H-FND: Hierarchical False-Negative Denoising for Robust Distantly-Supervised Relation Extraction

Tsu-Jui Fu
Academia Sinica
s103062110@m103.nthu.edu.tw

Wei-Yun Ma
Academia Sinica
ma@iis.sinica.edu.tw

Abstract

Distant supervision is a popular method for relation extraction, but is vulnerable to false-negative (FN) sentences. Treating FN input as non-relation training sentences can lead to degraded final model performance. Here, we present H-FND, a hierarchical false-negative denoising framework for robust relation extraction. H-FND uses a hierarchical policy which first determines whether non-relation (NA) sentences should be kept, discarded, or revised during training. Then the policy revises those into appropriate relations for better training input. We conducted experiments on SemEval-2010 and randomly filtered ratios of training/validation sentences into NA. The results show that H-FND revises FN sentences correctly and maintains high F1 scores even when 50% of the sentences have been filtered as NA.

1 Introduction

Relation extraction (Zelenko et al., 2003; Bunescu and Mooney, 2005; GuoDong et al., 2005) is a core task in information extraction, the goal of which is to determine the relation between two entities in a given sentence. For instance, given "Jobs founded apple", the relation to be extracted is "founder".

A major issue in relation extraction is data sparsity, against which distant supervision (Hoffmann et al., 2011; Surdeanu et al., 2012) is a useful labeling method. With knowledge bases which include entities and relations, distant supervision produces a large amount of training sentences if they contain entities in the entity knowledge base and their relation type is labeled directly from the relation knowledge base.

Despite the large number of labels in a distant supervision dataset, they are actually very noisy (Roth et al., 2013). Under distant supervision, there are two main problems: false-positives (FP) and false-negatives (FN). Table 1 shows the example. For FP, "Jobs ate an apple" should not reflect the relation between them, but distant supervision still labels them as "founder". For FN, as there is no relation between "Gates" and "Microsoft" in the knowledge base, "Gates created Microsoft" is labeled as a non-relation (NA). Both FP and FN are incorrectly labeled; this leads to degraded final model performance if they are treated as correct labels.

Sentence	Relation	Type
Jobs founded **Apple**	Founder (✓)	True-Positive
Jobs ate an **apple**	Founder (✗)	False-Positive
Gates created **Microsoft**	NA (✗)	False-Negative

Table 1: Different types of incorrectly labeled relations

There are many previous works (Lin et al., 2016; Qin et al., 2018a,b) that focus on the FP problem where they select one sample during training, apply sentence-level soft attention over noisy sentences, or even move probable FP sentences into NA. However, there is no work investigating the FN problem for relation extraction. Excessive FN sentences lead to low recall in the final model.

In this paper, we investigate the FN problem and propose H-FND, a hierarchical false-negative denoising framework which keeps, discards, or revises probable FN sentences and produces a robust training/validation set. Our contributions are three-fold:

- We propose a denoising framework focused on false-negative problem in relation extraction.
- Our method is model-independent and can thus be applied to any state-of-the-art model.
- We show that our method revises correctly and maintains high F1 score even under a high filter ratio.

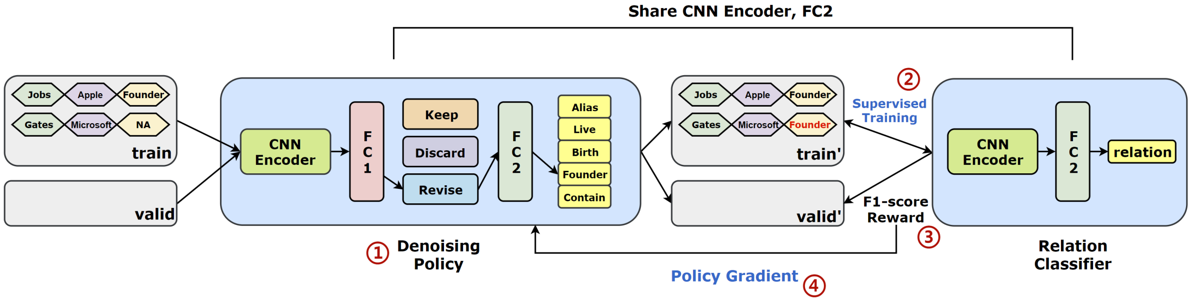


Figure 1: H-FND framework

2 Methodology

As illustrated in Fig. 1, H-FND is composed of a relation classifier and a hierarchical denoising policy. Both share a CNN encoder which extracts the sentence features of an input sentence.

2.1 Relation Classifier

A relation classifier predicts the relation between two entities in a given sentence. The 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, including a word embedding (WE) and positional embeddings (PE). The PE is defined as the relative distance from the current word to two entities. The final input vector V for each word is concatenated as $V = [WE|PE_1|PE_2]$.
2. **Convolutional**: The convolutional layer transforms the input vectors into feature maps by sliding filters over them. Assuming a filter size of f, we obtain the sequence C_f based on f-gram in sequence $V_{1:L}$.
   $C_{f,j} = W_f \cdot V_{j-f+1:f} + b_f$, where L is the length of the sentence and W_f and b_f are the trainable parameters of filter f. We concatenate all C_f under different filter size f as the convolutional feature $C = [C_{f_1}|C_{f_2}|\cdots|C_{f_n}]$.
3. **Max pooling**: The max pooling layer captures the most significant features into the pooling feature P by selecting the highest value in each feature map of C, where $P = \max(C)$.
   PCNN (Zeng et al., 2015) involves piecewise max pooling, which better suits the relation extraction task.
   $P = [\max(C_{1:e1})|\max(C_{e1:e2})|\max(C_{e2:L})]$, where $e1$ and $e2$ are the positions of two 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, where $O = \text{softmax}(\text{FC2}(P))$.

2.2 Hierarchical Denoising Policy

The proposed denoising policy in H-FND is a hierarchical model which first determines that an NA sentence should be kept, discarded, or revised:

- **Keep**: maintain NA during training/validation;
- **Discard**: remove the sentence to prevent it from misleading the model;
- **Revise**: as the sentence should not be NA, we predict its relation type and use it during the following training/validation.

Then, if it is to be revised, the policy determines an adequate relation as its revised relation. Note that we apply the denoising policy only on NA sentences.

In the determination step, we classify it into three classes based on the sentence feature P from the CNN encoder. The revising step predicts the relation type for probable false-negative (FN) sentences also based on their sentence feature P.

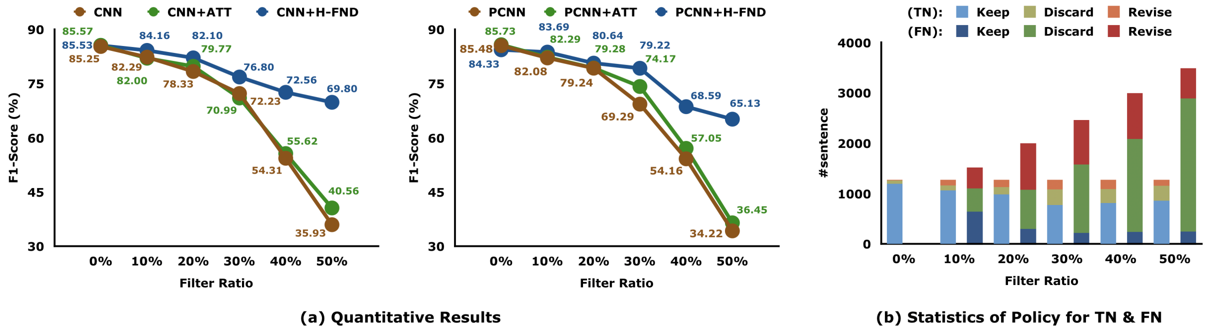
$$[\text{Keep}, \text{Discard}, \text{Revise}] = \text{softmax}(\text{FC1}(P))$$
$$\text{Revised Relation} = \text{softmax}(\text{FC2}(P))$$

To constrain the relation prediction, we share the FC2 between the relation classifier and the denoising policy.

We sample based on the output probability of the determination step and the revise step to decide how to denoise each NA sentence.

2.3 Co-Training Framework

To ensure the denoising policy benefits the CNN model, and to combine both, we propose the following training framework during each epoch:



(a) Quantitative Results

(b) Statistics of Policy for TN & FN

Figure 2: Experimental results; (a) quantitative results of CNN and PCNN; (b) statistics of denoising policy.

1. **Dataset denoising**: At the beginning of each epoch, H-FND applies the denoising policy on dataset first. For the training set, the policy keeps, discards, or revises the NA sentences. For the validation set, to obtain better results, we choose only to keep or revise those NA relations. After denoising, we have the revised training/validation set.
2. **CNN model training**: Given the revised training set, we train the CNN model in a supervised fashion based on the categorical loss:

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

   where G represents the revised ground-truth relation in training set.
3. **Reward determination**: We evaluate trained CNN model on the revised validation set to obtain the F1 score, which we view as reward R. As the denoising policy does not discard any sentences in the validation set, R reflects the actual performance of the trained CNN model.
4. **Denoising policy update**: Since the sampling step in the denoising policy is undifferentiable, here we adopt policy gradient (Sutton et al., 1999) to optimize the denoising policy by maximizing R:

   $$\nabla_\theta \approx \sum \log p(a|\theta)(R - b)$$
   $$ls_{RL} = -\nabla_\theta,$$

   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 problem of the REINFORCE algorithm (Williams, 1992). We treat b as 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, after which H-FND finds the best denoising policy adaptively between supervised training and reward maximization.

2.4 Implementation

In our implementation, we adopted pre-trained GloVe (300d) (Pennington et al., 2014) embeddings as the fixed word embeddings; the positional embedding (50d) was randomly initialized, and then trained with the following network. For PCNN, we applied four CNN filters ([2, 3, 4, 5]) and set all of their feature sizes to 230.

We used a dropout rate of 0.5 and learning rates of $ls_{CNN} = 0.008$ and $ls_{RL} = 0.0005$. We trained H-FND using the Adam optimizer (Kingma and Ba, 2015) and implemented it under PyTorch (Adam et al., 2017).

3 Experimental Result

3.1 Experimental Settings

To focus on the false negative problem, we evaluated the proposed H-FND on SemEval-2010 (Hendrickx et al., 2009), which is well-labeled and contains nine relations with an additional NA as non-relation. Experimenting on well-labeled SemEval-2010, rather than distant-supervision datasets like NYT (Riedel et al., 2010), prevents from the effects caused by the false-positive problem.

We used 10% of the training set for validation. To simulate FN conditions, we randomly filtered a ratio (10%–50%) of training/validation sentences into NA. Note that the filter process was only for training/validation: the testing set was well-labeled under all filter ratios. Also note that we did not know in advance which sentences were true negative (TN) and which were FN: in the training/validation set, both kinds were NA.

H-FND is implemented with both CNN and PCNN; compared with ATT (Lin et al., 2016), it is an effective sentence-level attention method for improving wrong labels in distant supervision.

Sentence	Action	Groundtruth
The system has its greatest application in an arrayed **configuration** of antenna **elements**.	**revise** Component-Whole	Component-Whole
The **author** of a keygen uses a **disassembler** to look at the raw assembly code.	**revise** Instrument-Agency	Instrument-Agency
She placed the **bread** in a serving **basket** and passed it around the classroom.	**revise** Content-Container	Entity-Destination
The **star** was blinded from his own **insight** about the incommensurability of time.	**keep**	NA
The size of the thumbnails is changed with a sliding **bar** on the upper right **corner** of the window.	**discard**	NA

Table 2: Example denoising results. Entities 1 and 2 shown in red and blue, respectively.

3.2 Quantitative Results

The quantitative results are shown in Fig. 2(a), including both CNN and PCNN.

We observe that for both the original CNN and CNN with ATT, the F1 score is heavily influenced by FN sentences: the performance drops to 40%. In contrast, for the proposed H-FND, since we can discard or even revise FN sentences, the performance is maintained under all filter ratios and yields almost a 70% F1 score even when 50% of sentences have been filtered out as NA.

A similar tendency can be found in PCNN. The F1 score for PCNN and PCNN with ATT is as low as 35% but PCNN with H-FND maintains a 65% F1 score.

3.3 Detailed Analysis

Apart from the quantitative result, we also analyzed H-FND in detail. We first analyzed the denoising policy for TN and FN sentences in the training set. Fig. 2(b) shows the number of sentences which are kept, discarded, or revised. The left histogram under each filter ratio is for TN; the right is for FN.

We observe that for TN sentences, H-FND mainly keeps them as NA. H-FND keeps almost all NA sentence under the 0% filter ratio and revise only a small portion of TN sentences into the wrong relation, even under the 50% filter ratio. This shows that H-FND actually detects TN sentences, and does not revise them arbitrarily.

For FN sentences, with a larger filter ratio, H-FND prefers to discard or revise them as there are too many NA sentences. H-FND tends to revise more sentences under a smaller (10%–30%) filter ratio, but chooses to discard more under a larger (40%–50%) one.

This is intuitive, as under a smaller filter ratio, there are enough true-positive (TP) sentences to provide sufficient information to revise FN sentences into the correct relation. However, with a bigger filter ratio, discarding uncertain sentences seems a safer policy. If H-FND were to revise them into the wrong relation, it would hurt model performance far more than just discarding them would. The proposed training framework determines the best policy adaptively, according to the distribution of both TN and FN sentences in the training/validation set.

	10%	20%	30%	40%	50%
Training	81.3%	78.5%	77.6 %	78.2%	71.3%
Validation	72.9%	68.5%	69.1%	66.5%	64.7%

Table 3: Training/validation set revision accuracy

We further investigated the approach's revisions. Table 3 shows the correctness of revisions for both training and validation sets under all filter ratios. For the training set, more than 71% of H-FND's revisions are correct. For the validation set, as H-FND cannot discard them, although fewer are correct still more than 64% are correct even under a 50% filter ratio. This shows that H-FND actually corrects FN sentences for both training/validation.

Table 2 shows examples of denoising. The first two cases show that H-FND revises them into the right relation, "Component-Whole" and "Instrument-Agency". The third shows H-FND revising into the wrong relation; however, this still makes sense as "bread in a basket" also reflects the relation of "Content-Container".

The fourth case is a TN sentence: H-FND detects that it should actually be NA. As the fifth is more difficult, H-FND chooses to discard it, preventing the wrong label from hurting the model.

4 Conclusion and Future Work

In this work, we present H-FND, a hierarchical false-negative (FN) denoising framework which keeps, discards, or revises non-relation (NA) inputs during training and validation. We evaluate the framework on SemEval-2010 with ratios of sentences filtered out as NA. H-FND revises FN sentences into appropriate relations and maintains high F1 scores. In addition to FN, false-positive (FP) is also a critical problem: we view this as future work which considers both FN and FP simultaneously for more robust relation extraction.

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 *NIPS (workshop)*.

Razvan Bunescu and Raymond J Mooney. 2005. Subsequence kernels for relation extraction. In *NIPS*.

Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. 2005. Exploring various knowledge in relation extraction. In *ACL*.

Iris Hendrickx, Su Nam Kim, Zornitsa Kozareva, Preslav Nakov, Diarmuid O Seaghdha, Sebastian Pado, Marco Pennacchiotti, Lorenza Romano, and Stan Szpakowicz. 2009. Semeval-2010 task 8: multi-way classification of semantic relations between pairs of nominals. In *SEW*.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *ACL*.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR*.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. 2016. Neural relation extraction with selective attention over instances. In *ACL*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *EMNLP*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018a. Dsgan: Generative adversarial training for distant supervision relation extraction. In *ACL*.

Pengda Qin, Weiran Xu, and William Yang Wang. 2018b. Robust distant supervision relation extraction via deep reinforcement learning. In *ACL*.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

Benjamin Roth, Tassilo Barth, Michael Wiegand, and Dietrich Klakow. 2013. A survey of noise reduction methods for distant supervision. In *AKBC*.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D Manning. 2012. Multi-instance multi-label learning for relation extraction. In *EMNLP*.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *NIPS*.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Machine Learning*.

Dmitry Zelenko, Chinatsu Aone, and Anthony. 2003. Kernel methods for relation extraction. In *JMLR*.

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