# FedED: Federated Learning via Ensemble Distillation for Medical Relation Extraction

**Dianbo Sui**[♡♠], **Yubo Chen**[♡♠], **Jun Zhao**[♡♠], **Yantao Jia**[♢], **Yuantao Xie**[♢], **Weijian Sun**[♢]
[♡] National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing, China
[♠] School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, China
[♢] Huawei Technologies Co., Ltd., Beijing, China
{dianbo.sui, yubo.chen, jzhao}@nlpr.ia.ac.cn,
jamaths.h@163.com {xieyuantao2, sunweijian}@huawei.com

## Abstract

Unlike other domains, medical texts are inevitably accompanied by private information, so sharing or copying these texts is strictly restricted. However, training a medical relation extraction model requires collecting these privacy-sensitive texts and storing them on one machine, which comes in conflict with privacy protection. In this paper, we propose a privacy-preserving medical relation extraction model based on federated learning, which enables training a central model with no single piece of private local data being shared or exchanged. Though federated learning has distinct advantages in privacy protection, it suffers from the communication bottleneck, which is mainly caused by the need to upload cumbersome local parameters. To overcome this bottleneck, we leverage a strategy based on knowledge distillation. Such a strategy uses the uploaded predictions of ensemble local models to train the central model without requiring uploading local parameters. Experiments on three publicly available medical relation extraction datasets demonstrate the effectiveness of our method.

主要是efficient，没涉及heterogeneous和non-IID的情况

## 1 Introduction

*Privacy - like eating and breathing - is one of life's basic requirements.*

— Katherine Neville

Relation extraction is a task of mining factual knowledge from the free text by labeling relations between entity mentions and has attracted increasing attention in recent years, such as Zeng et al. (2014); Xu et al. (2015a,b); Wang et al. (2016); Baldini Soares et al. (2019); Song et al. (2019). Applying automatic relation extraction to medical texts, such as electronic health records and discharge summaries, can be useful for many applications, including drug repurposing and medical knowledge graph construction.

Unlike other domains, medical texts are highly privacy-sensitive, because these texts can include some of the most intimate details about one's life, which document a patient's physical and mental health, and can include information on social behaviors, personal relationships and financial status (Gostin and Hodge, 2002). To prevent private information leakage, sharing or copying medical texts is strictly restricted.

Previous relation extraction methods require centralizing the underlying training data from different medical platforms, such as hospitals and healthcare centers, on one server for training, while holding the centralized privacy-sensitive data puts patients' privacy at risk. This is one of the reasons that hinder the use of relation extraction in clinical practice. As a possible solution, federated learning (McMahan et al., 2016) is proposed to make full use of privacy-sensitive data. Training local models with private data at local platforms and aggregating local models in the central server compose the federated learning process. In the framework of federated learning, no single piece of private data is uploaded to or stored on the central server, and only local models' parameters are sent to the server for updating the central model.

Though federated learning has distinct advantages in privacy protection compared to centralized training, federated learning algorithms, such as FedAvg (McMahan et al., 2016), require frequent communication between local platforms and the central server to upload and download models' parameters. Communication is a critical bottleneck of applying federated learning to relation extraction, which is largely due to the following reasons: First, the state-of-the-art relation extraction models (Baldini Soares et al., 2019; Li et al., 2019b; Thillaisundaram and Togia, 2019) usually utilize transformer-based pretrained language models (Raffel et al., 2019; Devlin et al., 2019; Liu et al.,

*Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 2118–2128,
November 16–20, 2020. ©2020 Association for Computational Linguistics

2019; Yang et al., 2019b) as backbone encoders, which have millions or even billions of parameters. Second, the framework of federated learning includes a massive number of local platforms (Li et al., 2019a), and communication between each platform and the central server is necessary. Third, upload bandwidth is typically limited to 1 MB/s or less in most situations [1]. Considering the cumbersome model, numerous local platforms and the limited upload bandwidth, it will take an excessive amount of time during frequent upload processes. For example, in a single communication, uploading a BERT-Large (Devlin et al., 2019) model takes more than 21 minutes and uploading a T5 (Raffel et al., 2019) model takes more than 12 hours. In order to overcome the communication bottleneck in federated relation extraction, it is necessary to develop a communication-efficient method that iteratively sends small messages as part of the training process, as opposed to sending the entire pretrained language encoder.

In this paper, we introduce a privacy-preserving medical relation extraction model, named `FedED`. To prevent private information leakage, we leverage federated learning without sharing raw privacy-sensitive medical texts. To overcome the communication bottleneck in federated relation extraction, we focus on reducing the size of transmitted messages at each communication round. To this end, we formulate the central aggregation process in federated learning as learning a compact central model (student) from the ensemble (Dietterich, 2000; Breiman, 2001) of multiple local models (teacher). From this perspective, only the predicted labels on a small dataset need to be uploaded to the central server, because learning from a "teacher" model only requires the behavior of the "teacher" rather than the entire"teacher" network (Hinton et al., 2015). Besides, the ensemble model (teacher) is powerful, which defines the upper extreme of aggregating when limited to a single communication in federated learning (Yurochkin et al., 2019). To transfer the knowledge in the ensemble model to the central model, we leverage a strategy based on knowledge distillation (Hinton et al., 2015), which trains the central model by forcing it to have a similar prediction with the ensemble model. To demonstrate the effectiveness of our method, we conduct extensive experiments on three different medical relation extraction datasets. The results show that our method not only outperforms the baselines but also is communication-efficient.

We summarize our contributions as follows:

- To protect patients' privacy, we propose the first (to the best of our knowledge) privacy-preserving medical relation extraction model based on federated learning, which decouples the model training from the need for direct access to the highly privacy-sensitive data.
- To overcome the communication bottleneck in federated learning, we leverage a knowledge distillation based strategy that utilizes the uploaded predictions of ensemble local models to train the central model without requiring uploading the entire local models' parameters.
- The method yields promising results on three different medical relation extraction datasets, and we perform various experiments to verify the effectiveness of the proposed method.

## 2 Related Work

Our work builds on a rich line of recent efforts on relation extraction models and federated learning.

### 2.1 Relation Extraction

Relation extraction is a long-standing NLP task of mining factual knowledge from free texts by labeling relations between entity mentions. There are a number of recent neural network approaches applied to relation extraction, such as Zeng et al. (2014); Nguyen and Grishman (2015); dos Santos et al. (2015); Zhang and Wang (2015); Zhang et al. (2017). Recently, the NLP community has seen excitement around neural models that make heavy use of pretraining based on language modeling (Radford et al.; Devlin et al., 2019; Liu et al., 2019; Yang et al., 2019b). Baldini Soares et al. (2019); Shi and Lin (2019) and Alt et al. (2019) achieved the state-of-the-art performance in relation extraction by fine-tuning the pretrained language models. In this paper, we also adopt a pretrained language model as the backbone encoder.

Applying relation extraction models to the medical field has great practical value, and there is a rich literature on medical relation extraction. Some studies focused on clinical relation extraction (Sahu et al., 2016; Munkhdalai et al., 2018; Ningthoujam et al., 2019) and some studies concentrated on

[1] the bandwidth and bitrate of the download are much greater than the upload, so we only consider the upload process.(`en.wikipedia.org/wiki/ADSL`).

biomedical relation extraction (Peng et al., 2017; Song et al., 2018, 2019). Compared with previous studies, we develop a federated relation extraction system to protect patients' privacy in medical relation extraction.

### 2.2 Federated Learning

Recently, McMahan et al. (2016), Konečnỳ et al. (2016a) and Konečnỳ et al. (2016b) proposed the concept of federated learning. The main idea of federated learning is to build machine learning models based on data sets that are distributed across multiple local platforms while preventing data leakage. Federated learning can be divided into three categories, i.e., horizontal federated learning, vertical federated learning and federated transfer learning, based on the distribution characteristics of the data (Yang et al., 2019a). This work focuses on horizontal federated learning, where local datasets share the same feature space but different in samples. There are a number of studies about horizontal federated learning, such as McMahan et al. (2016); Sahu et al. (2018); Ji et al. (2019); Wang et al. (2020).

Federated learning has the advantage of protecting privacy, so it is widely used in various fields. Chen et al. (2018) combined federated learning with meta learning for the recommendation. Kim et al. (2017) proposed federated tensor factorization for computational phenotyping without sharing patient-level data. Liu and Miller (2020) proposed federated pretraining of BERT model using clinical notes from multiple silos. Ge et al. (2020) proposed a privacy-preserving medical NER method based on federated learning.

## 3 Method

### 3.1 Task Definition

Relation Extraction devotes to extracting relational facts from sentences. Given a sentence with an entity pair $e_1$ and $e_2$, this task aims to identify the relation between $e_1$and $e_1$. In this paper, we focus on applying relation extraction to the medical domain. Define $K$ medical platforms $\{\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_K\}$, each with a private relation extraction dataset $D_i$, and a central server that has a small valid dataset $D_v$. Since the medical data is usually private and sensitive, the goal is to obtain a medical relation extraction model on the central server under the condition that any local platform $\mathcal{P}_i$ does not expose its private data $D_i$ to others.

To solve this task, we propose a privacy-preserving medical relation extraction model based on federated learning. In the following sections, we introduce the basic medical relation extraction model at first. Then, we present how to conduct privacy-preserving training in a communication-efficient way.

### 3.2 Medical Relation Extraction Model

Given the impressive performance of recent deep transformers (Vaswani et al., 2017) trained on variants of language modeling, we utilize the BERT model (Devlin et al., 2019) as the backbone encoder. In this section, we explore a simple way of representing relations with the deep transformers model. The model architecture is shown in Figure 1 and the details are as follows:

Firstly, we construct the input sequence $s = \{w_0, w_1, ..., w_n\}$, where $w_0$ = [CLS] and $w_n$ = [SEP] are special start and end markers. Next, to ensure generalization of the model, we follow previous studies (He et al., 2013; Kim et al., 2015; Liu et al., 2016; Chauhan et al., 2019) to perform entity blinding on the sequence, where the words in the sequence matching the entity are replaced with the target entity label. Then, in order to highlight entity mentions, we augment the sequence with four reserved word pieces, i.e., ⟨e1⟩,⟨/e1⟩,⟨e2⟩ and ⟨/e2⟩, to mark the begin and end of each entity mention. After that, we get the prepared sequence $\hat{s}$.

$$\begin{aligned}\hat{s} =&\{[\text{CLS}], w_1, ..., w_{i-2}, \langle \text{e1} \rangle, w_i, ..., w_j, \langle /\text{e1} \rangle,\\ &..., \langle \text{e2} \rangle, w_k, ..., w_l, \langle /\text{e2} \rangle, ..., w_{m-2}, [\text{SEP}]\}\end{aligned}$$

Given the prepared sequence $\hat{s}$ as input, the output of BERT is expressed as $H \in \mathcal{R}^{m \times d}$, where $m$ is the prepared sequence length and $d$ is the output dimension of the BERT encoder. We use the first token of the sequence (the [CLS] token) as the sequence representation, which is denoted as $h_0 \in \mathrm{R}^d$. In addition, we obtain entity mention representations by summing the final hidden layers corresponding to the word pieces in each entity mention, and get two vectors $h_{e_1} = \mathrm{sum}([h_i...h_j]) \in \mathrm{R}^d$ and $h_{e_2} = \mathrm{sum}([h_k...h_l]) \in \mathrm{R}^d$ representing the two entity mentions. Finally, the sequence representation and these two entity mention representations are concatenated to be the input of a fully connected layer:

$$h = h_0 \oplus h_{e_1} \oplus h_{e_2} \in \mathrm{R}^{3d} \quad (1)$$

$$p(y|\hat{s}, \Theta) = \mathrm{softmax}(\mathrm{Wh} + \mathrm{b}) \quad (2)$$

where W and b are trainable model parameters.

client基于server提供的验证集（有标签）做预测，然后上传预测的logits，server基于logtis avg做KD

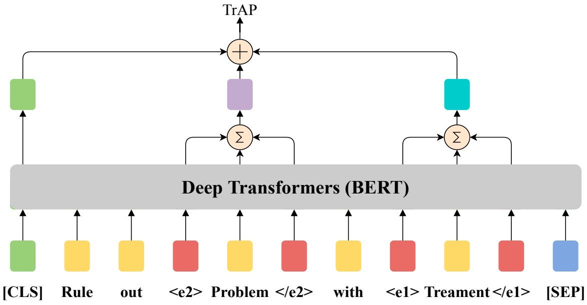

Figure 1: The architecture of our medical relation extraction model.

### 3.3 Federated Training

To protect patients' privacy, we utilize federated learning to train the medical relation extraction model. In the federated framework, two types of models are needed, i.e., the local model and the central model, which share the same network structure but have different permissions to access private data. Local models are deployed in local platforms, such as hospitals, and can access their respective private local data. In contrast, the central model is deployed in a central server, such as a cloud server, which is strictly prohibited from accessing to patients' private data. Here, following previous studies (McMahan et al., 2016; Bonawitz et al., 2019; Ge et al., 2020), we assume the central server belongs to one trusted third party, which means it will not make any vicious attack to local platforms. In this section, we present how to train the relation extraction model in the federated way, including secure local model update and the ensemble distillation based central model update.

#### 3.3.1 Secure Local Model Update

The local model in each medical platform is trained on its own private data. We assume that the local platform $\mathcal{P}_i$ is selected to perform local computation in a round. The local platform $\mathcal{P}_i$ computes the gradients of loss over all the data $D_i$ held by it to update the parameters of the local model. We adopt the cross-entropy as the local loss function, which is defined as follows:

$$L_{local}(\Theta) = -\frac{1}{|D_i|}\sum_{i=1}^{|D_i|} \log p(y_i|\hat{s}_i, \Theta) \qquad (3)$$

where $|D_i|$ represents the number of sentences in this local private data $D_i$ and $\Theta$ indicates all parameters of the local model. After local model training, the local model in $\mathcal{P}_i$ accesses to valid data $D_v$ in the central server, makes a prediction on it based on the trained parameters and uploads the predicted labels to the central server. Compared with centralized training, the local model is only trained on its own data, and only the predicted labels are uploaded rather than directly sharing raw data, which generally contains less privacy-sensitive information.

#### 3.3.2 Central Model Update via Ensemble Distillation

The central server coordinates massive local models to collaboratively train the central model. To this end, there are a coordinator and an aggregator in the central server.

**Coordinator** controls the entire training process and is responsible for accepting and forwarding local platform connections. At the beginning of each communication round, the coordinator builds the medical relation extraction model in the central server and initializes the model. Then, the coordinator randomly selects a $C$-fraction of local medical platforms, since we cannot require that all local platforms are always online in the real-world scenario. After that, the coordinator distributes the parameters of central model to all selected local platforms, and the selected local models are initialized based on these parameters, which ensures that all selected local models are trained from the same initial condition at this round. Then, the selected local models are trained on their respective private data at each local platform. The coordinator monitors each selected platform for any possible uploads. Once it receives uploads from one platform, the coordinator will store them for future aggregation. When all selected platforms finish local training, the stored uploads are sent to the aggregator to inference new central model parameters.

**Aggregator** is the most critical part of federated training, which optimizes the central model based on massive trained local models. To transfer the knowledge in the massive trained local models to the central model, we resort to teacher-student framework. The ensemble of local models is viewed as the teacher, while the central model is regarded as the student. The knowledge in the teacher is transferred to the central model by forcing them to have a similar prediction for any input instance. To this end, the central model is trained to minimize a distillation loss function where the target is the distribution of class probabilities predicted by the ensemble model. The typical choice of the distillation loss function is the Kullback-Leibler (KL) divergence between the distributions, $D_{KL}(q||p)$, where $p$ and $q$ are the output label distributions of the student and the teacher respectively. The distribution of the teacher can be attained as follows:

$$q(y_i|s) = \frac{\exp\left(z(y_i|s)/\tau\right)}{\sum_r \exp\left(z(y_r|s)/\tau\right)} \quad (4)$$

$$z(y_i|s) = \frac{1}{|\mathcal{J}|}\sum_{j\in\mathcal{J}} p(y_i|s,\Theta^{(j)}) \quad (5)$$

where $z(y_i|s)$ is the logit of ensemble model (teacher) for class i, which is represented as the mean of selected local models' logits for this class, and $\tau$ is a temperature parameter that controls the shape of the distribution for distilling richer knowledge from the ensemble model. In addition to the distillation loss, it is also beneficial to train the central model to predict the ground truth labels using the standard cross-entropy loss. The overall objective is defined as follows:

KD：基于server提供的公共验证集

$$L = \frac{1}{|D_v|}\sum_{i=1}^{|D_v|}\left(-\log p(y_i|\hat{s}_i) + D_{KL}(q(y|\hat{s}_i)||p(y|\hat{s}_i))\right) \quad (6)$$

The overall training procedure of `FedED` is illustrated in Algorithm 1.

## 4 Experiments

In this section, we carry out an extensive set of experiments with the aim of answering the following research questions:

- **RQ1**: Does our model outperform the baseline methods? (see Section 4.4)
- **RQ2**: Is federated learning effective in medical relation extraction? (see Section 4.5)
- **RQ3**: Is our approach communication-efficient? (see Section 4.6)
- **RQ4**: What is the impact of increasing parallelism on our model? (see Section 4.7)
- **RQ5**: What is the impact of increasing computation per local platform on our model? (see Section 4.8)

In the remainder of the section, we describe the datasets, experimental setting, and all baselines.

**Algorithm 1** `FedED`. The K local platforms are indexed by $k$. C is the fraction of local platforms that perform computation on each round. B is the local minibatch size. E is the number of local epochs, and $\eta$ is the learning rate.

```
Initialize $\Theta_0$ on the central server
for each communication round t = 0,1,2,... do
    m ← max(C× K, 1)
    $\mathcal{J}_t$ ← (random set of m local platforms)
    The server distributes $\Theta_t$ to $\mathcal{J}_t$.
    for each platform $k \in \mathcal{J}_t$ in parallel do
        Perform LocalUpdate(k, $\Theta_t$)
    end for
    // The procedure of Aggregator
    $\mathcal{V}$ ← (split $D_v$ into batches)
    for batch v in $\mathcal{V}$ do
        $\Theta_t \leftarrow \Theta_t - \eta\nabla L(\Theta_t; v)$
        // L is defined in Equation 6
    end for
    $\Theta_{t+1} \leftarrow \Theta_t$
end for

function LocalUpdate(k, $\Theta$):
    // Run on local platform k
    $\mathcal{B}$ ← (split $D_k$ into batches of size B)
    // $D_k$ is the private data of local platform k
    for each local epoch i from 1 to E do
        for batch b in $\mathcal{B}$ do
            $\Theta \leftarrow \Theta - \eta\nabla L_{local}(\Theta; b)$
            // $L_{local}$ is defined in Equation 3
        end for
    end for
    Access to $D_v$
    return $\{p(y|s,\Theta)|s \in D_v\}$ to server
```

### 4.1 Datasets

We conduct experiments on three publicly available medical relation extraction datasets: **2010 i2b2/VA challenge dataset** (Uzuner et al., 2011)[2], **GAD**(Bravo et al., 2015) and **EU-ADR** (Van Mulligen et al., 2012) [3]. The 2010 i2b2/VA challenge dataset is collected from three different hospitals viz, Partners Healthcare, Beth Israel Deaconess Medical Center, and the University of Pittsburgh Medical Center. It consists of discharge-summary and progress notes of the patients, and is manually annotated by medical practitioners. The EU-ADR dataset is annotated on a part of Medline

[2] https://portal.dbmi.hms.harvard.ed

[3] https://github.com/dmis-lab/biobert

| Dataset | Entiy Type | #Train | #Test | Relation |
|---|---|---|---|---|
| 2010 i2b2/VA | Test, Treatment and Problem | 10231 | 19114 | TrIP, TrWP, TrCP, TrAP, TrNAP, TeRP, TeCP, PIP |
| GAD | Gene and Disease | 4797 | 533 | True/False Associations |
| EU-ADR | Gene and Disease | 320 | 35 | True/False Associations |

Table 1: Statistics of the medical relation extraction datasets

| Relation Type | Number Instances |
|---|---|
| TeRP | 3053 |
| TrAP | 2617 |
| TrCP | 526 |
| PIP | 2203 |
| TeCP | 504 |
| TrIP | 203 |
| TrWP | 133 |
| TrNAP | 174 |
| None | 19932 |

Table 2: Relation types and number of instances of i2b2 dataset

abstracts from 2007 to 2008, and the GAD dataset is collected form Genetic Association Database, an archive of human genetic association studies of complex diseases and disorders. The detailed statistics of these three datasets are listed in Table 1 and 2. We random sample 20% of training data for validation. To evaluate our method, we use the standard evaluation metric for each dataset: Micro-F1 for 2010 i2b2/VA challenge dataset and F1-score for GAD and EU-ADR.

### 4.2 Experimental Settings

We use a controlled environment that is suitable for experiments and assume a synchronous update scheme that proceeds in rounds of communication. For 2010 i2b2/VA challenge dataset, we set the number of local platforms ($K$) to 100. For EU-ADR and GAD datasets, the number of local platforms ($K$) is set to 50, since these datasets are small. The training data is randomly shuffled and then partitioned into $K$ local platforms each receiving $1/K$ of the training data. This data partitioning simulates the scenario where each hospital is treated as a local platform and the central server is located in a trusted third party.

In our experiments, we use hugginface's implementation (Wolf et al., 2019) of BERT (base version) and initialize parameters of the BERT encoding layer with pretrained clinical BERT (Alsentzer et al., 2019) models. The learning rate is set to 2.5e-05. We use the dropout strategy to mitigate overfitting, which is set to 0.1. To conduct a fair comparison (presented in Section 4.4), we set all federated methods hyper-parameters as follows. The random fraction of local platforms $C$ is 0.1, and we also study adding more local platforms at each round of communication in Section 4.7. Since the batch size and the number of local epochs are related to the number of secure local updates per round, the batch size $B$ is fixed to 4 and the number of local epochs $E$ is set to 2. We independently repeat each experiment 9 times and report the median F-score. All experiments are run with an NVIDIA GeForce RTX 2080 Ti.

### 4.3 Baselines

Under centralized training settings, we compare our medical relation extraction model (depicted in Section 3.2) with the following studies: (1) Sahu et al. (2016) leverage convolutional neural network (CNN) to extract relations in clinical texts; (2) Chauhan et al. (2019) build CNN upon the embeddings generated by the BERT model and train the models with a ranking loss; (3) Bravo et al. (2015) combine the shallow linguistic kernel with the dependency kernel to mine the syntactic features of text; (4) Bhasuran and Natarajan (2018) employ an ensemble SVM with a rich feature set covering conceptual, syntax and semantic information; (5) Lee et al. (2020) propose a domain-specific language representation model, called BioBERT, pre-trained on large-scale biomedical corpora.

In the federated training manner, We compare our federated framework (depicted in Section 3.3) with the following baselines: (1) FedAvg (McMahan et al., 2016) averages element-wise parameters of local models with weights proportional to sizes of the local datasets; (2) FedAtt (Ji et al., 2019) leverages a layer-wise attention mechanism for model aggregation. which can automatically attend to the weights of the relation between the central model and different local models.

### 4.4 Results

Table 3, 4 and 5 answer **RQ1** by showing the results of our model against baselines on the real-world medical datasets. In overall, our model significantly outperforms baselines on these datasets.

In the centralized training manner, our method outperforms REflex (Chauhan et al., 2019) on i2b2 dataset, which builds CNN upon the embeddings

| Method | P | R | F1 |
|---|---|---|---|
| Centralized Training | | | |
| Bravo et al. (2015) | 77.80 | 87.20 | 82.20 |
| Bhasuran and Natarajan (2018) | **79.21** | 89.25 | **83.93** |
| Lee et al. (2020) | 77.32 | 82.68 | 79.83 |
| Our | 77.58 | **91.1** | 83.8 |
| Federated Training | | | |
| FedAvg | 69.89 | 87.54 | 77.73 |
| FedAtt | 72.14 | 87.54 | 79.1 |
| Our | **74.77** | **88.61** | **81.11** |

Table 3: Results on GAD

| Method | P | R | F1 |
|---|---|---|---|
| Centralized Training | | | |
| Bravo et al. (2015) | 75.1 | 97.7 | 84.6 |
| Bhasuran and Natarajan (2018) | 76.43 | **98.01** | 85.34 |
| Lee et al. (2020) | 77.86 | 83.55 | 79.74 |
| Our | **78.79** | 96.3 | **86.67** |
| Federated Training | | | |
| FedAvg | 71.43 | 92.59 | 80.65 |
| FedAtt | 72.22 | 96.3 | 82.54 |
| Our | **74.29** | **96.3** | **83.87** |

Table 4: Results on EU-ADR

| Method | P | R | F1 |
|---|---|---|---|
| Centralized Training | | | |
| Sahu et al. (2016) | **76.34** | 67.35 | 71.16 |
| Raj et al. (2017) | 67.91 | 61.98 | 64.38 |
| Chauhan et al. (2019) | – | – | 71.01 |
| Our | 74.78 | **80.1** | **77.35** |
| Federated Training | | | |
| FedAvg | 74.75 | 70.48 | 72.55 |
| FedAtt | 74.48 | 71.32 | 72.86 |
| Our | **75.4** | **74.78** | **75.09** |

Table 5: Results on 2010 i2b2/VA challenge dataset

generated by the BERT model. We conjecture this is largely due to that our model adopts the fine-tuning strategy on the relation extraction tasks instead of leveraging fixed embeddings generated by BERT. Previous studies (Peters et al., 2019) on BERT show that fine-tuning significantly outperforms the frozen pretrained weights strategy. Our method outperforms BioBERT (Lee et al., 2020) on EU-ADR and GAD datasets. The BioBERT only uses sequence representation, while our method use both sequence representations and entity mention representations. Moreover, we introduce four entity markers to highlight entity mention. Previous research (Baldini Soares et al., 2019) on relation extraction shows that entity markers and entity mention representation has a positive impact on the result.

In the federated training manner, our federated framework outperforms FedAvg (McMahan et al., 2016) and FedAtt (Ji et al., 2019). There are two possible reasons: (1) The performance of the ensemble model defines the upper extreme of aggregating when limited to a single communication in federated learning (Yurochkin et al., 2019), and the central model benefits from learning from the ensemble model. (2) FedAvg and FedAtt only model the simple process of central optimization by averaging or weighted averaging local model parameters, which overlook complicated relationships between local model parameters. FedED forces the central model to mimic the behavior of the ensemble model rather than modeling the complex relationship between parameters.

Comparing the federated training manner to the centralized training manner, we find that applying the centralized training manner achieves better performance. There are three reasons: (1) In federated learning, a 10% fraction of local platforms are selected in each epoch. In other words, only 10% of the training examples are used in each epoch. However, all training examples are used at each epoch in centralized training. (2) As the size of each local private data is small, the local model is prone to overfitting on it. (3) The local platforms are independent of each other; therefore, compared with centralized training, federated training lack the ability to model the overall data distribution. Although federated training does not perform as well as centralized learning, federated training is uniquely positioned to protect privacy. Moreover, our approach narrows the gap between federated training and centralized training in terms of performance.

### 4.5 Effectiveness Test of Federated Learning

To test the effectiveness of federated learning, we simulate a real-world scenario where a third party only has a small data, i.e., validation set, and copying data from hospitals is prohibited. The results are shown in Table 6, which answers the **RQ2**. From this table, we find that: (1) Due to data scarcity, the model trained only on the validation set can not achieve satisfactory performance; (2) FedAvg and FedAtt can effectively improve the performance of relation types with abundant examples, such as “TeRP”, “TrAP” and ”PIP”, but perform poorly in relation types with few examples (The distribution of relation types is shown in Table 2);

| Relation Types | Trained on Validation Set | FedAvg | FedAtt | FedED |
|---|---|---|---|---|
| TeRP | 79.74 | 84.96 | 85.24 | 85.3 |
| TrAP | 67.75 | 72.68 | 71.55 | 76.05 |
| TrCP | 43.11 | 50.41 | 47.73 | 54.9 |
| PIP | 66.98 | 71.07 | 75.16 | 73.69 |
| TeCP | 27.36 | 49.41 | 51.46 | 58.02 |
| TrIP | 24.24 | 23.08 | 1.3 | 48.03 |
| TrWP | 0 | 0 | 0 | 11.97 |
| TrNAP | 22.54 | 21.79 | 18.79 | 56.16 |

Table 6: The classwise performance on the 2010 i2b2/VA challenge dataset

(3) Our proposed FedED is able to improve performance in all relation types. We conjecture that ensemble distillation can capture the rich similarity structure between relation types, which boosts the performance.

### 4.6 Communication Efficiency Test

We turn to **RQ3** in this section. Table 7 presents the message size uploaded by each local platform at each communication round. From this table, we notice that our proposed method is communication-efficient and the amount of data uploaded by our method is much smaller than the others. The reason is that FedAvg and FedAtt require each selected local platform to upload the entire medical relation extraction model at each communication round, while only the predicted labels on a small dataset are uploaded to the central server in FedED. Considering numerous local platforms and the limited upload bandwidth, our proposed method can save a lot of time in communication.

| Method | i2b2 | GAD | EU-ADR |
|---|---|---|---|
| FedAvg | 423MB | 423MB | 423MB |
| FedAtt | 423MB | 423MB | 423MB |
| FedED | 42KB | 5KB | 323B |

Table 7: The message size uploaded by each local platform at each communication round.

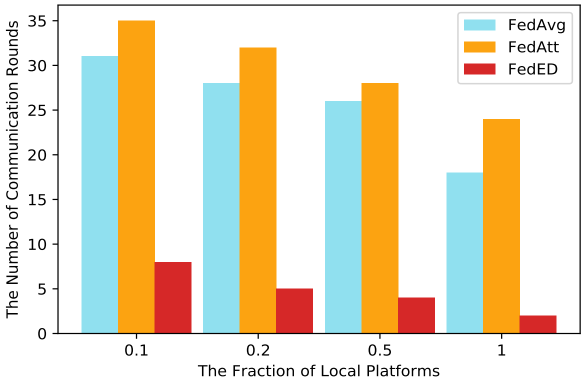

Figure 2: The number of communication rounds necessary to achieve an F1 value of 72% on the test set, fixing local epoch $E$ to 2 and the batch size $B$ to 4.

### 4.7 Increasing Parallelism

Figure 2 answers **RQ4** by showing the impact of varying the fraction of local platforms for all approaches on the 2010 i2b2/VA challenge dataset. The fraction of local platforms $C$ controls the amount of local platforms selected by the coordinator in each round. In Figure 2, we report the number of communication rounds necessary to achieve an F1 value of 72% on the test set. We find that: (1) Increasing parallelism will speed up convergence for all methods. When all local platforms are selected ($C = 1$), all methods reach the target F1 value with minimal communication cost. This is mainly due to the fact that the increased parallelism leads to more data used in each round of training; (2) Our method requires a much smaller number of communication rounds to reach the target F1 value than the other methods. We conjecture that this is due to that the central model (student) learns much faster and more reliably when trained with the outputs of the ensemble model (teacher) as soft labels (Phuong and Lampert, 2019).

### 4.8 Increasing Computation Per Platform

Finally, we address **RQ5**. The number of local computation per round is given by $\frac{|D_k|}{B}E$, where $B$ is the local batch size, $E$ is the number of local training epoch and $|D_k|$ is the size of private data in local platform $k$. Decreasing $B$, increasing $E$, or both will add more computation per local platform per round. Table 8 lists the number of communication rounds necessary to achieve an F1

| **B** | **E** | FedAvg | FedAtt | FedED |
|---|---|---|---|---|
| 2 | 1 | 27 | 28 | 9 |
| 8 | 1 | 48 | 48 | 20 |
| 16 | 1 | – | – | 34 |
| 2 | 3 | 20 | 20 | 7 |
| 8 | 3 | 34 | 30 | 11 |
| 16 | 3 | 47 | 46 | 15 |

Table 8: The number of communication rounds necessary to achieve an F1 value of 72% on the test set, fixing $C$ to 0.1. "–" means that the run did not reach the target F1 value in the allowed time.

value of 72% with different $E$ and $B$. From this Table, we see that increasing computation per local platform by varying both $B$ and $E$ is effective for all methods, and our method converges to the target F1 value faster than baselines.

## 5 Conclusion and Future Work

In this paper, we propose a privacy-preserving medical relation extraction model based on federated learning, namely FedED. The main obstacle of applying federated learning to medical relation extraction is communication bottleneck, which is caused by the need to upload cumbersome parameters. To overcome this bottleneck, we leverage a knowledge distillation based strategy, which uses the uploaded predictions of ensemble local models to train the central model without requiring uploading cumbersome parameters. Our experiments on three benchmark datasets illustrate the advantages of our approach over previous federated algorithms. As to future work, we plan to explore how to jointly extract entities and relations in federated settings.

## Acknowledgments

We thank the anonymous reviewers for their insightful comments. We also thank Yushan Xie for helpful comments and suggestions.

This work is supported by the National Natural Science Foundation of China (No.61533018, No.61806201), the Key Research Program of the Chinese Academy of Sciences (Grant NO. ZDBS-SSW-JSC006), the independent research project of National Laboratory of Pattern Recognition and Huawei Technologies Co., Ltd.

## References

Emily Alsentzer, John Murphy, William Boag, Wei-Hung Weng, Di Jin, Tristan Naumann, and Matthew McDermott. 2019. Publicly available clinical BERT embeddings. In *Proceedings of the 2nd Clinical Natural Language Processing Workshop*, pages 72–78, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Christoph Alt, Marc Hübner, and Leonhard Hennig. 2019. Improving relation extraction by pre-trained language representations. *arXiv preprint arXiv:1906.03088*.

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

Balu Bhasuran and Jeyakumar Natarajan. 2018. Automatic extraction of gene-disease associations from literature using joint ensemble learning. *PloS one*, 13(7).

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, Dzmitry Huba, Alex Ingerman, Vladimir Ivanov, Chloe Kiddon, Jakub Konecny, Stefano Mazzocchi, H Brendan McMahan, et al. 2019. Towards federated learning at scale: System design. *arXiv preprint arXiv:1902.01046*.

Àlex Bravo, Janet Piñero, Núria Queralt-Rosinach, Michael Rautschka, and Laura I Furlong. 2015. Extraction of relations between genes and diseases from text and large-scale data analysis: implications for translational research. *BMC bioinformatics*, 16(1):55.

Leo Breiman. 2001. Random forests. *Machine learning*, 45(1):5–32.

Geeticka Chauhan, Matthew McDermott, and Peter Szolovits. 2019. Reflex: Flexible framework for relation extraction in multiple domains. *arXiv preprint arXiv:1906.08318*.

Fei Chen, Zhenhua Dong, Zhenguo Li, and Xiuqiang He. 2018. Federated meta-learning for recommendation. *arXiv preprint arXiv:1802.07876*.

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

Thomas G Dietterich. 2000. Ensemble methods in machine learning. pages 1–15.

Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. 2020. Fedner: Medical named entity recognition with federated learning. *arXiv preprint arXiv:2003.09288*.

LO Gostin and JG Hodge. 2002. Personal privacy and common goods: a framework for balancing under the national health information privacy rule. *Minnesota law review*, 86(6):1439.

Linna He, Zhihao Yang, Zhehuan Zhao, Hongfei Lin, and Yanpeng Li. 2013. Extracting drug-drug interaction from the biomedical literature using a stacked generalization-based approach. *PloS one*, 8(6).

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. 2019. Learning private neural language modeling with attentive aggregation. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Sun Kim, Haibin Liu, Lana Yeganova, and W John Wilbur. 2015. Extracting drug–drug interactions from literature using a rich feature-based linear kernel approach. *Journal of biomedical informatics*, 55:23–30.

Yejin Kim, Jimeng Sun, Hwanjo Yu, and Xiaoqian Jiang. 2017. Federated tensor factorization for computational phenotyping. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 887–895. ACM.

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

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

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. 2019a. Federated learning: Challenges, methods, and future directions. *arXiv preprint arXiv:1908.07873*.

Xiaoya Li, Fan Yin, Zijun Sun, Xiayu Li, Arianna Yuan, Duo Chai, Mingxin Zhou, and Jiwei Li. 2019b. Entity-relation extraction as multi-turn question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1340–1350, Florence, Italy. Association for Computational Linguistics.

Dianbo Liu and Tim Miller. 2020. Federated pretraining and fine tuning of bert using clinical notes from multiple silos. *arXiv preprint arXiv:2002.08562*.

Shengyu Liu, Buzhou Tang, Qingcai Chen, and Xiaolong Wang. 2016. Drug-drug interaction extraction via convolutional neural networks. *Computational and mathematical methods in medicine*, 2016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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

Tsendsuren Munkhdalai, Feifan Liu, and Hong Yu. 2018. Clinical relation extraction toward drug safety surveillance using electronic health record narratives: classical learning versus deep learning. *JMIR public health and surveillance*, 4(2):e29.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado. Association for Computational Linguistics.

Dhanachandra Ningthoujam, Shweta Yadav, Pushpak Bhattacharyya, and Asif Ekbal. 2019. Relation extraction between the clinical entities based on the shortest dependency path based lstm. *arXiv preprint arXiv:1903.09941*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. *Transactions of the Association for Computational Linguistics*, 5:101–115.

Matthew E Peters, Sebastian Ruder, and Noah A Smith. 2019. To tune or not to tune? adapting pretrained representations to diverse tasks. pages 7–14.

Mary Phuong and Christoph Lampert. 2019. Towards understanding knowledge distillation. pages 5142–5151.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.

Desh Raj, Sunil Sahu, and Ashish Anand. 2017. Learning local and global contexts using a convolutional recurrent network model for relation classification in biomedical text. pages 311–321.

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

Sunil Kumar Sahu, Ashish Anand, Krishnadev Oruganty, and Mahanandeeshwar Gattu. 2016. Relation extraction from clinical texts using domain invariant convolutional neural network. *arXiv preprint arXiv:1606.09370*.

Cícero dos Santos, Bing Xiang, and Bowen Zhou. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 626–634, Beijing, China. Association for Computational Linguistics.

Peng Shi and Jimmy Lin. 2019. Simple bert models for relation extraction and semantic role labeling. *arXiv preprint arXiv:1904.05255*.

Linfeng Song, Yue Zhang, Daniel Gildea, Mo Yu, Zhiguo Wang, and Jinsong Su. 2019. Leveraging dependency forest for neural medical relation extraction.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph-state lstm. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2226–2235.

Ashok Thillaisundaram and Theodosia Togia. 2019. Biomedical relation extraction with pre-trained language representations and minimal task-specific architecture. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 84–89, Hong Kong, China. Association for Computational Linguistics.

Özlem Uzuner, Brett R South, Shuying Shen, and Scott L DuVall. 2011. 2010 i2b2/va challenge on concepts, assertions, and relations in clinical text. *Journal of the American Medical Informatics Association*, 18(5):552–556.

Erik M Van Mulligen, Annie Fourrier-Reglat, David Gurwitz, Mariam Molokhia, Ainhoa Nieto, Gianluca Trifiro, Jan A Kors, and Laura I Furlong. 2012. The eu-adr corpus: annotated drugs, diseases, targets, and their relationships. *Journal of biomedical informatics*, 45(5):879–884.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

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

Linlin Wang, Zhu Cao, Gerard De Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. 2015a. Semantic relation classification via convolutional neural networks with simple negative sampling. *arXiv preprint arXiv:1506.07650*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015b. Classifying relations via long short term memory networks along shortest dependency paths. pages 1785–1794.

Qiang Yang, Yang Liu, Tianjian Chen, and Yongxin Tong. 2019a. Federated machine learning: Concept and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 10(2):1–19.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019b. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Trong Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. *arXiv preprint arXiv:1905.12022*.

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, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

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

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.