Fu**[*2], **Chen-Kang Lee**[3], and **Wei-Yun Ma**[3]

[1]University of California Los Angeles
[2]University of California Santa Barbara
[3]Institute of Information Science, Academia Sinica
jwchen101@g.ucla.com, tsu-juifu@ucsb.edu
{cklee, ma}@iis.sinica.edu.tw

Abstract

Although distant supervision automatically generates training data for relation extraction, it also introduces false-positive (FP) and false-negative (FN) training instances to the generated datasets. While both types of errors degrade the final model performance, previous work on distant supervision denoising focuses more on suppressing FP noise and less on resolving the FN problem. We here propose H-FND, a hierarchical false-negative denoising framework for robust distant supervision relation extraction, as an FN denoising solution. H-FND uses a hierarchical policy which first determines whether non-relation (NA) instances should be kept, discarded, or revised during the training process. For those learning instances which are to be revised, the policy further reassigns them appropriate relations, making them better training inputs. Experiments on SemEval-2010 and TACRED were conducted with controlled FN ratios that randomly turn the relations of training and validation instances into negatives to generate FN instances. In this setting, H-FND can revise FN instances correctly and maintains high F1 scores even when 50% of the instances have been turned into negatives. Experiments on NYT10 is further conducted to show that H-FND is applicable in a realistic setting. [1]

1 Introduction

Relation extraction (Zelenko et al., 2003; Mooney and Bunescu, 2006; Zhou et al., 2005) is a core task in information extraction. Its goal is to determine the relation between two entities in a given sentence. For instance, given the sentence "Jobs was born in San Francisco", with head and tail entities "Jobs" and "San Francisco", the relation to be extracted is "Place of Birth". Relation extraction can be applied for many applications, such as question answering and knowledge graph completion. A major difficulty with supervising relation extraction models is the cost of collecting training data, against which distant supervision (DS) (Hoffmann et al., 2011; Surdeanu et al., 2012) is proposed. DS obtains the relational facts from a knowledge base and aligns these facts to all sentences in the corpus to generate learning instances. In specific, if a relation triple $r(h, t)$ exists in a knowledge base, then for a sentence s which mentions both the head entity h and the tail entity t, it is tagged with relation r to form a learning instance (r, h, t, s). Since an effective classifier is expected not only to extract relation triples from a given text but also have to identify those unrelated entity pairs, the negative samples from texts are also needed for the training. In distant supervision, the negative samples are generated by randomly selecting two entities in the given text to form an entity pair that does not appear in any relation triples in the knowledge base.

Knowledge base	Relation	
Steve Jobs, **San Francisco**	PoB	
Corpus	Relation	Type
Jobs was born in **San Francisco**	PoB (✓)	TP
Jobs moved back to **San Francisco**	PoB (✗)	FP
Manuela was born in **New York**	NA (✗)	FN

Table 1: Distant supervision and different types of incorrectly labeled relations. The head and tail entities are shown in boldface, and "PoB" stands for the relation "Place of Birth".

Datasets generated using distant supervision contain considerable noise (Roth et al., 2013). More specifically, the noise generated can be classified into false positives (FP) and false negatives (FN). Table 1 shows an example. The FP "Jobs moved back to San Francisco" should not reflect the relation 'Place of Birth'. Also, an FN: as there is no relation between "Manuela" and "New York" in the knowledge base, "Manuela was born in New

[*]Equal contribution.
[1]The code can be found at https://github.com/ckiplab/hfnd

York" is wrongly labeled as a non-relation (NA) under the closed world assumption. Both FP and FN degrade model performance if they are treated as correct labels at training time. FPs harm prediction precision, while excessive FNs lead to low recall rates.

In addition to denoising methods for learning robustly with noisy data (Han et al., 2018; Northcutt et al., 2019), many works focus on alleviating the FP problem in DS datasets, including those on pattern-based extraction (Alfonseca et al., 2012; Jia et al., 2019), multiple-instance learning (Surdeanu et al., 2012; Lin et al., 2016; Zeng et al., 2018), and sentence-level denoising with adversarial training or reinforcement learning (Qin et al., 2018a,b; Feng et al., 2018). However, few investigate the FN problem for distant supervision (Xu et al., 2013; Roller et al., 2015). To the best of our knowledge, there is no previous study on this problem for deep neural networks.

In this paper, we investigate the impact of FNs on neural-based models and propose H-FND, a hierarchical false-negative denoising framework for robust distant supervision. Specifically, this framework integrates a deep reinforcement learning agent which keeps, discards, or revises probable FN instances with a relation classifier to generate revised relations. In addition, to constrain the study to the FN problem and to construct ground-truth relations to further analyze model behavior, we conduct our research on the following two human-annotated datasets: SemEval-2010 (Hendrickx et al., 2010) and TACRED (Zhang et al., 2017), with controlled FN ratios that randomly flip relations of training/validation instances into negatives to generate FN instances. Then, we further conduct our experiment on a distantly supervised dataset NYT10 (Riedel et al., 2010) and fix its positive set, to demonstrate that our framework is applicable for resolving FN problem in a realistic setting. In summary, our contributions are threefold:

- We propose a denoising framework focused on false negatives in relation extraction.
- We present a special transfer learning scheme for pretraining denoising agent as training data is not available for this pretraining task.
- We show that our method revises correctly and maintains high F1 scores even under a high percentage of false negatives, and is applicable in a realistic setting.

2 Related Work

Mintz et al. (2009) propose distant supervision (DS) to automatically generate labeled data for relation classification, a new paradigm that synthesizes positive training data by aligning a knowledge base to an unlabeled corpus, and produces negatives with a closed-world assumption. Although this method requires no human effort for sentence labeling, it introduces FPs and FNs into the generated data and degrades the performance of relation extraction models.

Many previous works have attempted to solve the FP problem. Among these works, denoising methods that utilize reinforcement learning (RL) are the most relevant to ours. Feng et al. (2018) propose a sentence-level denoising mechanism that trains a positive instance selector using RL, and set the RL reward to the prediction probability of the relation classifier. Qin et al. (2018b) also utilizes RL, but in a different way. It learns a denoising agent to redistribute FPs to NA via prediction accuracy of the classifier as the RL reward.

To solve the FN problem, one method is to align the KB to the corpus after performing KB completion using inference (Roller et al., 2015). Although this does reduce the number of FNs in DS datasets, it helps little when the FN relations cannot be inferred from the KB, e.g., the entities mentioned in the FN are not in the KB. IRMIE (Xu et al., 2013), another method, constructs a negative set in a more conservative sense, in which the head or tail entities have already participated in other relation triples in the KB. Other sentences outside the positive and negative sets are left unlabeled (labeled as RAW in original paper) to prevent FNs. After training on the positive and negative sets, positive relation triples are retrieved from the unlabeled set to expand the KB, after which the original DS is performed to improve the quality of relation extraction. The final performance of this method depends heavily on the heuristic for constructing the negative set, which may not be applicable for all possible relation types.

To address the FN problem in DS datasets more generally, we propose a hierarchical denoising method to mitigate the negative effect of FNs, ensuring a more robust relation extraction model when the presence of FN instances is unavoidable.

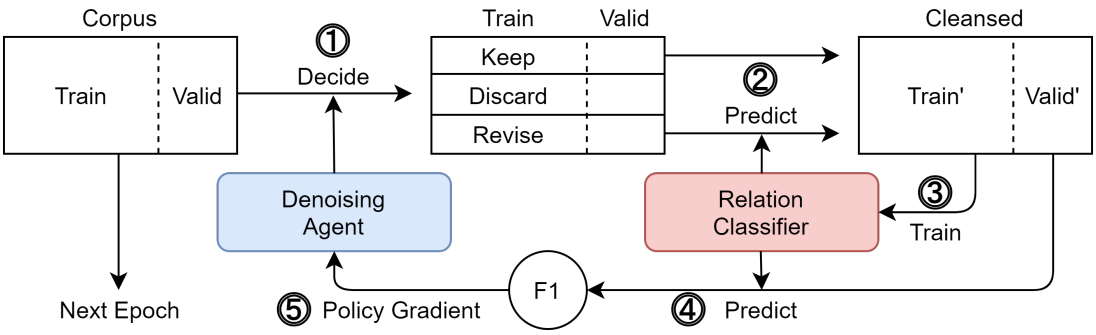

Figure 1: H-FND framework. The process in this diagram is executed per epoch.

3 H-FND Framework

We propose H-FND, a hierarchical false-negative denoising framework that determines whether to keep, discard, or revise negative instances. As illustrated in Fig. 1, H-FND is composed of the denoising agent and relation classifier modules. The denoising agent makes a ternary decision on the action to take on each negative instance, and after discarding, the relation classifier predicts a new relation for each to-be-revised instance to produce a cleaned dataset.

3.1 Convolutional Neural Network

Convolutional neural networks (CNN) are commonly adopted for sentence-level feature extraction (Kim, 2014) in language understanding tasks, such as relation extraction (Zeng et al., 2014; Nguyen and Grishman, 2015). PCNNs (Zeng et al., 2015), a variation of CNN that applies piecewise max pooling, are also widely used for extracting sentence features (Lin et al., 2016; Qin et al., 2018a). We included both as the base model in our experiments to show that our framework is base model agnostic. In our implementation, the extracted features of a learning instance s are fed into a fully connected softmax classifier to compute the final logits:

$$O(r) = \text{softmax}(\text{FC}(\text{CNN}(s))).$$

For detailed mathematical descriptions of CNN and PCNN, please refer to the Appendix.

3.2 Hierarchical Denoising Policy

The proposed hierarchical denoising policy is a framework using policy-based reinforcement learning (RL). Previous work utilizing RL to suppress noise from FPs (Feng et al., 2018; Qin et al., 2018b) can be categorized in two types of strategies: the first decides whether to remove the input instance, and the second decides whether to revise the input instance to be negative. Both policies make a binary decision on each input instance, and successfully reduce FP instances in DS datasets.

While applicable on the FP problem, it is risky to directly apply these strategies on the FN problem. First, discarding a negative instance even when it is most likely positive can result in a loss of useful learning instances. Second, changing a negative instance to positive is not enough for the training process: we must also know which type of positive relation to revise to.

Therefore, we propose a hierarchical denoising policy to perform the FN denoising in two steps. The first step, a soft policy that combines the two above-mentioned denoising methods, is an agent that takes an action from the action set {*Keep, Discard, Revise*} for a negative instance s:

- *Keep:* maintain s as a negative instance for training/validation;
- *Discard:* remove s to prevent it from misleading the model;
- *Revise:* predict a new relation type for s and treat it as a positive for the following training/validation.

The policy $\pi(a|s)$ of this ternary decision is calculated based on the sentence feature extracted from s with the base model CNN encoder:

$$\pi(a|s) = \text{softmax}(\text{FC}_1(\text{CNN}_1(s)));$$

each action a has the possibility of $\pi(a|s)$ of being taken by the denoising agent.

Then, if the negative instance s is to be revised, the hierarchical policy goes on to the second step and gives the revised relation by selecting the most

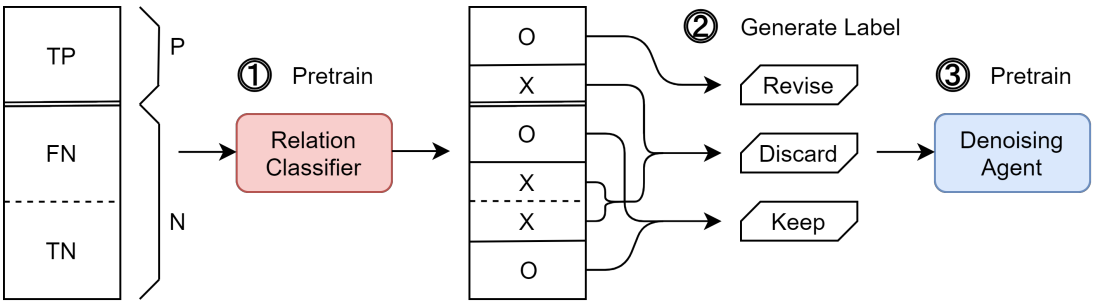

Figure 2: A special transfer learning scheme for H-FND pretraining. Symbols "P" and "N" represent positive and negative instances for relation classifier pretraining. Symbols "O" and "X" indicate two sets of training instances which are correctly predicted and wrongly predicted by pretrained relation classifier correspondingly.

likely relation (excluding NA) predicted by the relation classifier:

$$r' = \arg \max_{r \in R \setminus \{NA\}} \text{FC}_2(\text{CNN}_2(s)).$$

3.3 Pretraining

Supervised pretraining (Qin et al., 2018b), commonly used to accelerate RL agent training, is easily performed for the relation classifier on the original DS dataset (Han et al., 2018). For the denoising agent, however, there is no available training data. Therefore, we propose a special transfer learning scheme that utilizes the learnt knowledge in the relation classifier (source domain) to help generate action labels for pretraining denoising agent (target domain) (See Fig. 2).

First, we select the positives for which the pretrained relation classifier correctly predicts the relation, and tag these with *Revise*. This prepares the denoising agent to identify positive instances in the negative set in future training, and then pass these kinds of instances to the relation classifier to predict the correct positive relations for them. Similarly, we tag with *Keep* those negatives correctly predicted by the relation classifier. Lastly, for instances in which the relation classifier wrongly predicts their relation, we tag them with *Discard*, encouraging the denoising agent to discard such instances to avoid incorrect revisions.

In summary, our pretraining strategy is thus:

1. **Relation classifier pretraining**: pretrain the relation classifier (RC) directly on the original training set with the categorical loss function:

   $$ls_{\text{RC}} = \text{cross-entropy}(O, G),$$

   where G represents the distantly supervised relation in the training set. Then, fix the parameters of the relation classifier for the next step.

2. **Label generation**: generate labels H with the predictions of the relation classifier.

3. **Denoising agent pretraining**: Supervise the denoising agent (DA) with categorical loss:

   $$ls_{\text{DA}} = \text{cross-entropy}(\pi, H).$$

3.4 Co-Training

To combine the training of the relation classifier and the denoising agent, we propose the following co-training framework during each epoch (see Fig. 1):

1. **Denoising agent decision**: At the beginning of each epoch, the denoising agent first executes the denoising policy on the dataset. For both training and validation sets, the policy keeps, discards, or revises NA instances.

2. **Relation classifier revision**: For instances to be revised, the relation classifier generates revision relations for them. Denoising yields the cleaned training and validation sets.

3. **Relation classifier training**: Given the cleaned training set, we train the relation classifier in a supervised fashion based on categorical loss:

   $$ls_{\text{RC}} = \text{cross-entropy}(O, G'),$$

   where G' represents the modified training set, which contains all the positives and the kept or revised negatives. Note that discarded negatives are not included in G'.

4. **Reward determination**: We evaluate the trained relation classifier on the cleaned validation set to obtain the F1 score, which we use as reward R for denoising. As the validation set is cleaned by the denoising policy, R reflects the efficacy of the policy.

Datasets	#training	#validation	#testing
SemEval	6,599	1,154	2,717
TACRED	63,782	20,088	15,509
NYT10	477,454	120,318	194,328

Table 2: Number of instances in each dataset

5. **Denoising policy update**: To maximize the reward R, we adopt policy gradient (Sutton et al., 2000) to optimize the denoising agent by maximizing the objective function $J(\theta)$:

$$J(\theta) \approx \sum \log p(a|\theta)(R - b),$$

where θ is the parameter of the denoising policy, $p(a|\theta)$ represents the softmax probability of the sampled determination or revision step, and b is the baseline which mitigates the high variance of the REINFORCE algorithm (Williams, 1992). We set b to the average reward of the previous five epochs.

For each epoch, we obtain the revised set from the original training/validation set via the denoising policy, and H-FND finds the best denoising policy adaptively between supervised training and reward maximization.

4 Experiment

4.1 Datasets

In order to quantify our model's performance on denoising false negatives, we evaluated the proposed H-FND under two settings, human-annotated datasets with synthetic noise and dataset generated using distant supervision. Table 2 shows the statistics of each dataset used in the experiments.

1. **Human-Annotated Datasets**:

   SemEval-2010[2] contains nine relations with an additional NA as a non-relation, and the number of instances for each relation is roughly equal. TACRED[3] is about 10 times larger than SemEval, and it has 42 relations including NA, and the number of negative instances accounts for 80% of the entire corpus. For SemEval, we used 10% of the training set for validation, and for TACRED we simply used the dev set as the validation set (see Table 2).

   We filtered out the training and validation instances which had relation triples that appeared in the testing set to eliminate any overlap between relation triples in the training, validation, and testing sets, to simulate the held-out evaluation settings in distant supervision (Mintz et al., 2009).

   To simulate FN conditions, we randomly filtered a ratio (10%–50%) of training/validation positives into negatives. Note that the filtering process was only for training/validation: the testing sets were well-labeled under all FN ratios. Also note that the models were not aware in advance which sentences were TN and which were FN.

2. **Distantly Supervised Dataset**: The NYT10 dataset[4] uses Freebase as knowledge base for distant supervision. The relations are extracted from a December 2009 snapshot of Freebase. Four categories of Freebase relations are used: "people", "business", "person", and "location". These types of relations are chosen because they appear frequently in the newswire corpus. All pairs of Freebase entities that are at least once mentioned in the same sentence are chosen as candidate relation instances. For consistency with previous research (Lin et al., 2016; Feng et al., 2018; Qin et al., 2018b), we excluded five relations: *'/business/company/industry'*, *'/business/company_shareholder/major_shareholder_of'*, *'/people/ethnicity/includes_groups'*, *'/people/ethnicity/people'*, *'/sports/sports_team_location/teams'*

   This results in a total of 53 relations (including none-relation, 'NA').

   The corpus is chosen from a external source articles published by The New York Times between January 1, 1987 and June 19, 2007. The Freebase relations were divided into two parts, one for training and one for testing. The former is aligned to the years 2005-2006 of the NYT corpus, the latter to the year 2007.

4.2 Baselines and Experiment Settings

A simple H-FND baseline was the original CNN and PCNN relation classifier. To demonstrate the

[2] http://www.kozareva.com/downloads.html
[3] https://catalog.ldc.upenn.edu/LDC2018T24
[4] http://iesl.cs.umass.edu/riedel/ecml/

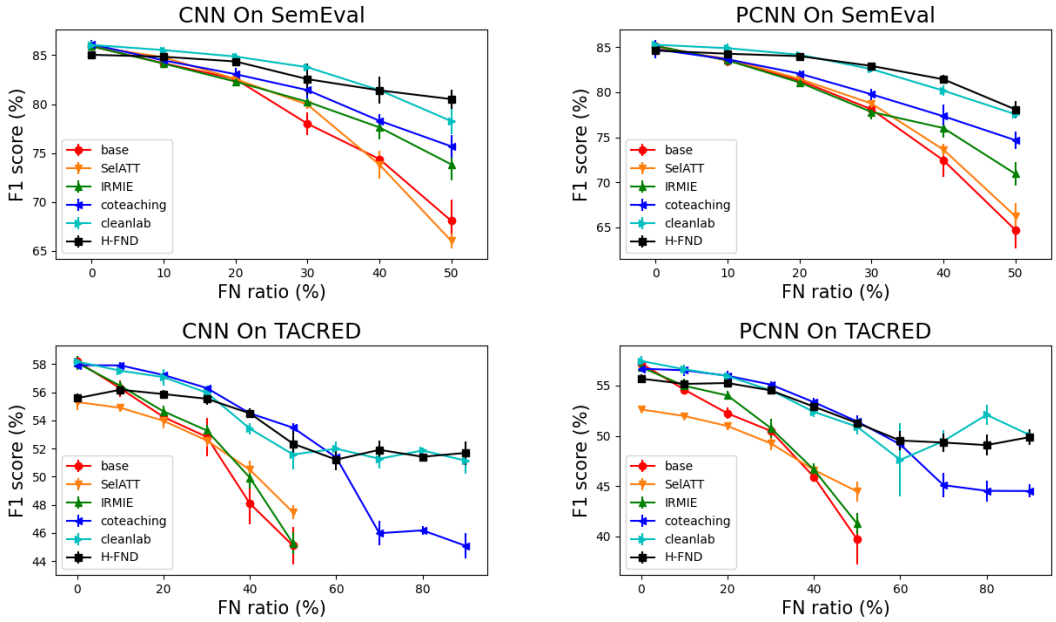

Figure 3: CNN and PCNN results on SemEval and TACRED, where the errorbars represent the standard deviations. The noise rate denotes the percent of positive relation triples flipped to create false negatives. The denoising method cleanlab and our method H-FND perform the best, but cleanlab requires a given noise rate of data, while H-FND does not requires such information.

impact of FNs, we also included SelATT (Lin et al., 2016), an FP noise resistant model.

We further compared our H-FND framework with the following strong baselines: the FN denoising method IRMIE (Xu et al., 2013) and two other general-purpose denoising methods: co-teaching (Han et al., 2018) and cleanlab (Northcutt et al., 2019). Co-teaching is a general training method for deep neural networks to combat extremely noisy labels. It simultaneously maintains two networks (each with the same structure), each of which samples its small-loss instances with a given overall noise rate as clean batches to its peer networks for further training. Cleanlab is a state-of-the-art robust learning method which directly estimates the joint distribution of noisy observed labels and latent uncorrupted labels with a consistent estimator, filters out noisy instances based on this joint distribution, and trains the relation classifier on the cleaned dataset with co-teaching mentioned above. We use these denoising methods to train the base CNN and PCNN models on our simulated FN datasets. [5]

As the focus of this paper is on the FN problem, all the positives of the simulated FN datasets are kept error-free, the H-FND framework assumes that no positives need be changed. Hence, for a fair comparison, we kept the positive sets of the FN datasets unchanged for the two general-purpose denoising methods, preventing them from discarding error-free positives. Also, we fix the positive set of NYT10 to evaluate the applicability of H-FND of resolving FN problem in a realistic setting.

In the experiments on SemEval and Tacred, every data point is the average of five independent runs. In the experiment of NYT10, data points are the average of three best results out of five independent runs for H-FND and all baselines. See Appendix for more detailed information on experiment and model implementation.

4.3 Quantitative Results

The quantitative SemEval results are shown in the upper part of Fig. 3, including both CNN and PCNN. Under the 50% FN ratio, for both the base CNN and PCNN models, with or without SelATT, the F1 scores are heavily influenced by FN sentences: the performance drops by nearly 20%. ERMIE and co-teaching enhance the performance by more than 5% and 8% correspondingly. Except for cleanlab, H-FND denoising remains competitive to the baselines for FN ratios from 10% to 30%, and significantly wins after 30%. Among all

[5] The IRMIE KB was reconstructed from the positives of the simulated FN dataset.

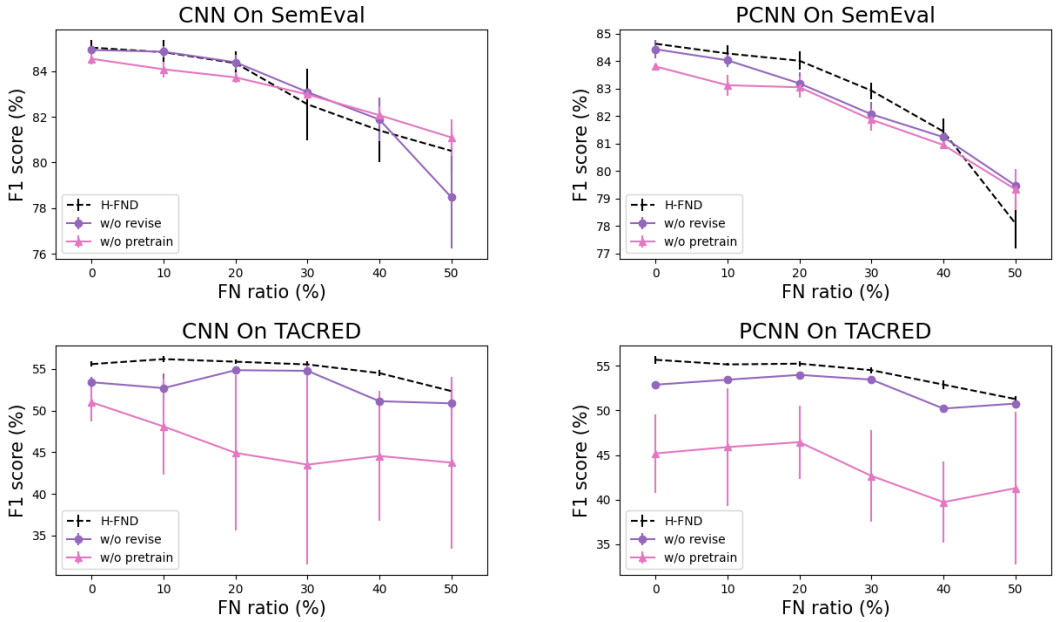

Figure 4: CNN and PCNN ablation analysis on SemEval and TACRED, where the errorbars represent the standard deviations.

baselines, cleanlab's performance is the strongest and is competitive with our approach, but as cleanlab relies on a co-teaching model to train the relation classifier, a given noise rate is required. In our experiments, these are directly provided to the model. However, in practice, the noise rate (the FN ratios in our experiment) is unknown and must be estimated correctly, entailing extra effort. In contrast, H-FND has no such requirement.

The quantitative results on TACRED are shown in the lower part of Fig. 3. CNN, PCNN, and the two models with SelATT are all vulnerable to FN instances. As IRMIE fails to exclude enough FNs from the negative set on TACRED,[6] its performance is also strongly influenced by FN instances. Although the F1 scores of H-FND are 2% behind co-teaching and cleanlab for FN ratios from 0% to 20%, it successfully maintains its performance when the FN ratio exceeds 30% and becomes competitive with these two baselines. This is similar to the experimental results on SemEval for FN ratios less than 30%. Together with the fact that TACRED has many more positives than SemEval, we increased the FN ratio to 90%. The result of this extended experiment shows that when the FN ratio exceeds 60%, the F1 scores for co-teaching drop significantly, whereas H-FND maintains a relatively high F1 score. Here, again, although cleanlab performs similar to ours with the predefined FN ratios,[7] the proposed approach needs no such information, which better fits real-world circumstances of distant-supervised relation classification.

4.4 Ablation Study

Fig. 4 shows the result of the ablation study to justify the effectiveness of the *Revise* action and pretraining strategy. On Semeval, pretraining boosts the F1 score for the PCNN architecture for FN ratios from 10% to 40%, but yields no significant difference for the other ratios. On TACRED, however, the *Revise* action and the pretraining strategy clearly yield improved results. This improvement is substantial in particular for pretraining. As TACRED has more positive relation types and a much larger negative set, the FN denoising problem is more severe than on SemEval; thus the pretraining strategy is crucial to provide a better initial point for the denoising agent and to ensure more stable performance.

4.5 Detailed Analysis

We first analyzed the distribution of the denoising policy for TN and FN instances in the training set. Figure 5 shows the percentage of kept, discarded,

[6]The size of the RAW set is less than 10% of the original negative set under all FN ratios.

[7]We have measured the performance of cleanlab when it was provided with a wrong FN ratio - 40% FN ratio. Under the actual FN ratio of 80% , its F1 scores dropped by 0.5% for CNN and 1.8% for PCNN.

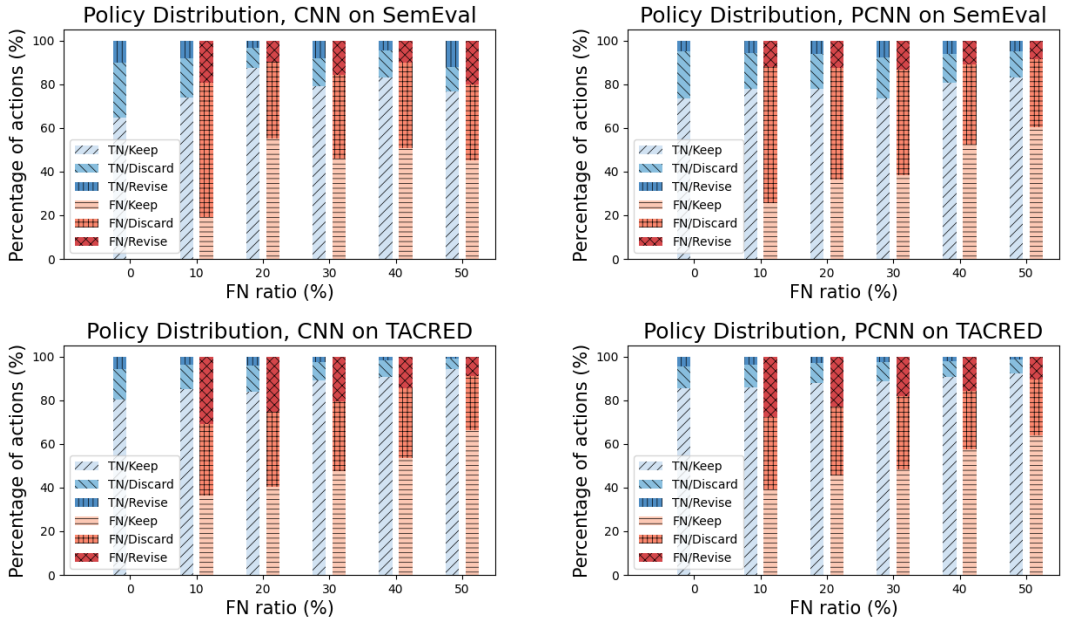

Figure 5: Denoising policy distribution on true negatives and false negatives.

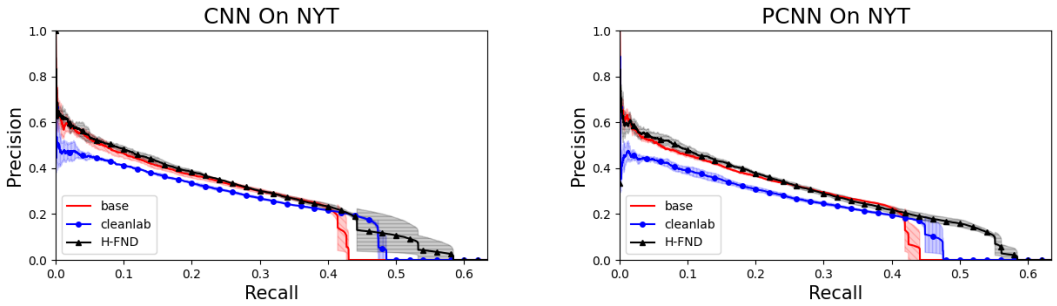

Figure 6: Precision-recall curve on the NYT dataset. The shaded areas indicate one standard deviation. The precision rate of each algorithm run drops to zero at certain recall rate, hence the steep drops in the curves.

or revised training instances. The left histogram under each filter ratio is for TN; the right is for FN.

On SemEval, we observe that for TN instances, H-FND mainly keeps them as NA and revises only a small portion to the wrong relation, even under the 50% filter ratio. For FN instances, H-FND prefers to discard or revise them. This difference shows that H-FND distinguishes FN instances from TN instances, and does not take arbitrary actions on them.

On TACRED, the policy distribution also shares the same tendency, but the portion of kept instances is generally larger. This is due to a higher ratio of negative instances in TACRED. As more negative instances result in more *Keep* labels in the generated pretraining data, after pretraining, the probability of the model taking the *Keep* action is generally higher. It also explains that the portion of kept instances grows when the filter ratio is raised. Note that this prevents H-FND from revising too many instances at the beginning of co-training, making co-training more stable.

Table 3 show the correctness of revisions on FN instances which are determined to be revised. The accuracy is around 90% for both CNN and PCNN architectures and for both SemEval and TACRED. This shows that H-FND accurately corrects FN instances once they are identified and determined to be revised in the first stage.

4.6 Results on Realistic Dataset

Lastly, we evaluated H-FND on real DS dataset NYT10. For baselines, apart from the base model, we included cleanlab, as it is the best performing baseline in the controlled FN experiments. We conducted human evaluation on 200 negative instances randomly sampled from the training set

SemEval	10%	20%	30%	40%	50%
CNN	88.81	88.07	88.72	86.25	85.94
	± 3.55	± 7.80	± 1.04	± 1.53	± 1.46
PCNN	89.54	88.04	86.83	90.31	84.17
	± 1.98	± 3.12	± 2.11	± 0.54	± 3.56
TACRED	10%	20%	30%	40%	50%
CNN	91.43	90.62	90.89	91.65	93.39
	± 0.72	± 0.99	± 0.63	± 0.99	± 1.79
PCNN	90.99	89.64	87.15	86.75	86.15
	± 0.82	± 0.39	± 0.49	± 0.60	± 1.16

Table 3: Revision accuracy (%)

and came to an estimate of 14% noise. [8]

We followed Zeng et al. (2015) and plotted the precision-recall curve to demonstrate the result on NYT10 (see Fig. 6). At recall rate lower than 40% cleanlab performs slightly worse than the base model, while H-FND remains competitive in terms of precision. This could be a result of inaccuracies in the estimation of FN rate in the dataset. Since H-FND does not require a given FN rate, it is not encumbered by such estimation error. At higher recall rates ($> 50\%$), H-FND retains significantly higher precision. This result shows that H-FND is applicable for real DS datasets, especially when the recall rate matters.

5 Conclusion and Future Work

In this work, to increase the robustness of DS, we present H-FND, a hierarchical false-negative denoising framework, which keeps, discards, or revises non-relation (NA) inputs during training and validation phases to suppress noise from FN instances. We also present a special transfer learning scheme for pretraining the denoising agent.

To investigate the effects of FN instances addressed by our approach, we generate FN instances from SemEval-2010 and TACRED under controlled ratios. The results show that H-FND revises FN instances and facilitates robust relation extraction. Further experiment on NYT10 demonstrates that our framework is also applicable to realistic DS setting.

In realistic DS setting, both FP and FN instances may emerge simultaneously. Both of which should be addressed. We leave this as future work. Also, we plan to attempt other advanced relation classification approach like R-BERT (Wu and He, 2019) to replace CNN or PCNN in our architecture.

[8] This also demonstrates our synthesized datasets are a good approximation to realistic DS setting: for the NYT10, around 72% of the instances are negatives. This gives us around (0.72 * 0.14) = 10.1% of FN in all triples. For our synthetic dataset, the noise rate is the percent of positive relation triples flipped to create false negatives. Positive triples make up only around 20% of the whole TACRED, and in our experiments, the noise rate indicates that we have flipped 50% of the triples. This gave us a total of around (0.2 * 0.5) = 10% of FN in all triples.

6 Acknowledgements

We are grateful for the insightful comments from anonymous reviewers. This work is supported by the Ministry of Science and Technology of Taiwan under grant numbers MOST110-2634-F-001-011.

References

Paszke Adam, Gross Sam, Chintala Soumith, Chanan Gregory, Yang Edward, DeVito Zachary, Lin Zeming, Desmaison Alban, Antiga Luca, and Lerer Adam. 2017. Automatic differentiation in PyTorch. In *Neural Information Processing Systems workshop: The future of gradient-based machine learning software & techniques*.

Enrique Alfonseca, Katja Filippova, Jean-Yves Delort, and Guillermo Garrido. 2012. Pattern learning for relation extraction with a hierarchical topic model. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 54–59, Jeju Island, Korea. 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 Thirty-Second AAAI Conference on Artificial Intelligence*.

Bo Han, Quanming Yao, Xingrui Yu, Gang Niu, Miao Xu, Weihua Hu, Ivor Tsang, and Masashi Sugiyama. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *Advances in neural information processing systems*, pages 8527–8537.

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.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S. Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 541–550, Portland, Oregon, USA. Association for Computational Linguistics.

Matthew Honnibal and Mark Johnson. 2015. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378.

Wei Jia, Dai Dai, Xinyan Xiao, and Hua Wu. 2019. ARNOR: Attention regularization based noise reduction for distant supervision relation classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1399–1408, Florence, Italy. Association for Computational Linguistics.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

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, Berlin, Germany. Association for Computational Linguistics.

Mike Mintz, Steven Bills, Rion Snow, and Daniel 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, Suntec, Singapore. Association for Computational Linguistics.

Raymond J Mooney and Razvan C Bunescu. 2006. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the NAACL Workshop on Vector Space Modeling for Natural Language Processing*.

Curtis G Northcutt, Lu Jiang, and Isaac L Chuang. 2019. Confident learning: Estimating uncertainty in dataset labels. *arXiv preprint arXiv:1911.00068*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018a. DSGAN: Generative adversarial training for distant supervision relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 496–505, Melbourne, Australia. Association for Computational Linguistics.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018b. 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, Melbourne, Australia. Association for 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.

Roland Roller, Eneko Agirre, Aitor Soroa, and Mark Stevenson. 2015. Improving distant supervision using inference learning. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 273–278, Beijing, China. Association for Computational Linguistics.

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *Proceedings of the 2013 workshop on Automated knowledge base construction*, pages 73–78.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 455–465.

Richard S Sutton, David A McAllester, Satinder P Singh, and Yishay Mansour. 2000. Policy gradient methods for reinforcement learning with function approximation. In *Advances in neural information processing systems*, pages 1057–1063.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Shanchan Wu and Yifan He. 2019. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. 2013. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 665–670, Sofia, Bulgaria. Association for Computational Linguistics.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. *Journal of machine learning research*, 3:1083–1106.

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*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2335–2344.

Xiangrong Zeng, Shizhu He, Kang Liu, and Jun Zhao. 2018. Large scaled relation extraction with reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, pages 5658–5665.

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, 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, Ann Arbor, Michigan. Association for Computational Linguistics.

A Appendices

A.1 Convolutional Neural Network

We use a convolutional neural network (CNN) (Nguyen and Grishman, 2015) as our base model for both the denoising agent and the relation classifier. This architecture consists of four main layers (the first three layers compose the CNN encoder):

1. **Embedding**: The embedding layer transforms a word into a vector representation, which is a concatenation of a word embedding V_w and a pair of positional embedding vectors V_{p_1}, V_{p_2} (Lin et al., 2016). Word embedding V_w is a vector that represents the semantics of a word, and positional embedding pair V_{p_1}, V_{p_2} is two vectors representing the relative distance from the current word to two entities in the sentence.

   The final embedding vector V of dimension d_e for each word is the concatenation of V_w, V_{p_1}, and V_{p_2}:

   $$V = [V_w | V_{p_1} | V_{p_2}].$$

2. **Convolution**: The convolutional layer transforms the embedding vectors of words into local features by applying sliding filters over them. Each filter consists of a weight matrix $A_i \in \mathbb{R}^{f \times d_e}$ and a bias term $b_i \in \mathbb{R}$, to extract specific patterns in the embedding vectors. With h filters of length f, the entry in the feature map $C_f \in \mathbb{R}^{h \times (L-f+1)}$ for the i-th filter at position t is

   $$[C_f]_{it} = \sum_{j=1}^{f} \sum_{k=1}^{d_e} A_{ijk} \cdot V_{t+j-1,k} + b_i,$$

   where L is the length of the input sentence. To capture information expressed in phrases of all lengths, we further use n different lengths of filters, and concatenate all C_f under filter size f as the jointed feature map $C \in \mathbb{R}^{nf \times d_e}$:

   $$C = [C_{f_1} | C_{f_2} | \cdots | C_{f_n}].$$

3. **Max pooling**: The max pooling layer captures the most significant feature into the pooling feature P_i by selecting the highest value in the feature map extracted by the i-th filter C_i over all positions:

   $$P_i = \max(C_i).$$

   PCNN (Zeng et al., 2015) involves piecewise max pooling, which better suits the relation extraction task. It divides an input sentence into three segments based on the two selected entities, and then extracts features from all the three segments to capture fine-grained features for relation extraction. For PCNN, the extracted feature map

   $$P_i = [\max(C_{i1}) | \max(C_{i2}) | \max(C_{i3})],$$

   where C_{i1}, C_{i2}, and C_{i3} are the three feature map segments separated by the two selected entities. We also view P as the sentence feature, as it represents the essential features of the whole sentence.

4. **Fully connected**: The fully connected layer (FC) performs relation classification based on sentence feature P with softmax activation over each relation. The computed logits $O(r)$ is written as

   $$\begin{aligned} O(r) &= \text{softmax}(\text{FC}(P)) \\ &= \text{softmax}(\text{FC}(\text{CNN}(s))). \end{aligned}$$

A.2 Implementation

H-FND was implemented with PyTorch 1.6.0 (Adam et al., 2017) in python 3.6.9. In our implementation, we used pretrained word embeddings provided by SpaCy (Honnibal and Johnson, 2015) as the fixed word embeddings ($d_w = 300$). The positional embedding ($d_p = 50$) was randomly initialized and then trained with the following network; therefore the overall dimension of embedding vector $d_e = d_w + 2d_p = 400$. In the convolutional layer, we applied four different sizes of filters ($f \in [2, 3, 4, 5]$) and set all of their feature sizes to $h = 230$. Both CNN and PCNN architectures were implemented. The total trainable parameters of each models are listed in table 4. To prevent overfitting, we inserted dropout layers with a dropout rate of 0.5 before the convolutional layer and after the max pooling layer.

We trained H-FND using the Adam optimizer (Kingma and Ba, 2015). In addition, we

#(Params)		SemEval	TACRED	NYT
RC	CNN	1,318,130	1,347,602	1,388,933
	PCNN	1,336,530	1,424,882	1,486,453
RC+SelATT	CNN	1,327,330	1,386,242	–
	PCNN	1,364,130	1,540,802	–
DA	CNN	1,311,683	1,311,683	1,342,883
	PCNN	1,317,203	1,317,203	1,348,403

Table 4: Number of trainable parameters in each model.

used mini-batches (batch size $b = 256$) only when training the relation classifier; the prediction of the relation classifier and both the decision and policy gradient of the denoising agent were executed per epoch. Last, the revised result of H-FND in each epoch was used by the classifier only in the same epoch and did not accumulate over epochs, which means that at the beginning of each epoch, H-FND applied the denoising policy on the original dataset but not on the revised dataset of the last epoch.

We list in Table 5 the learning rates for base CNN and PCNN relation classifiers (RC), for RC with SelATT, and for RC with denoising agent (DA) under pretraining and co-training phrases. The learning rate of RC is selected from {1e-4, 3e-4, 1e-3, 3e-3, 1e-2}, with the F1 score on the noise-free version of SemEval and TACRED as the selection criteria. Except SelATT and DA co-training, the learning rates for the other models are the same to the learning rate of base RC. For SelATT, the learning rate is selected from {1e-6, 3e-6, 1e-5, 3e-5, 1e-4}, also with the F1 score on the noise-free version of the two datasets as the selection criteria. For DA cotraining, the learning rate is selected from {1e-6, 3e-6, 1e-5, 3e-5, 1e-4}, with the F1 score on the SemEval and TACRED under a 50% FN ratio as the selection criteria.

All the RC of each method are trained to converge with validation-based early stopping. In specific, we train all the model for 150 epochs on SemEval and for 200 epochs on TACRED. For NYT, we trained all the models for 30 epochs.

The pretraining of H-FND trains the RC and DA for 5 and 20 epochs respectively. We select these pretraining periods by the criteria that the two models can achieve about 80% performance comparing to the converged ones. By this means, we can prevent H-FND from overfitting the noisy labels (Han et al., 2018) and initialize H-FND with good parameters for co-training.

All the implemented models are trained on NVIDIA GTX 1080 Ti and Intel(R) Xeon(R) Silver 4110 CPU, with 12GN GPU memory, 128GB RAM, clock rate 2.10 GHz, and Linux as the operating system. The expected running time for each model on each dataset is listed in Table 6.

Learning rate	SemEval	TACRED	NYT
lr_{RC}	3e-3	3e-4	3e-4
$lr_{\text{RC, SelATT}}$	1e-5	3e-6	–
$lr_{\text{RC, pre}}$	3e-3	3e-4	3e-4
$lr_{\text{DA, pre}}$	3e-3	3e-4	3e-4
$lr_{\text{RC, co}}$	3e-3	3e-4	3e-4
$lr_{\text{DA, co}}$	1e-4	3e-6	3e-6

Table 5: Learning rates.

Runtime	SemEval	TACRED	NYT
Base	0.05	0.63	3.25
SelAtt	1.95	22.70	–
IRMIE	0.05	0.67	–
Co-teaching	0.10	1.10	–
Cleanlab	0.25	6.47	16.25
H-FND	0.55	15.28	44.44

Table 6: Runtimes for models training (hrs).

A.3 Performance on Validation Set

The F1 scores of each model running on validation sets of SemEval and TACRED are provided in Figure 7 and 8. Notice that the validation sets are noisy in our experiment, so the performance on validation sets do not fully reflect the robustness of each models. Also, in IRMIE and H-FND, the validation sets are modified, so their validation F1 scores can only be compared with their own across different FN ratios. For more accurate performance measurement, please refer to Figure 3 and 4, whose F1 scores are measured on noise-free testing sets.

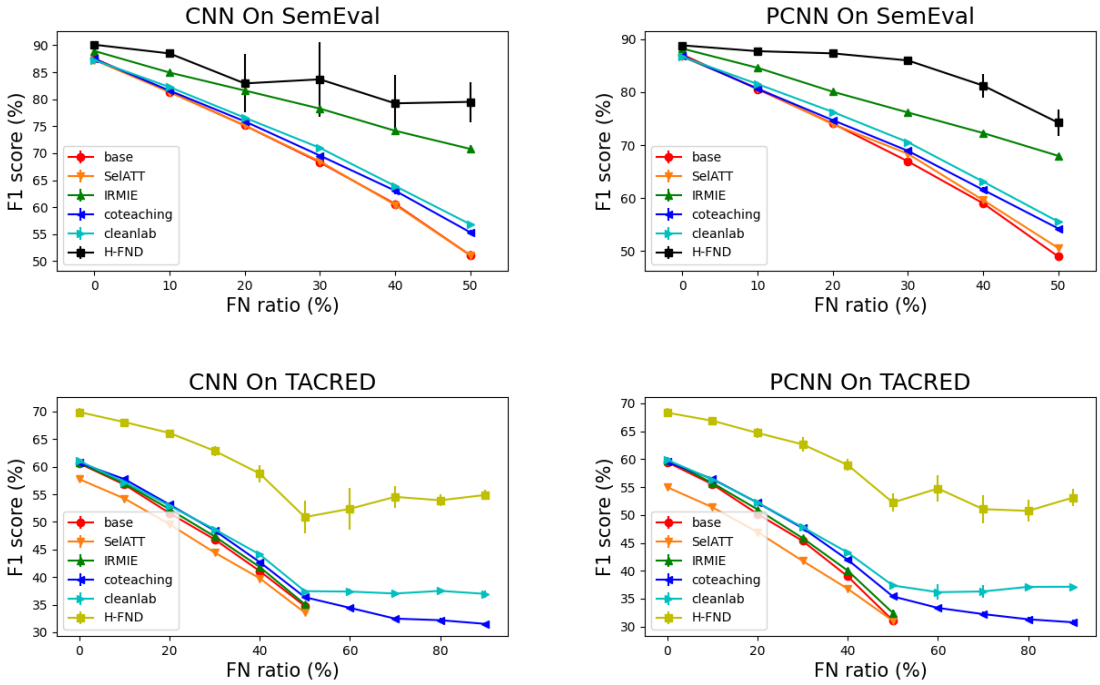

Figure 7: Validation F1 scores of quantitative result, where the errorbars represent the standard deviations.

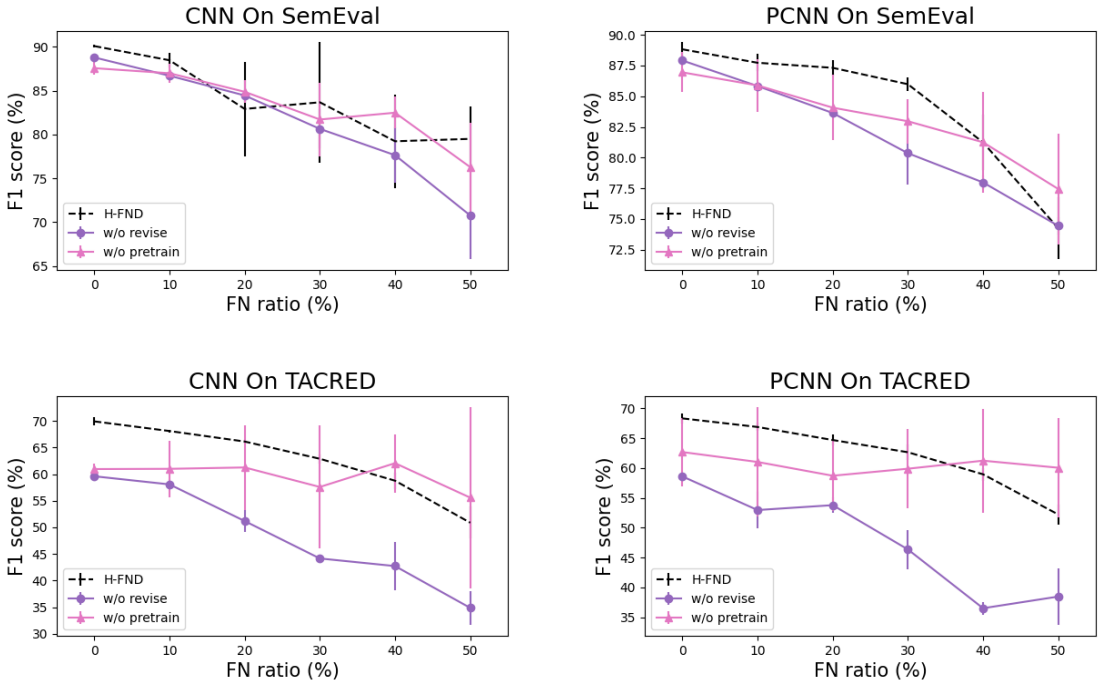

Figure 8: Validation F1 scores of ablation analysis, where the errorbars represent the standard deviations.

A.4 Denoising policy with Standard Deviations

On SemEval and TACRED, the Denoising policy distributions with standard deviation are provided in Table 7, 8, 9, and 10.

CNN on SemEval	0%	10%	20%	30%	40%	50%
TN/Keep	64.89 ± 6.55	74.04 ± 8.37	87.68 ± 9.74	79.17 ± 10.65	83.03 ± 6.65	76.83 ± 11.19
TN/Discard	24.88 ± 9.59	17.82 ± 5.55	8.81 ± 7.28	12.56 ± 6.53	12.34 ± 4.84	11.17 ± 3.15
TN/Revise	10.23 ± 5.00	8.14 ± 5.90	3.51 ± 2.57	8.27 ± 4.47	4.63 ± 1.93	12.00 ± 9.25
FN/Keep	0.00 ± 0.00	19.04 ± 5.65	55.03 ± 26.98	45.97 ± 24.85	50.91 ± 13.15	45.24 ± 11.06
FN/Discard	0.00 ± 0.00	61.58 ± 10.16	35.29 ± 22.58	38.33 ± 17.48	38.81 ± 10.05	34.70 ± 5.73
FN/Revise	0.00 ± 0.00	19.38 ± 9.77	9.68 ± 5.52	15.70 ± 7.49	10.28 ± 3.75	20.05 ± 10.99

Table 7: Denoising policy distribution for CNN on SemEval (%).

PCNN on SemEval	0%	10%	20%	30%	40%	50%
TN/Keep	73.57 ± 6.19	77.79 ± 4.11	77.74 ± 3.93	73.61 ± 5.11	80.72 ± 6.08	82.95 ± 4.45
TN/Discard	21.63 ± 4.49	16.55 ± 4.71	16.24 ± 3.78	18.76 ± 5.51	13.25 ± 4.96	12.15 ± 4.21
TN/Revise	4.79 ± 2.08	5.67 ± 1.89	6.01 ± 2.16	7.62 ± 2.14	6.04 ± 2.17	4.91 ± 0.68
FN/Keep	0.00 ± 0.00	25.37 ± 5.33	36.46 ± 7.12	38.25 ± 5.81	52.36 ± 9.94	60.38 ± 8.54
FN/Discard	0.00 ± 0.00	62.62 ± 8.65	51.12 ± 8.63	48.45 ± 8.44	36.59 ± 8.87	31.31 ± 8.13
FN/Revise	0.00 ± 0.00	12.00 ± 3.51	12.42 ± 3.42	13.30 ± 3.36	11.05 ± 3.10	8.31 ± 1.42

Table 8: Denoising policy distribution for PCNN on SemEval (%).

CNN on TACRED	0%	10%	20%	30%	40%	50%
TN/Keep	80.48 ± 3.49	85.18 ± 1.36	84.07 ± 4.66	89.14 ± 1.38	90.81 ± 1.95	94.11 ± 2.23
TN/Discard	13.99 ± 2.76	10.99 ± 1.16	11.78 ± 2.36	8.50 ± 1.18	7.60 ± 1.61	5.00 ± 1.92
TN/Revise	5.54 ± 0.84	3.82 ± 0.32	4.15 ± 2.44	2.35 ± 0.37	1.59 ± 0.37	0.90 ± 0.35
FN/Keep	0.00 ± 0.00	36.53 ± 2.21	40.36 ± 1.88	47.42 ± 3.88	53.60 ± 4.16	66.31 ± 7.85
FN/Discard	0.00 ± 0.00	32.40 ± 3.08	34.23 ± 3.38	32.12 ± 3.42	31.86 ± 2.45	24.60 ± 5.73
FN/Revise	0.00 ± 0.00	31.07 ± 1.70	25.42 ± 1.79	20.46 ± 1.86	14.54 ± 1.83	9.09 ± 2.31

Table 9: Denoising policy distribution for CNN on TACRED (%).

PCNN on TACRED	0%	10%	20%	30%	40%	50%
TN/Keep	85.31 ± 0.45	85.91 ± 3.20	88.05 ± 2.73	88.60 ± 2.54	90.63 ± 2.23	92.46 ± 1.57
TN/Discard	10.30 ± 0.53	10.55 ± 2.97	9.10 ± 2.39	9.01 ± 2.07	7.41 ± 2.00	6.27 ± 1.40
TN/Revise	4.39 ± 0.34	3.54 ± 0.32	2.85 ± 0.53	2.39 ± 0.70	1.96 ± 0.31	1.26 ± 0.25
FN/Keep	0.00 ± 0.00	39.10 ± 4.23	45.62 ± 4.58	48.53 ± 6.01	57.50 ± 3.90	64.01 ± 4.44
FN/Discard	0.00 ± 0.00	33.03 ± 5.09	31.68 ± 3.64	33.32 ± 4.97	26.67 ± 3.58	25.55 ± 3.42
FN/Revise	0.00 ± 0.00	27.87 ± 1.88	22.70 ± 1.86	18.15 ± 2.82	15.83 ± 1.44	10.45 ± 1.59

Table 10: Denoising policy distribution for PCNN on TACRED (%).