DISTANTLY SUPERVISED RELATION EXTRACTION IN FEDERATED SETTINGS

Anonymous authors
Paper under double-blind review

ABSTRACT

Distant supervision is widely used in relation extraction in order to create a large-scale training dataset by aligning a knowledge base with unstructured text. Most existing studies in this field have assumed there is a great deal of centralized unstructured text. However, in practice, text may be distributed on different platforms and cannot be centralized due to privacy restrictions. Therefore, it is worthwhile to investigate distant supervision in the federated learning paradigm, which decouples the training of the model from the need for direct access to the raw text. However, overcoming label noise of distant supervision becomes more difficult in federated settings, because the sentences containing the same entity pair scatter around different platforms. In this paper, we propose a federated denoising framework to suppress label noise in federated settings. The core of this framework is a multiple instance learning based denoising method that is able to select reliable sentences via cross-platform collaboration. Various experimental results on New York Times dataset and miRNA gene regulation relation dataset demonstrate the effectiveness of the proposed method.

1 INTRODUCTION

Relation extraction (RE) aims to mine factual knowledge from free text by labeling relations between entity mentions, which is a crucial step in knowledge base (KB) construction. For example, given a sentence "$[\textit{Steve Jobs}]_{e_1}$ and Wozniak co-founded $[\textit{Apple}]_{e_2}$ in 1967", a relation extractor should identify that "*Steve Jobs*" and "*Apple*" are in a "*Founder*" relationship.

Most existing supervised RE systems, such as Zeng et al. (2014); Zhang & Wang (2015); Wang et al. (2016); Zhou et al. (2016), rely on a large-scale manually annotated training dataset, which is extremely expensive and cannot cover all walks of life. To ease the reliance on annotated data, Mintz et al. (2009) proposed distant supervision to automatically generate training data by heuristically aligning a KB with unstructured text. The key assumption of distant supervision is that if two entities have a relation in the KB, then all sentences that mention these two entities will express this relation. Since then, there has been a rich literature devoted to this topic, such as Riedel et al. (2010); Hoffmann et al. (2011); Zeng et al. (2015); Lin et al. (2016); Ye & Ling (2019); Yuan et al. (2019).

Though the progress is exciting, distant supervision approaches have so far been limited to the centralized learning paradigm, which assumes that a great deal of text is easily accessible. However, in practice, text may be distributed on different platforms and be massively convoluted with sensitive personal information, especially in the healthcare and financial fields (Yang et al., 2019; Zerka et al., 2020; Chamikara et al., 2020). Due to privacy restrictions, it is almost impossible or cost-prohibitive to centralize text from multiple platforms. Recently, federated learning (McMahan et al., 2016) provides a compelling solution for learning a model from decentralized and privacy-sensitive data. The main idea behind federated learning is that each platform trains a local model based on its own local data and a master server coordinates massive platforms to collaboratively train a global model by aggregating these local model updates.

Unfortunately, directly applying federated learning to the decentralized distantly supervised data fails, because conventional federated learning requires the local data to come with labels without noise (Tuor et al., 2020), however, in distant supervision, automatic labeling inevitably accompanies with **label noise** (Riedel et al., 2010; Hoffmann et al., 2011; Zeng et al., 2015; Lin et al., 2016),

which means not all sentences that mention an entity pair can represent the relation between them. Training on such noisy data will substantially hinder the performance of the RE model.

Moreover, even involving previous denoising methods, such as Zeng et al. (2015); Lin et al. (2016); Ye & Ling (2019), cannot handle label noise well in federated settings. This point can be illustrated by the example in Figure 1. S_1 and S_2 contain the same entity pair (*"Steve Jobs"*, *"Apple"*) but are distributed on two platforms. S_1 is true positive while S_2 is a false positive instance, which does not express the *"founder"* relation. In centralized training, there is no barrier between Platform 1 and Platform 2; therefore, simultaneously considering S_1 and S_2 can easily filter out noise via only selecting S_1 (Zeng et al., 2015) or placing a small weight on S_2 (Lin et al., 2016; Ye & Ling, 2019). However, raw data exchange between platforms is prohibited in federated settings. Due to the lack of comparison with S_1, previous denoising methods would mistakenly regard S_2 as a true positive instance. As a result, S_2 is retained and then poisons the local model in platform 2, which would affect the global model in turn.

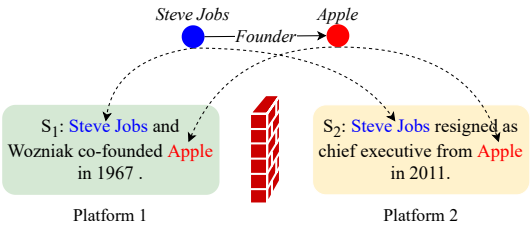

Figure 1: An example of the sentences that contain the same entity pair distributed on two platforms. The triple (*Steve Jobs*, *Founder*, *Apple*) is a fact in the KB

To suppress label noise in federated settings, we propose a federated denoising framework in this paper. The core of this framework is a multiple instance learning (MIL) (Dietterich et al., 1997; Maron & Lozano-Pérez, 1998) based denoising algorithm, called **Lazy MIL**, which is only executed at the beginning of each communication round and then would rest until the next round. Since the sentences containing the same entity pair scatter around different platforms, Lazy MIL algorithm coordinates multiple platforms to jointly select reliable sentences. Once sentences have been selected, they would be used repeatedly to train local models until the end of this round.

In summary, the contributions of this paper are:

- Considering data decentralization and privacy protection, we investigate distant supervision under the federated learning paradigm, which decouples the model training from the need for direct access to the raw data. To our best knowledge, combining federated learning with distant supervision is still an unexplored territory, which is the main focus of this paper.
- Since the automatic labeling in distant supervision inevitably accompanies with label noise, we present a multiple instance learning based denoising method, which can select reliable instances via cross-platform collaboration.
- The proposed method yields promising results on two benchmarks datasets, and we perform various experiments to verify the effectiveness of the proposed method. The code will be released at `http://anonymized`.

2 Related Work

In this section, we will briefly review the recent progress in distant supervision and some existing studies in federated learning.

Distant supervision. Relation extraction is a task of mining factual knowledge from free text by labeling relations between entity mentions. To alleviate the dependence of supervised methods on annotated data, Mintz et al. (2009) proposed distant supervision by using a knowledge base to annotate a large-scale dataset automatically. However, automatic labeling inevitably accompanies with label noise. To deal with label noise, most distantly supervised approaches (Riedel et al., 2010; Hoffmann et al., 2011; Surdeanu et al., 2012; Zeng et al., 2015; Lin et al., 2016; Luo et al., 2017; Ye & Ling, 2019; Yuan et al., 2019) focus on reducing label noise at bag [1] level prediction. These studies fall under multiple instance learning framework, which assumes that at least one sentence expresses the relation in a bag. Another line of work aims to reduce label noise at sentence level prediction.

[1] A set of sentences containing the same entity pair is called a "bag"

These studies (Zeng et al., 2018; Feng et al., 2018; Qin et al., 2018a;b) use reinforcement learning or adversarial training to select trustable relation labels by matching the predicted labels with distantly supervised labels. Compared with previous studies, our work focuses on reducing label noise at bag level prediction and extends distant supervision to federated settings.

Distant supervision has also been applied to other natural language processing tasks, such as named entity recognition (Ghaddar & Langlais, 2018; Shang et al., 2018; Nooralahzadeh et al., 2019), event extraction (Chen et al., 2017), sentiment classification (Go et al., 2009) and question answer (Joshi et al., 2017; Lin et al., 2018; Cheng et al., 2020).

Federated Learning. Recently, federated learning (McMahan et al., 2016; Konečnỳ et al., 2016a;b; Bonawitz et al., 2017; Smith et al., 2017; Caldas et al., 2018; Zhao et al., 2018; Li et al., 2018; Jeong et al., 2018; Peng et al., 2019; Li et al., 2019; Wang et al., 2020; Rothchild et al., 2020; Yu et al., 2020) has become a rapidly developing topic in the research community, since it provides a new communication-efficient way of learning a model over a collection of highly distributed platforms while still preserving data privacy. However, most of the previous studies require the data stored by the local platforms to come with ground-truth labels without noise. The problem of how to adapt federated learning to a noisy environment is relatively ignored. In terms of overcoming noise in federated settings, Tuor et al. (2020) is most relevant to our work but require a clean benchmark dataset to train a benchmark model. Compared with Tuor et al. (2020), our work does not rely on a clean benchmark dataset, which does not exist in distant supervision.

3 Federated Denoising Framework

3.1 Task Definition

In this paper, we focus on distant supervision in federated settings. Define K platforms $\{P_1, ...P_K\}$ with respective unlabeled corpora $\{D_1, ...D_K\}$. Under the assumption of centralized training, each platform transfers or shares its local corpus to a server, and the server will take the integrated corpus $D = D_1 \cup ... \cup D_K$ to conduct training, while the task of distant supervision in federated settings requires platform P_i does not expose its corpus D_i to others (including the server). In distant supervision, a KB is required to automatically label these corpora. In this paper, we only focus on the data security of these unlabeled corpora and assume the KB is publicly available for all platforms. The issue of protecting the security of KB is beyond the scope of the current work.

To solve this task, we propose a federated denoising framework. The key components of this framework will be elaborated in the following section. Concretely, we firstly introduce the basic relation extractor in Section 3.2, which is the network architecture shared by the global model and local models. Then, we present how to select reliable instances via cross-platform collaboration in Section 3.3. Next, we describe how to use the selected instances to train the local model in Section 3.4. Finally, we present how to use the FedAvg algorithm to update the global model in Section 3.5.

3.2 Relation Extractor

Following previous studies (Zeng et al., 2015), we adopt the Piecewise Convolutional Neural Network (PCNN) as our relation extractor. Given a sentence s and two entities within this sentence, we first split the sentence into tokens, and then each token w_i is mapped into a dense word embedding $\mathbf{e}_i \in \mathbb{R}^{d_w}$. To specify the entity pair, relative distances between the current token w_i and the two entities are transformed into two positional features by looking up the position embedding matrices. Next, the token is represented as the concatenation of the word embedding and two positional features, and is fed into a convolutional neural network. Then, piecewise max pooling (Zeng et al., 2015) is employed to extract the high-level sentence representation. In the piecewise max pooling, an input sentence is divided into three segments based on the two entities, and the maximum value of CNN outputs in each segment is returned. After that, we apply a single fully connected layer to output the logit value $\mathbf{o}$. Finally, the conditional probability of j-th relation is denoted as follows:

$$p(rel_j|s, \Theta) = \frac{exp(\mathbf{o}_j)}{\sum_{i=1}^{M} exp(\mathbf{o}_i)} \tag{1}$$

where Θ is the model parameter and M is the total number of relation.

Algorithm 1 Lazy Multiple Instance Learning

Input: global model parameters Θ, the set of activated platforms A.
Define two dictionary on the server, named V and I ▷ Run on the master server
Distribute Θ to each platform in A
for each platform $i \in A$ **in parallel do** ▷ Run on the activated platforms
  for each triple (h, r, t) in KB **do**
    for each sentence s_z^i in the bag b^i **do**
      Compute $p(r|s_z^i, \Theta)$ ▷ According to Equation 1
    $v^i, id^i \leftarrow \max_z(p(r|s_z^i, \Theta))$, $s_z^i \in b^i$ ▷ v^i is called uploaded value
    Upload $[v^i, id^i, i]$ to the server and append $[v^i, id^i, i]$ to $V[(h, r, t)]$
for each key (h, r, t) in V **do** ▷ Run on the master server
  Sort $V[(h, r, t)$ in descending order according to the uploaded values.
  $I[(h, r, t)] \leftarrow V[(h, r, t)][0]$
Broadcast I to each platform in A

3.3 Lazy Multiple Instance Learning

To avoid the local relation extractor being poisoned by false positive instances, we propose lazy multiple instance learning (Lazy MIL), which can select reliable instances via cross-platform collaboration. The overview of Lazy MIL is illustrated in Algorithm 1.

Suppose that there is a triple (h, r, t) in the public KB, the set of sentences containing the head entity h and tail entity t is represented as $\{(s_1^1, s_2^1, ..., s_{n_1}^1), ..., (s_1^K, s_2^K, ..., s_{n_k}^K)\}$, where s_i^j indicates the i-th instance in the platform j. In the q-th communication round, assume that only platform i and platform j are activated. At the beginning of this round, the parameters of the global model Θ_q are distributed to the activated platforms i and j for initializing local models, which ensures that all activated local models share the same parameters in Lazy MIL. In platform i, the sentences in the set $(s_1^i, s_2^i, ..., s_{n_i}^i)$ are fed into the local model to get conditional probabilities associated with the relation r according to Equation 1, where r is the predicate of the triple. The value v^i and index id^i of the instance with the maximum conditional probability associated with the relation r are computed as follows:

$$v^i, id^i = \max_z(p(r|s_z^i, \Theta_q)) \quad 1 \leq z \leq n_i \tag{2}$$

After computation, platform i uploads the value v^i and index id^i to the master server. At the same time, the same procedure is performed on platform j, and the value v^j and index id^j are also uploaded to the server.

The master server decides which local instance can be selected among all activated platforms based on the uploaded values. If $v^i > v^j$, then the id^i-th sentence in platform i is selected as the reliable sentence that expresses the triple (h, r, t) in this round. This decision, called denoising information, is broadcast to all activated platforms. Each activated platform selects reliable training instances from its local corpus according to this denoising information. Note that since only values and indices of conditional probabilities are uploaded to the master server, Lazy MIL almost does not leak the corpus information in each platform.

3.4 Local Model Training

After platform i selects reliable instances from its local corpus D_i, the selected reliable instance set $D_i^\star$ is used for training the local relation extractor. We use the cross-entropy loss function to optimize parameters Θ_q, which is defined as follows:

$$J(\Theta_q; D_i^\star) = -\frac{1}{|D_i^\star|} \sum_{u=1}^{|D_i^\star|} \log p(r_u|s_u^\star, \Theta_q) \tag{3}$$

where $s_u^\star$ indicates the u-th sentence in the selected reliable instance set $D_i^\star$. After training E epochs on the selected reliable instance set, the trained parameters Θ_{q+1}^i are uploaded to the master server, where the superscript i indicates the parameters are trained on platform i.

Algorithm 2 Federated Denoising Framework

Hyperparameters: K is the total number of platforms; C is the fraction of platforms; B is the local minibatch size; E is the number local epochs; η is the learning rate.
Master server executes:
  Initialize Θ_0
  for communication round q = 0,1,... **do**
    $m \leftarrow \max(C \times K, 1)$ ▷ Select activated platforms
    $A_q \leftarrow$ (random set of m platforms)
    Execute lazy multiple instance learning algorithm ▷ Defined in Algorithm 1
    for each platform $i \in A_q$ **in parallel do**
      $\Theta^k_{q+1} \leftarrow$ Local_Training(i, Θ_q)
    $\Theta_{q+1} \leftarrow \sum_{i \in A_q} \frac{|D^\star_i|}{\sum_{j \in A_q} |D^\star_j|} \Theta^i_{q+1}$ ▷ Defined in Equation 5

Function Local_Training(i, Θ): ▷ Run on platform i
  Generate denoised dataset $D^\star_i$ from D_i based on the denoising information I
  $\mathcal{B} \leftarrow$ (split $D^\star_i$ into batches of size B)
  for each local epoch e from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      $\Theta \leftarrow \Theta - \eta \nabla J(\Theta; b)$ ▷ J is defined in Equation 3
  return Θ to the master server

3.5 Global Model Update

Suppose A_q is the set of activated platforms in the q-th communication round. After all activated platforms finish local training, the master server collects all trained parameters $\{\Theta^i_{q+1} | i \in A_q\}$ to update the global model. We define the goal of the global model as follows:

$$\min_{\Theta_q} \sum_{i \in A_q} \frac{|D^\star_i|}{\sum\limits_{j \in A_q} |D^\star_j|} J(\Theta_q; D^\star_i) \tag{4}$$

where $J(\Theta_q; D^\star_i)$ is the local loss function for the platform i. Follow previous studies (McMahan et al., 2016), we optimize this global objective function via taking the weighted average of all trained parameters, which is shown as follows:

$$\Theta_{q+1} = \sum_{i \in A_q} \frac{|D^\star_i|}{\sum\limits_{j \in A_q} |D^\star_j|} \Theta^i_{q+1} \tag{5}$$

where Θ^i_{q+1} is the optimal parameters obtained by minimizing the local loss function on the local data of platform i. Since all trained parameters from different platforms are aggregated together, the corpus information of each platform is hard to be inferred. Thus, corpora in platforms are well-protected. Complete pseudo-code of this framework is given in Algorithm 2.

4 Experiments

4.1 Datasets and Evaluation Metrics

Since conducting experiments on non-public privacy-sensitive datasets is not reproducible, we choose public distantly supervised relation extraction datasets to investigate the effectiveness of the proposed framework.

NYT 10[2] (Riedel et al., 2010) is a widely used dataset in distant supervision. It was automatically generated by aligning the semantic triples in Freebase with the New York Times corpus. The training set contains 466,876 sentences, 251,928 entity pairs and 16,444 relational facts. The development set contains 55167 sentences, 28077 entity pairs and 1,808 relational facts. The test set contains

[2] https://github.com/thunlp/OpenNRE

172,448 sentences, 96,678 entity pairs and 1,950 relational facts. There are 52 actual relations and a special relation NA for representing no relation between two entities.

MIRGENE[3] (Li et al., 2017) is a large-scale biomedical dataset. This dataset is generated by aligning Tarbase and miRTarBase with the abstracts in Medline. An example is shown in the following: " ***MicroRNA-223*** *regulates* ***FOXO1*** *expression and cell proliferation*", where MicroRNA-223 is a miRNA and FOXO1 is a gene. There are 172727 sentences in the training set and 1239 sentences in the test set. We randomly sampled 10% of the data from the training set as the development set.

Data Partitioning. To study distant supervision in federated settings, we need to specify how to distribute the data across platforms. In this paper, we focus on the IID situation, where the training data are shuffled and then partitioned into K (the total number of platforms) platforms.

Evaluation Metrics. We evaluate our approach and baseline methods on the held-out test set of these two datasets. Precision-recall (PR) curves, area under curve (AUC) values and Precision@N (P@N) values are adopted as evaluation metrics in our experiments.

4.2 Experimental Settings

For a fair comparison, we implement our method and all baselines in the same experimental settings. We divide the hyperparameters into three parts, i.e., fixed hyperparameters, unfixed hyperparameters and federated hyperparameters. Fixed hyperparameters follow the hyperparameter settings in Lin et al. (2016), including the 50-dimensional pretrained word embeddings for NYT, the 5-dimensional position embeddings, and CNN module that includes 230 filters with a window size of 3. For MIRGENE, 200-dimensional word embeddings pretrained on PubMed and MIMIC-III are used. The optimal unfixed hyperparameters are determined by grid search based on the performance of the development set, and the search space of unfixed hyperparameters is shown in Table 1. Federated hyperparameters include the total number of platforms K, the fraction of platforms C, the local minibatch size B, the number of local epochs E. All of these control the amount of computation. In the end-to-end comparison, we fix the K to 100, B to 32, E to 3, and set the hyperparameter space of C as $\{0.1, 0.2, 0.5, 1\}$ following McMahan et al. (2016). We use stochastic gradient descent as the local training optimizer and all experiments can be done by using a single GeForce GTX 1080 Ti.

Hyperparameter	Search Space
Learning Rate (η)	0.05, 0.08, 0.1,0.2
Learning Rate Decay	0.01, 0.05
Dropout	0.1, 0.2, 0.5
Weight Decay	1e-5, 1e-6

Table 1: The search space of unfixed hyperparameter.

4.3 Baselines

We compare our method with the following baselines in federated settings: (1) Directly applying FedAvg algorithm (McMahan et al., 2016) to the automatically labeled data is the first baseline, which is called **NONE**. In this case, there is no denoising module in this method. (2) Zeng et al. (2015) proposed to leverage multiple instance learning to choose the most reliable sentence as the bag representation, and we abbreviate this method as **ONE**; (3) **ATT** was proposed by Lin et al. (2016), which uses the attention mechanism to select reliable instances by placing soft weights on a set of noisy sentences; (4) **AVE** (Lin et al., 2016) is a naive version of ATT and represents each sentence set as the average vector of sentences inside the set; (5) **ATT_RA** (Ye & Ling, 2019) is a variant of ATT, which calculates the bag representations in a relation-aware way. The federated framework of these baselines is shown in Algorithm 3 in the appendix.

4.4 Main Results

Figure 2 and Figure 3 show the precision-recall curves on NYT dataset and MIRGENE datasets. At the appendix, we also present AUC values of these curves in Table 8 and detailed precision values measured at different points along these curves in Table 9. To reduce randomness, we run 10

[3] https://github.com/leebird/bionlp17

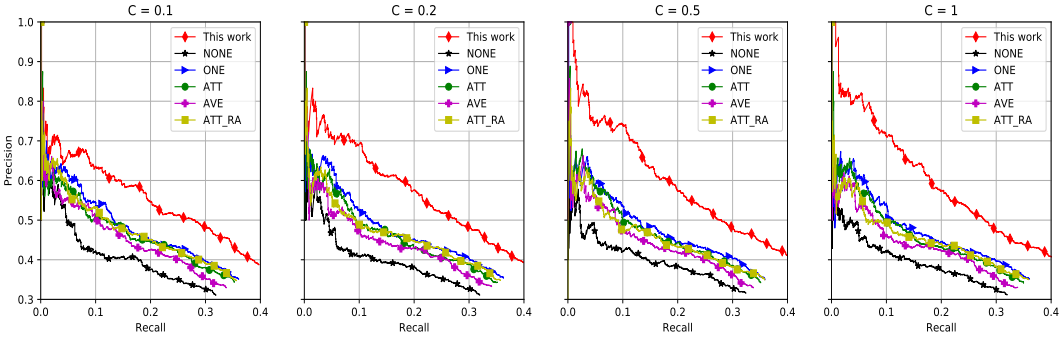

Figure 2: Aggregate precision-recall curves on NYT 10 dataset, where C is the fraction of platforms that are activated on each round.

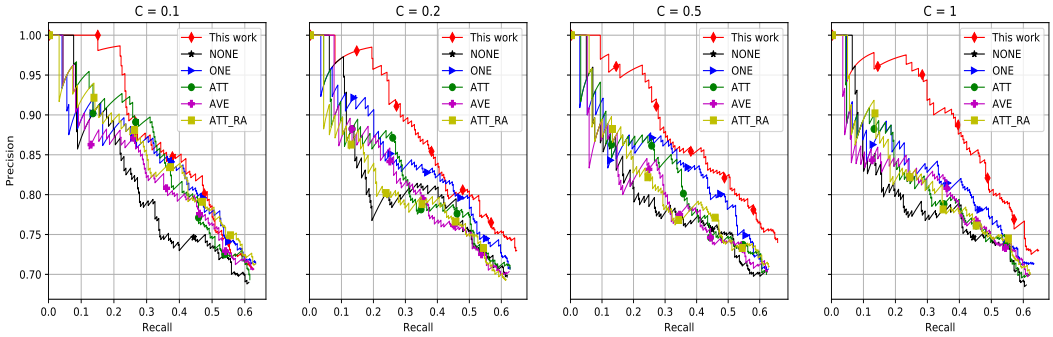

Figure 3: Aggregate precision-recall curves on MIRGENE dataset, where C is the fraction of platforms that are activated on each round.

AUC	NONE	ONE	ATT	AVE	ATT_RA	Ours
C=0.1	0.1287±0.0034	0.1719±0.0030	0.1638±0.0030	0.1521±0.0029	0.1664 ±0.0026	**0.2189±0.0025**
C=0.2	0.1255±0.0032	0.1710±0.0029	0.1630±0.0028	0.1517±0.0027	0.1642±0.0022	**0.2285±0.0023**
C=0.5	0.1239±0.0045	0.1701±0.0020	0.1619±0.0025	0.1513±0.0024	0.1630±0.0020	**0.2420±0.0021**
C=1.0	0.1223±0.0037	0.1689±0.0021	0.1604±0.0022	0.1491±0.0015	0.1625±0.0022	**0.2447±0.0019**

Table 2: AUC values on NYT 10 dataset. We run 10 models using random seeds with early stopping on the development set, and report the mean and standard deviation of test AUC values for all methods.

AUC	NONE	ONE	ATT	AVE	ATT_RA	Ours
C=0.1	0.7316± 0.0069	0.7665±0.0087	0.7535± 0.0062	0.7499±0.0055	0.7514± 0.0053	**0.7846±0.0066**
C=0.2	0.7246±0.0047	0.7610±0.0092	0.7472±0.0055	0.7428± 0.0052	0.7431±0.0071	**0.7897±0.0059**
C=0.5	0.7251±0.0054	0.7605±0.0065	0.7453±0.0058	0.7409±0.0062	0.7423 ±0.0079	**0.7915±0.0065**
C=1.0	0.7229± 0.0059	0.7559±0.0080	0.7424 ±0.0067	0.7368±0.0063	0.7395±0.0072	**0.7942±0.0060**

Table 3: AUC values on MIRGENE dataset. We run 10 models using random seeds with early stopping on the development set, and report the mean and standard deviation of test AUC values for all methods.

models using random seeds with early stopping on the development set. Table 2 and Table 3 show the mean and standard deviation test AUC values for each method on NYT 10 dataset and MIRGENE dataset, respectively. We find that: (1) Our method significantly outperforms all baselines in federated settings. We believe the reason is that our denoising method can use cross-platform information to hinder false positive instances from poisoning local models, which leads to a better performance of the global model. (2) Directly applying FedAvg algorithm (McMahan et al., 2016) to the automatically labeled data achieve the worst results in both datasets. The reason behind that

is training on the noisy data will substantially hinder the performance of the model. Therefore, it is necessary to conduct denoise in federated distant supervision. (3) C is the fraction of platforms that are activated on each round, which controls the amount of multi-platform parallelism. With increasing platform parallelism, the performance of all baselines declines slightly while our method performs better. Intuitively, increasing platform parallelism is able to lead to better results, since involving more platforms in training can increase the likelihood that all sentences with the same entity pair appear simultaneously. However, due to lack of cross-platform collaboration, all baselines handle label noise only based on its own local data, which may hamper the performance. In contrast, our method selects reliable instances among all activated platforms, which can effectively reap the benefits of increasing platform parallelism. (4) Leveraging attention mechanisms to denoise, an effective solution in centralized settings, seems not to work in federated settings. Compared with centralized training, the sentences in a bag scatter around different platforms in federated settings, so the number of the sentences with the same entity pair on a platform is small, which may lead to placing large attention weights on noisy sentences due to lack of inter-bag contrast.

4.5 Increasing the Size of Local Datasets

In this section, we increase the size of local datasets by setting K to 50. In such a way, each local dataset is twice as large as it was (when K is set to 100). For a fair comparison, we fix $C = 0.1$, $B = 32$ and $E = 3$. Figure 4 show the results of AUC values. At the appendix, we also present corresponding precision-recall curves in Figure 7 and detailed precision values measured at different points along these curves in Table 10. From the results, we observe that: (1) Our proposed method significantly surpasses all baselines in both datasets. (2) Compared with setting K to 100, the result of directly applying FedAvg algorithm (McMahan et al., 2016) to the automatically labeled data remains almost unchanged when K is set to 50. (3) As the size of local datasets increases, all denoising methods can achieve better results. The most likely reason is that compared with setting K to 100, setting K to 50 increases the probability that all sentences with the same entity pairs simultaneously exist in the same platform.

AUC	NYT	MIRGENE
NONE	0.1325	0.7430
ONE	0.1856	0.7786
ATT	0.1806	0.7726
AVE	0.1687	0.7592
ATT_RA	0.1842	0.7639
Ours	**0.2285**	**0.7941**

Table 4: AUC values on NYT 10 dataset and MIRGENE dataset when $K = 50$.

4.6 Increasing the Number of Local Updates

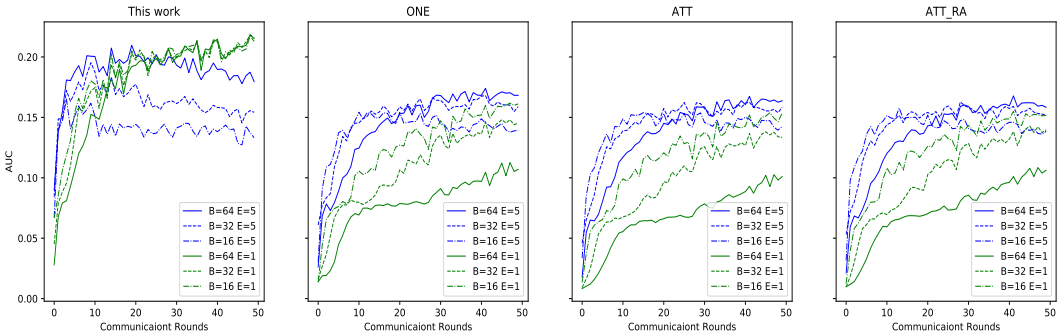

Figure 4: AUC values vs. communication rounds on NYT data with different E (the number of local epochs) and B (the local minibatch size).

In this section, we investigate the impact of varying the number of local updates in this section. The number of local updates is given by $E\frac{|D_i^*|}{B}$, where $|D_i^*|$ is the size of the denoised dataset in platform i at a round, B is the local minibatch size and E is the number of local epochs. Increasing B, decreasing E, or both will reduce computation on each round. We fix C to 0.1 and only B and

E are varied in this section [4]. The results are shown in Figure 4. We find that: (1) Compared with the other denoising baselines, our method converges faster to the optimal results. We conjecture that is due to that the proposed denoising method can effectively filter out the noise, which makes the relation extractor less affected by false positive instances and converge faster. (2) When setting B to 64 and E to 1, our method achieves the best AUC value. (3) Increasing the local minibatch B may improve extraction performance. (4) Increasing the local epoch E can speed up converge, but may not make the global model converge to a higher level of AUC value. These findings are in line with McMahan et al. (2016), which shows it may hurt performance when we over-optimize on the local dataset.

5 Conclusion

Considering data decentralization and privacy protection, we investigate distant supervision under the federated learning paradigm, which permits learning to be done while data stays in its local environment. To suppress label noise in federated settings, we propose a federated denoising framework, which can select reliable instances via cross-platform collaboration. Extensive experiments on two datasets have demonstrated the effectiveness of our method.

Distant supervision in federated settings is far from being solved and this work is just the beginning. There are still many problems need to be solved, such as noisy bag problem (Xu et al., 2013; Liu et al., 2017) and shifted label problem (Ye et al., 2019). Noisy bag problem means that all sentences containing the same entity pair are incorrectly labeled, and shifted label problem means the label distribution of training set does not align with that of test set. In federated settings, how these problems affect the relation extractor is still unknown. In our future work, we will devote to solve these problems in federated settings.

References

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017.

Sebastian Caldas, Jakub Konečnỳ, H Brendan McMahan, and Ameet Talwalkar. Expanding the reach of federated learning by reducing client resource requirements. *arXiv preprint arXiv:1812.07210*, 2018.

M. A. P. Chamikara, P. Bertok, I. Khalil, D. Liu, and S. Camtepe. Privacy preserving distributed machine learning with federated learning. 2020.

Yubo Chen, Shulin Liu, Xiang Zhang, Kang Liu, and Jun Zhao. Automatically labeled data generation for large scale event extraction. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Hao Cheng, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Probabilistic assumptions matter: Improved models for distantly-supervised document-level question answering. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 1997.

Jun Feng, Minlie Huang, Li Zhao, Yang Yang, and Xiaoyan Zhu. Reinforcement learning for relation classification from noisy data. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[4] The lr, lr decay, weight decay and dropout are fix to is 0.1, 0.01, 1e-5 and 0.1 respectively, which are not the optimal hyperparameters for most experiments

Abbas Ghaddar and Phillippe Langlais. Robust lexical features for improved neural network named-entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, August 2018.

Alec Go, Richa Bhayani, and Lei Huang. Twitter sentiment classification using distant supervision. *CS224N Project Report, Stanford*, 2009.

Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013.

Raphael Hoffmann, Congle Zhang, Xiao Ling, Luke Zettlemoyer, and Daniel S Weld. 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*, 2011.

Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *arXiv preprint arXiv:1811.11479*, 2018.

Mandar Joshi, Eunsol Choi, Daniel Weld, and Luke Zettlemoyer. TriviaQA: A large scale distantly supervised challenge dataset for reading comprehension. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Jakub Konečnỳ, H Brendan McMahan, Daniel Ramage, and Peter Richtárik. Federated optimization: Distributed machine learning for on-device intelligence. *arXiv preprint arXiv:1610.02527*, 2016a.

Jakub Konečnỳ, H Brendan McMahan, Felix X Yu, Peter Richtárik, Ananda Theertha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. *arXiv preprint arXiv:1610.05492*, 2016b.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. BioBERT: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 2019.

Gang Li, Cathy Wu, and K. Vijay-Shanker. Noise reduction methods for distantly supervised biomedical relation extraction. In *BioNLP 2017*, 2017.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. *arXiv preprint arXiv:1812.06127*, 2018.

Tian Li, Maziar Sanjabi, Ahmad Beirami, and Virginia Smith. Fair resource allocation in federated learning. *arXiv preprint arXiv:1905.10497*, 2019.

Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. Denoising distantly supervised open-domain question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018.

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

Bingfeng Luo, Yansong Feng, Zheng Wang, Zhanxing Zhu, Songfang Huang, Rui Yan, and Dongyan Zhao. Learning with noise: Enhance distantly supervised relation extraction with dynamic transition matrix. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics*, 2017.

Oded Maron and Tomás Lozano-Pérez. A framework for multiple-instance learning. In *Advances in neural information processing systems*, 1998.

Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pp. 109–165. Elsevier, 1989.

H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. *arXiv preprint arXiv:1602.05629*, 2016.

Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. 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*, 2009.

Farhad Nooralahzadeh, Jan Tore Lønning, and Lilja Øvrelid. Reinforcement-based denoising of distantly supervised NER with partial annotation. In *Proceedings of the 2nd Workshop on Deep Learning Approaches for Low-Resource NLP (DeepLo 2019)*, 2019.

Xingchao Peng, Zijun Huang, Yizhe Zhu, and Kate Saenko. Federated adversarial domain adaptation. *arXiv preprint arXiv:1911.02054*, 2019.

Pengda Qin, Weiran Xu, and William Yang Wang. Dsgan: Generative adversarial training for distant supervision relation extraction. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018a.

Pengda Qin, Weiran Xu, and William Yang Wang. Robust distant supervision relation extraction via deep reinforcement learning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, 2018b.

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

Daniel Rothchild, Ashwinee Panda, Enayat Ullah, Nikita Ivkin, Ion Stoica, Vladimir Braverman, Joseph Gonzalez, and Raman Arora. Fetchsgd: Communication-efficient federated learning with sketching. *arXiv preprint arXiv:2007.07682*, 2020.

Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1715–1725, 2016.

Jingbo Shang, Liyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018.

Virginia Smith, Chao-Kai Chiang, Maziar Sanjabi, and Ameet S Talwalkar. Federated multi-task learning. In *Advances in Neural Information Processing Systems*, pp. 4424–4434, 2017.

Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2895–2905, 2019.

Mihai Surdeanu, Julie Tibshirani, Ramesh Nallapati, and Christopher D. Manning. 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*, 2012.

Tiffany Tuor, Shiqiang Wang, Bong Jun Ko, Changchang Liu, and Kin K Leung. Data selection for federated learning with relevant and irrelevant data at clients. *arXiv preprint arXiv:2001.08300*, 2020.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, 2017.

Hongyi Wang, Mikhail Yurochkin, Yuekai Sun, Dimitris Papailiopoulos, and Yasaman Khazaeni. Federated learning with matched averaging. *arXiv preprint arXiv:2002.06440*, 2020.

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

Wei Xu, Raphael Hoffmann, Le Zhao, and Ralph Grishman. Filling knowledge base gaps for distant supervision of relation extraction. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, 2013.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2019.

Qinyuan Ye, Liyuan Liu, Maosen Zhang, and Xiang Ren. Looking beyond label noise: Shifted label distribution matters in 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)*, 2019.

Zhi-Xiu Ye and Zhen-Hua Ling. 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*, 2019.

Felix X Yu, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Federated learning with only positive labels. *arXiv preprint arXiv:2004.10342*, 2020.

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

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

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

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

Fadila Zerka, Samir Barakat, Sean Walsh, Marta Bogowicz, Ralph TH Leijenaar, Arthur Jochems, Benjamin Miraglio, David Townend, and Philippe Lambin. Systematic review of privacy-preserving distributed machine learning from federated databases in health care. *JCO Clinical Cancer Informatics*, 2020.

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

Yue Zhao, Meng Li, Liangzhen Lai, Naveen Suda, Damon Civin, and Vikas Chandra. Federated learning with non-iid data. *arXiv preprint arXiv:1806.00582*, 2018.

Peng Zhou, Wei Shi, Jun Tian, Zhenyu Qi, Bingchen Li, Hongwei Hao, and Bo Xu. Attention-based bidirectional long short-term memory networks for relation classification. In *Proceedings of the 54th annual meeting of the association for computational linguistics*, 2016.

A CASE STUDIES

Platform	Sentence	Type	NONE	ONE	ATT	ATT_RA	This Work
10	Most Muslims in Montenegro, mindful of the Serbs' killings of Muslims in Bosnia, are expected to vote to end ties with Serbia, but villagers in Podgorica are worried about how their Serb neighbors would react to separation.	Fasle Positive	✓	✓	✓	✓	✗
7	They have passed through Zagreb; Novi Sad; Belgrade; Pristina, in Kosovo; Skopje; Tirana, Albania; and Podgorica, Montenegro, on their way to Sarajevo.	False Positive	✓	✓	✓	✓	✗
56	This is a great day for the citizens of Montenegro to regain independence after 88 years, "said Ljubomir Djurkovic, a theater director from Centinje, a picturesque, pro-independence town to the west of Podgorica.	False Positive	✓	✓	✓	✓	✗
26	The time has come, " Montenegro's prime minister, Milo Djukanovic, said Thursday at a jubilant final rally in Podgorica, the capital.	True Positive	✓	✓	✓	✓	✓

Table 5: A case to illustrate the effectiveness of the proposed model. A fact in KB is (*Podgorica*, /location/country/capital, *Montenegro*). Only the sentence in Platform 26 expresses the "/location/country/capital" relation, while the other sentences are all false positive.

Table 5 shows how different denoising methods select reliable instances in the training phase. In this case, a KB fact is (*Podgorica*, /location/country/capital, *Montenegro*). Aligning this KB fact with decentralized raw text generates four training instances, which are distributed in four different platforms. Only the sentence in Platform 26 correctly represents the "/location/country/capital" relation. The other sentences distributed in the other platforms are all false positive instances, which do not express the "/location/country/capital" relation.

From this case, we can find that: (1) If FedAvg algorithm (McMahan et al., 2016) was directly applied to the automatically labeled data, it would face a noisy environment where most sentences are false positive. (2) Previous denoising methods, such as ONE (Zeng et al., 2015), ATT (Lin et al., 2016) and ATT_RA (Ye & Ling, 2019), all fail to filter out false positive instances. In the worst cases, these methods will lose their denoising function. (3) Our proposed method can remove all false positive instances and only keep the true positive instance to train local models.

B ADDITIONAL EXPERIMENTS

B.1 CAN A STRONG EXTRACTOR MITIGATE THE LABEL NOISE IN FEDERATED SETTINGS?

In this section, we investigate the impact of involving a stronger extractor. More concretely, we replace the PCNN-based extractor with a BERT-based extractor (Devlin et al., 2018). In the BERT-based extractor, we use the architecture of entity mention pooling (Soares et al., 2019) to represent relations with the Transformer model (Vaswani et al., 2017), which is shown in Figure 5. Given a sentence s and two entities within this sentence, we first segment the given sentence into tokens by the byte pair encoding (Sennrich et al., 2016) and feed these tokens into the BERT encoder. The output of the BERT encoder is the context-aware embeddings of tokens. After that, we use max pooling on the context-aware embeddings that correspond to the word pieces in each entity mention, to get two vectors h_{e1} and h_{e2} representing the two entity mentions. Finally, We concatenate these two vectors to get the representation of relation.

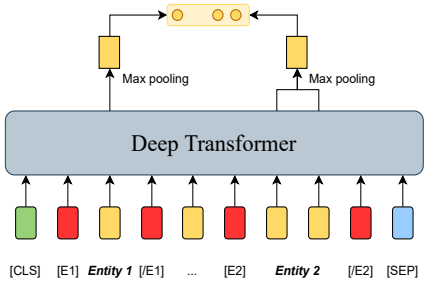

Figure 5: The main architecture for BERT-based extractor.

For a fair comparison, we fix $C = 0.1$, $B = 32$, $K = 100$ and $E = 3$. For the BERT-based extractor, we set the lr, lr decay and weight decay to 1e-5, 0.01 and 1e-5, and we use the pretrained BioBERT (Lee et al., 2019) and cased base version of BERT [5] as the initialization parameters in MIRGENE and NYT 10 dataset, respectively. The AUC values of PCNN-based extractor and BERT-based extractor on NYT 10 dataset and MIRGENE dataset are shown in Table 5. From the result, we find that: (1) Involving a stronger encoder can improve the performance for all denoising method. (2) Whether leveraging PCNN or BERT as the encoder, our method significantly outperforms all baselines.

Dataset	Extractor	NONE	ONE	ATT	AVE	ATT_RA	Ours
NYT 10	BERT-based Extractor	0.1744	0.2217	0.2156	0.2120	0.2086	**0.2526**
	PCNN-based Extractor	0.1287	0.1719	0.1638	0.1521	0.1664	**0.2189**
MIRGENE	BERT-based Extractor	0.7510	0.7773	0.7798	0.7650	0.7768	**0.8103**
	PCNN-based Extractor	0.7316	0.7665	0.75335	0.7499	0.7514	**0.7846**

Table 6: The AUC values of PCNN-based extractor and BERT-based extractor on NYT 10 dataset and MIRGENE dataset when k is set to 100 and C is set to 0.1.

B.2 ABLATION STUDY OF PARALLEL COMPUTATION

In this section, we conduct an ablation study of parallel computation in federated settings. Conventional federated learning approach (McMahan et al., 2016) is based on a master-slave topology [6]. In this topology, each platform (slave) trains a local model based on its own local data and a master server (master) coordinates massive platforms to collaboratively train a global model by aggregating these local model updates.

[5] https://github.com/google-research/bert
[6] https://en.wikipedia.org/wiki/Master/slave_(technology)

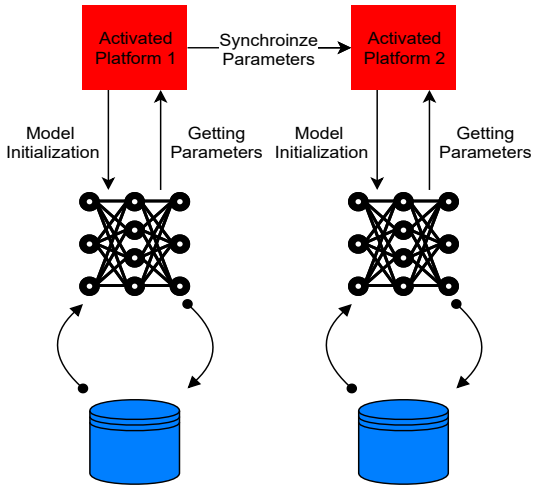

Figure 6: The main architecture for chain training.

To ablate parallel computation from the conventional federated learning topology, we propose a simple way, called chain learning, to handle decentralized data. In chain learning, we only train one local platform at a time, and then synchronize model parameters to the next platform for further training. We show the architecture of chain learning in Figure 6. Compared with federated learning, chain learning does not require a master server and parallel training.

	ONE	ATT	AVE	ATT_RA	Ours
Centralized Learning	0.2323	0.2345	0.2147	**0.2365**	-
Federated Learning	0.1719	0.1638	0.1521	0.1664	**0.2189**
Chain Learning	0.1574	0.1538	0.1466	**0.1590**	-

Table 7: AUC values of different learning paradigms on NYT 10 dataset. In federated learning and chain learning, we fix C = 0.1, K =100 and E =3.

Table B.2 shows the AUC values of centralized learning, federated learning and chain learning on NYT 10 dataset. For a fair comparison, we set batch size B to 32, set learning rate to 0.1, and set weight decay to 1e-5 for all learning paradigms. From the result, we find that: (1) Centralized learning achieves the best result compared to federated learning and chain learning; (2) Compared to federated learning, chain learning cannot achieve satisfactory results. We conjecture that catastrophic forgetting (McCloskey & Cohen, 1989; Goodfellow et al., 2013) may be the cause. In federated learning, averaging step for model integration is carried out in each epoch, which may ensures the generalization of the model. However, in chain learning, the model may forget the previous local data and overfit to the latest local data, which hampers the performance. (3) Our approach narrows the gap between federated training and centralized training in terms of performance. Therefore, we can conclude that not chain learning but cross-platform collaboration is the key to mitigate label noise in federated settings.

C BASELINES

Algorithm 3 Federated Denoising Baseline

Hyperparameters: K is the total number of platforms; C is the fraction of platforms; B is the local minibatch size; E is the number local epochs; η is the learning rate.
Master server executes:
  Initialize Θ_0
  for communication round q = 0,1,... **do**
    $m \leftarrow \max(C \times K, 1)$ ▷ Select activated platforms
    $A_q \leftarrow$ (random set of m platforms)
    for each platform $i \in A_q$ **in parallel do**
      $\Theta^k_{q+1} \leftarrow$ Local_Training(i, Θ_q)
    $\Theta_{q+1} \leftarrow \sum_{i \in A_q} \frac{|D_i|}{\sum_{j \in A_q} |D_j|} \Theta^i_{q+1}$ ▷ Defined in Equation 5

Function Local_Training(i, Θ): ▷ Run on platform i
  $\mathcal{B} \leftarrow$ (split D_i into batches of size B) ▷ A batch is a set of bag
  for each local epoch e from 1 to E **do**
    for batch $b \in \mathcal{B}$ **do**
      Conduct the denoising method ▷ In **NONE**, we do not carry out this step
      Update Θ based on the gradients of the loss function
  return Θ to the master server

In Algorithm 3, we present the federated framework of denoising baseline. Compared with FedAvg algorithm (McMahan et al., 2016), we only add one step in local training to denoise. Compared with the proposed federated denoising framework, local platforms in the baseline framework handle label noise only based on its own local data.

D SOME TABLES AND FIGURES MENTIONED IN THE MAIN TEXT

AUC	NYT						MIRGENE					
	NONE	ONE	ATT	AVE	ATT_RA	Ours	NONE	ONE	ATT	AVE	ATT_RA	Ours
C=0.1	0.1318	0.1747	0.1658	0.1544	0.1695	**0.2207**	0.7436	0.7705	0.7696	0.7577	0.7597	**0.7893**
C=0.2	0.1293	0.1747	0.1642	0.1531	0.1666	**0.2315**	0.7432	0.7649	0.7528	0.7491	0.7448	**0.7923**
C=0.5	0.1305	0.1725	0.1657	0.1527	0.1647	**0.2448**	0.7365	0.7626	0.7516	0.7431	0.7484	**0.7946**
C=1.0	0.1283	0.1715	0.1631	0.1503	0.1637	**0.2465**	0.7358	0.7570	0.7483	0.7432	0.7493	**0.7966**

Table 8: AUC values on NYT 10 dataset and MIRGENE dataset.

P@N(%)		NYT						MIRGENE					
		NONE	ONE	ATT	AVE	ATT_RA	Ours	NONE	ONE	ATT	AVE	ATT_RA	Ours
C=0.1	p@100	57.0	63.0	60.0	57.0	62.0	**69.0**	83.0	87.0	**89.0**	87.0	86.0	**89.0**
	P@200	49.0	60.0	57.0	55.0	55.5	**67.0**	75.0	79.5	77.5	78.0	77.0	**80.0**
	P@300	44.7	54.7	52.7	53.0	53.3	**63.0**	69.0	**71.3**	69.3	70.7	**71.3**	70.7
	Mean	50.2	59.2	56.6	55.0	56.9	**66.3**	75.7	79.3	78.6	78.6	78.1	**79.9**
C=0.2	p@100	56.0	66.0	59.0	59.0	61.0	**74.0**	80.0	85.0	87.0	85.0	80.0	**91**
	P@200	46.5	58.5	57.0	51.5	54.0	**70.5**	78.0	79.5	78.0	76.0	76.5	**80.5**
	P@300	42.3	55.0	52.7	50.7	51.0	**68.7**	69.7	70.7	70.7	70.3	69.3	**73.0**
	Mean	48.3	59.8	56.2	53.7	55.3	**71.1**	75.9	78.4‘	78.6	77.1	75.3	**81.5**
C=0.5	p@100	47	65.0	63.0	58.0	60.0	**77.0**	79.0	87.0	87.0	84.0	83.0	**92.0**
	P@200	47	59.0	57.5	53.5	54.5	**74.5**	75.5	80.0	75.0	75.0	77.0	**82.5**
	P@300	44.3	55.0	53.3	52.7	50.3	**71.7**	70.3	70.7	70.0	70.3	71.0	**74.0**
	Mean	46.1	59.7	57.9	54.7	54.9	**74.4**	74.9	79.2	77.3	76.4	77.0	**82.8**
C=1.0	p@100	48.0	62.0	65.0	60.0	60.0	**80.0**	78.0	82.0	82.0	82.0	83.0	**95.0**
	P@200	47.5	60.0	56.5	54.0	54.5	**75.5**	75.0	78.5	76.0	77.0	76.0	**82.0**
	P@300	43.3	56.0	52.3	49.7	49.0	**71.3**	68.7	71.3	70.0	70.0	70.0	**73.0**
	Mean	46.3	59.3	57.9	54.6	54.5	**75.6**	73.9	77.3	76.0	76.3	76.3	**83.3**

Table 9: P@100, P@200, P@300 and the mean of them for each model in held-out evaluation on NYT 10 dataset and MIRGENE dataset.

P@N(%)	NYT						MIRGENE					
	NONE	ONE	ATT	AVE	ATT_RA	Ours	NONE	ONE	ATT	AVE	ATT_RA	Ours
P@100	53.0	63.0	65.0	63.0	69.0	**73.0**	82.0	90.0	88.0	84.0	85.0	**94.0**
P@200	46.0	62.0	58.0	59.5	61.0	**69.5**	74.0	80.5	78.0	77.5	80.5	**83.0**
P@300	45.0	59.3	54.7	56.7	59.0	**68.7**	69.7	**71.7**	70.7	70.7	**71.7**	71.0
Mean	48.0	61.4	59.2	59.7	63.0	**70.4**	75.2	80.7	78.9	77.4	78.6	**82.7**

Table 10: P@100, P@200, P@300 and the mean of them for each model in held-out evaluation on NYT 10 dataset and MIRGENE dataset when K is set to 50 and C is set to 0.1.

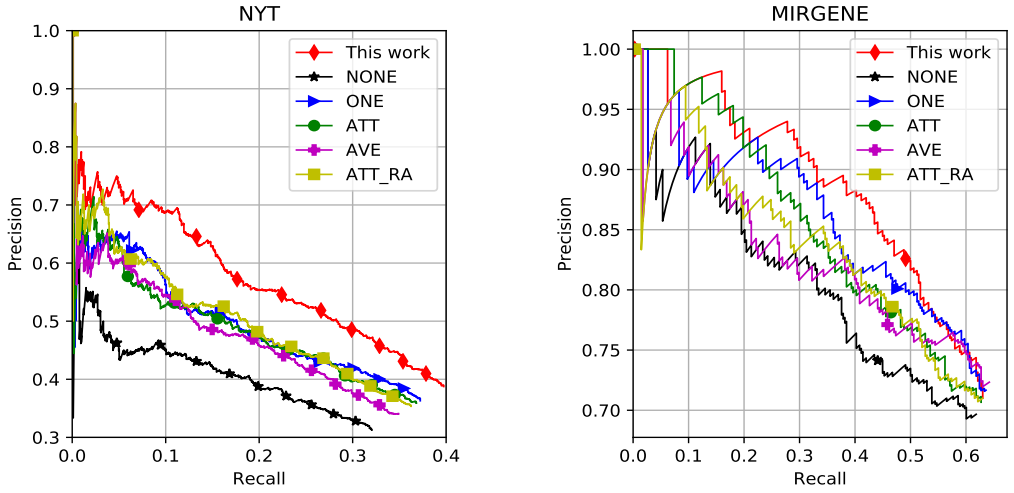

Figure 7: Aggregate precision-recall curves on NYT 10 dataset and MIRGENE dataset when K is set to 50 and C is set to 0.1.