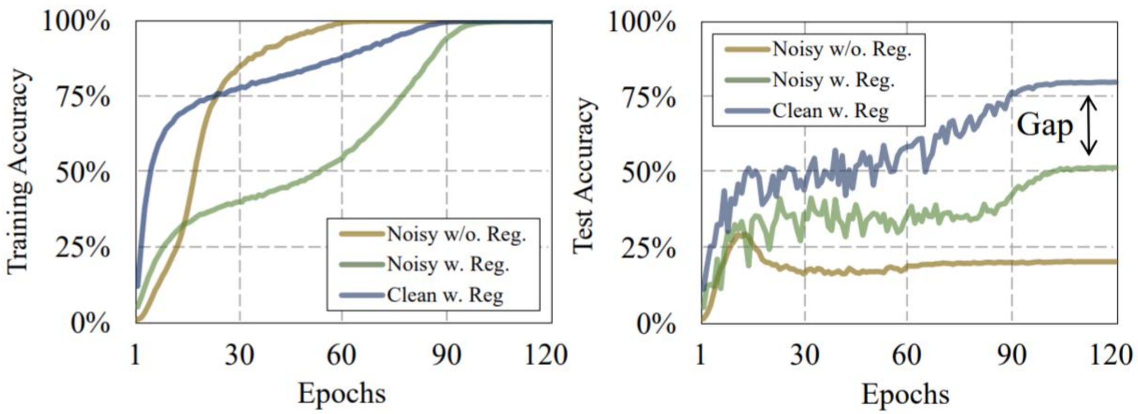
Learning from Noisy Labels for Entity-Centric Information Extraction

Wenxuan Zhou, Muhao Chen
University of Southern California

Noisy Labels

Labeling on large corpora inevitably introduces noisy (incorrect) labels. They can lead to degradation of model performance, and have affected on popular IE benchmarks.



Model performance decreases when trained with noisy labels*

Our focus: develop a model that is robust to noisy training labels.

*Source: Learning from Noisy Labels with Deep Neural Networks: A Survey

Task Definition

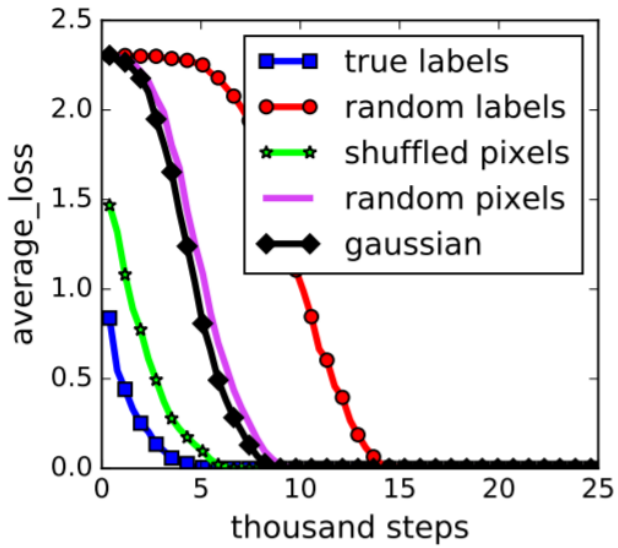
Noisily labeled data: Given a noisily labeled dataset D, an unknown subset $D_s \subset D$ is wrongly labeled (which portion being D_s is unknown to training).

Goal: Training a noise-robust model solely from D, i.e., with no additional resources, such as a clean validation set.

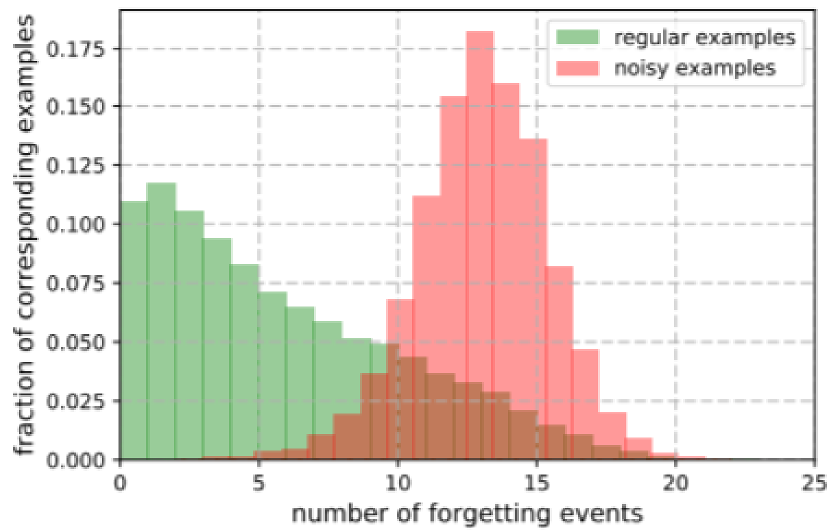
Tasks: In this work, we focus on two information extraction tasks, relation extraction (RE) and named entity recognition (NER).

Properties of Noisy Labels

P1. Take longer time to be learned by models.
P2. Easily forgotten in later epochs.



P1



P2

Noisy labels can be identified by their learning curve

Source: Understanding deep learning requires rethinking generalization, An Empirical Study of Example Forgetting during Deep Neural Network Learning

Motivation

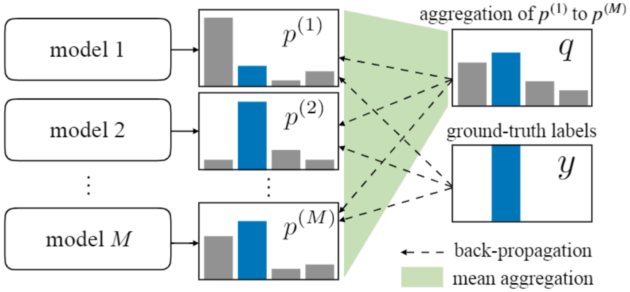
(1) Noisy labels take longer time to be learned.
(2) Noisy labels are frequently forgotten.

They are outliers to the task inductive bias.

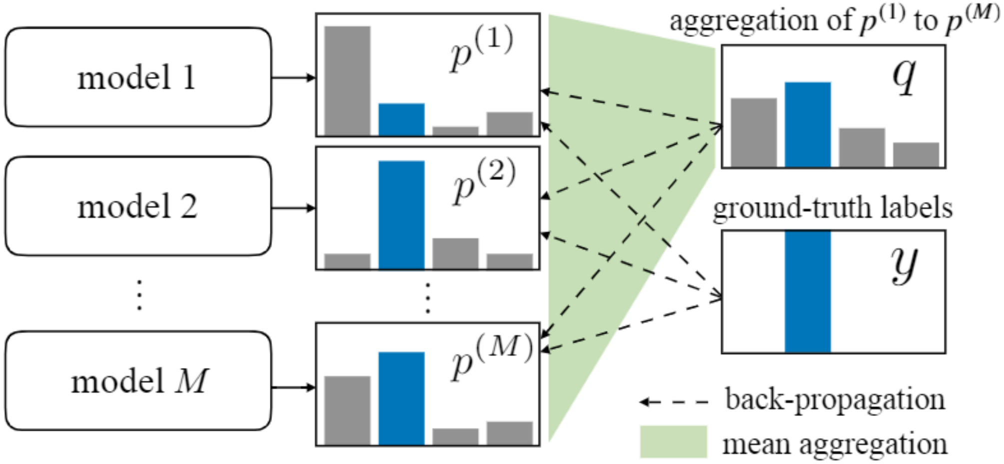
Model prediction is often inconsistent or oscillates on noisy labels in later epochs.

Given $\boldsymbol{M}(\geq \mathbf{2})$ independently trained models, (with high probability) at least one prediction is inconsistent to the noisy label.

Co-regularization Framework



Framework



Algorithm:

1. Create $\boldsymbol{M}(\geq \mathbf{2}$; 2 is enough) identical neural models with **different initialization**.
2. Train the models with **the task loss** for certain steps (warm-up phase).
3. Train the models with both **the task loss** and an additional **agreement loss**.
4. Return a random model.

Agreement Loss

$$\boldsymbol{q}_i = \frac{1}{M}\sum_{k=1}^{M} \boldsymbol{p}_i^{(k)}, \qquad \text{Eq. 1}$$

$$d(\boldsymbol{q}_i||\boldsymbol{p}_i^{(k)}) = \sum_{j=1}^{C} \boldsymbol{q}_{ij} \log\left(\frac{\boldsymbol{q}_{ij}+\epsilon}{\boldsymbol{p}_{ij}+\epsilon}\right), \qquad \text{Eq. 2}$$

$$\mathcal{L}_{\mathrm{agg}} = \frac{1}{MN}\sum_{i=1}^{N}\sum_{k=1}^{M} d(\boldsymbol{q}_i||\boldsymbol{p}_i^{(k)}), \qquad \text{Eq. 3}$$

Encourage M models to generate similar label distribution

- Clean labels: predictions similar to labels ⇒ little effect on training
- Noisy labels: predictions different to labels ⇒ large L_{agg}, prevent overfitting on those labels.

Experiment Settings

Datasets: TACRED, CoNLL03

Baselines:

- RE: C-GCN, BERT (base, large), LUKE
- NER: BERT (base, large), LUKE

	Noisy rate
TACRED	6.62%
CoNLL03	5.38%

Experiments

Model	Original		Relabeled	
	Dev F_1	Test F_1	Dev F_1	Test F_1
C-GCN ♣ (Zhang et al., 2018)	67.2	66.7	74.9	74.6
C-GCN-CrossWeigh	67.8	67.4	75.6	75.7
C-GCN-CR	67.7	67.2	75.6	75.4
$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019)	69.1	68.9	76.4	76.9
$\text{BERT}_{\text{BASE}}$-CrossWeigh	71.3	70.8	79.2	79.1
$\text{BERT}_{\text{BASE}}$-CR	71.5	71.1	79.9	80.0
$\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2019)	70.9	70.2	78.3	77.9
$\text{BERT}_{\text{LARGE}}$-CrossWeigh	72.1	71.9	79.5	79.8
$\text{BERT}_{\text{LARGE}}$-CR	73.1	73.0	81.3	82.0
LUKE ♣ (Yamada et al., 2020)	71.1	70.9	80.1	80.6
LUKE-CrossWeigh	71.0	71.6	80.4	81.6
LUKE-CR	71.8	72.4	81.9	83.1

RE (TACRED)

Model	Original		Relabeled
	Dev F_1	Test F_1	Test F_1
$\text{BERT}_{\text{BASE}}$ (Devlin et al., 2019)	95.58	91.96	92.91
$\text{BERT}_{\text{BASE}}$-CrossWeigh	95.65	92.15	93.03
$\text{BERT}_{\text{BASE}}$-CR	95.87	92.53	93.48
$\text{BERT}_{\text{LARGE}}$ (Devlin et al., 2019)	96.16	92.24	93.22
$\text{BERT}_{\text{LARGE}}$-CrossWeigh	96.32	92.49	93.61
$\text{BERT}_{\text{LARGE}}$-CR	96.59	92.82	94.04
LUKE ♣ (Yamada et al., 2020)	97.03	93.91	95.60
LUKE–CrossWeigh	97.09	93.98	95.75
LUKE-CR	97.21	94.22	95.88

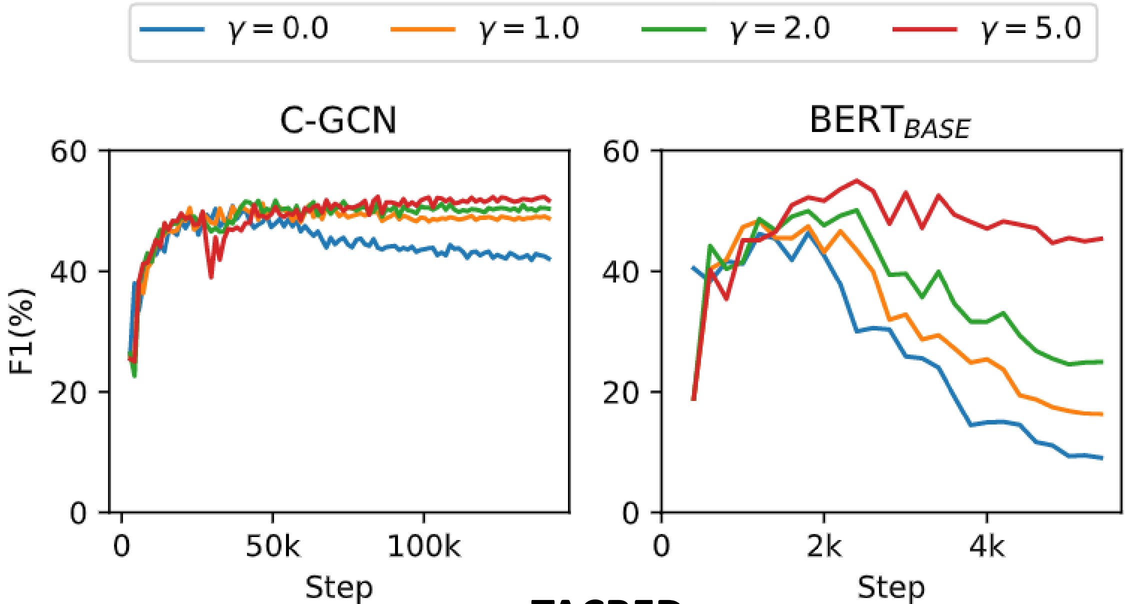
NER (CoNLL03)

- Co-regularization (CR) significantly outperforms compared baselines
- On larger pre-trained models, CR offers more prominent noising effects.

*Note: performance reported for CR w/ M=2 model copies

Noise Filtering Analysis

Training: clean + noisy labels
Test: noisy labels



TACRED

When using co-regularization ($\gamma > 0$), scores on test are much higher, indicating less over-fitting to noisy labels.

Different Noise Rates

Flipped labels (%)	10	30	50	70	90
$\text{BERT}_{\text{BASE}}$	74.2	70.8	62.9	48.6	0
$\text{BERT}_{\text{BASE}}$-CrossWeigh	77.3	75.6	71.6	61.3	25.1
$\text{BERT}_{\text{BASE}}$-CR	79.3	78.3	73.2	63.5	34.1
$\text{BERT}_{\text{BASE}}$ w/o flipped labels	76.5	74.9	72.9	70.8	57.4

TACRED

- The more noisy the training data are, the higher performance gain the co-regularization offers (in comparison to the base model).
- Co-regularization w/ only $\boldsymbol{M}$=2 model copies offer significantly better denoising than the ensemble-based CrossWeigh with **30** models.

Conclusion

1. We propose a co-regularization framework for learning supervised IE models with noisy labels.
2. Experiments on RE and NER demonstrate the effectiveness of our method.
3. Future work includes extending our framework to more IE tasks such as event extraction and coreference resolution.