FedNLP: Federated Learning for Natural Language Processing

Xiang Ren, USC CS
Mahdi Soltanolkotabi, USC ECE
Prof. Xiang Ren (PI)

• **Expertise**: natural language processing, explainable AI, continual learning
• Make NLP systems trustworthy, cheaper to build, easier to maintain
• Create benchmark datasets for a range of NLP tasks

Commonsense reasoning

Human-in-the-loop learning, Explainability & Robustness

Cross-task Generalization
Prof. Mahdi Soltanolkotabi

• **Expertise:** Foundations of AI & ML, (non)convex optimization, high-dimensional statistics and probability
• Developed some of the first guarantees for deep learning
• Extensive contributions to distributed/federated learning

![Graphs and illustrations related to optimization and generalization in machine learning.]
Bill Yuchen Lin, PhD candidate @ USC

• (Bill) Yuchen Lin is a Ph.D. candidate in USC CS, working with Prof. Xiang Ren at the Intelligence and Knowledge Discovery Research Lab

• Develop intelligent systems that demonstrate a deep understanding of the world with common-sense knowledge and reasoning ability

• Research interests: information extraction, knowledge graphs, logical reasoning, graph neural networks, explanations, robustness
Chaoyang He, PhD candidate @ USC

• Chaoyang He is a Ph.D. Candidate in the CS department at the University of Southern California. He is advised by Professor Salman Avestimehr, Professor Mahdi Soltanolkotabi, and Professor Murali Annavaram (USC). He also works closely with researchers/engineers at Google, Facebook, Amazon, and Tencent. Previously, He was an R&D Team Manager and Staff Software Engineer at Tencent (2014-2018), a Team Leader and Senior Software Engineer at Baidu (2012-2014), and a Software Engineer at Huawei (2011-2012).


• Chaoyang He has received a number of awards in academia and industry, including Amazon ML Fellowship (2021-2022), Qualcomm Innovation Fellowship (2021-2022), Tencent Outstanding Staff Award (2015-2016), WeChat Special Award for Innovation (2016), Baidu LBS Group Star Awards (2013), and Huawei Golden Network Award (2012).
A Surprisingly “Simple” Recipe for Modern NLP

Model + Labeled Data + Computing Power
A Surprisingly “Simple” Recipe for Modern NLP

Model + Labeled Data + Computing Power

pip install transformers
from transformers import BertModel
from transformers import RobertaModel

aws ec2 run-instances
  --instance-type p3.2xlarge
  --instance-type p3.16xlarge

"translate English to German: That is good."
"cola sentence: The course is jumping well."
"sstb sentence: The rhino grazed on the grass. sentence2: A rhino is grazing in a field."
"summarize: state authorities dispatched emergency crews tuesday to survey the damage after an onslaught of severe weather in mississippi."
"Das ist gut." "not acceptable" "3.8"
"six people hospitalized after a storm in attala county."
The “pre-train then fine-tune” paradigm for NLP

Randomly masked: A quick [MASK] fox jumps over the [MASK] dog
Predict: A quick brown fox jumps over the lazy dog

Embedding to vocab + softmax
Classification layer: Fully-connected layer + GELU + Norm
Transformer encoder

Downstream data
Transformer LMs
Task Model
Fine-tuning a pretrained transformer
What happen in workplace..

**Data-Centralized Learning**

- Collect training signals/examples from users and store them on a server.
- Fine-Tuning a pre-trained Transformer-based LM.
- Deploy the fine-tuned model for client users.

**Disadvantages of Data-Centralized Learning**
- User Privacy Concerns
- Data Sharing Regulation Laws
- High Cost of Transferring Raw Data
- Expensive Computation of Centralized Training
FedNLP

Federated Learning

Upload the updates of a local model
Download the updated global model
Private Local Data (*never exposed*)
Federated models for an NLP task.
Federated Learning (FL)

Taking \textit{FedAvg} as an example FL method

the local data is never exposed to others!
**Background:** Current FL research mainly focus on testing methods on toy datasets or computer vision (image classification).

**Our goal:** Provide a universal platform for benchmarking and developing FL methods for various NLP tasks.

Different NLP tasks have distinct task formulations.
• Most existing FL studies are customized for computer vision tasks/datasets, with specialized model architectures.

• Modern NLP are primarily based on pre-training & fine-tuning Transformer LMs.

• No existing FL framework connecting Transformer LMs with FL methods.
Creating non-IID datasets for FL in NLP is an open problem

- Current NLP datasets are mainly collected for *centralized* learning, and thus does not have a *natural, non-IID* partition

- Existing FL datasets are mainly for computer vision tasks such as object detection

- Ideal yet not accessible: private user data in large companies; but we cannot make them public for community use

**Data Partitioning Strategy**

Existing Centralized Dataset

A pool of clients where each client has a relatively unique data distribution.
What happen in workplace..

Three ways to create non-IID data partitions.

- Centralized Dataset
  - Dir. over # labels
  - + Sampling
  - Clustering w/ Input Features
  - Clients w/ different label distribution
  - Clients w/ different quantity of examples

- Dirichlet Allocation
  - N different label/feature/quantity distribution.
  - De-Centralized, non-IID clients.
We select four typical datasets for each formulation.

<table>
<thead>
<tr>
<th>Task</th>
<th>Txt.Cls.</th>
<th>Seq.Tag.</th>
<th>QA</th>
<th>Seq2Seq</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dataset</td>
<td>20News</td>
<td>Onto.</td>
<td>MRQA</td>
<td>Giga.</td>
</tr>
<tr>
<td># Training</td>
<td>11.3k</td>
<td>50k</td>
<td>53.9k</td>
<td>10k</td>
</tr>
<tr>
<td># Test</td>
<td>7.5k</td>
<td>5k</td>
<td>3k</td>
<td>2k</td>
</tr>
<tr>
<td># Labels</td>
<td>20</td>
<td>37*</td>
<td>N/A</td>
<td>N/A</td>
</tr>
</tbody>
</table>

We select four typical datasets for each formulation.

Three ways to create non-IID data partitions.
A showcase of non-IIDness on 20news dataset using *Dirichlet Allocation Methods*

- **Label distribution**: darker color
  - Higher probability for a client’s data assigned with a certain label

- Smaller alpha $\alpha$ more distinct label distributions between the clients (non-IID)

- When alpha=100 $\alpha$ uniform label distribution for all clients
Our Proposed FedNLP framework

- We support a wide range of FL methods such as FedAvg, FedOpt, FedNova, FedProx, etc.
- We implement the interface between these FL methods to Transformer LMs.
  - We support fine-tuning the full model as well as part of the models (last few layers).
  - All task formulations.

Algorithm 1: FedOPT (Reddi et al. 2020)): A Generic FedAvg Algorithm

```
Input: Initial model $x^{(0)}$, CLIENTOPT, SERVEROPT

for $t \in \{0, 1, \ldots, T - 1\}$ do
  Sample a subset $S^{(t)}$ of clients
  for client $i \in S^{(t)}$ in parallel do
    Initialize local model $x_{i}^{(t,0)} = x^{(t)}$ Download from server.
    for $k = 0, \ldots, \tau_i - 1$ do
      Compute local stochastic gradient $g_i(x_{i}^{(t,k)})$
      Perform local update $x_{i}^{(t,k+1)} = \text{CLIENTOPT}(x_{i}^{(t,k)}, g_i(x_{i}^{(t,k)}), \eta, t)$
    Compute local model changes
    $\Delta_{i}^{(t)} = x_{i}^{(t,\tau_i)} - x_{i}^{(t,0)}$ Upload to server.
  Aggregate local changes
  $\Delta^{(t)} = \sum_{i\in S^{(t)}} p_i \Delta_{i}^{(t)} / \sum_{i\in S^{(t)}} p_i$
  Update global model
  $x^{(t+1)} = \text{SERVEROPT}(x^{(t)}, -\Delta^{(t)}, \eta_s, t)$ On the server.
```
The FedNLP Framework
Q1: How do popular FL methods perform over different NLP tasks?

- FedOpt outperforms the other two FL methods in the first three tasks.
- FedProx and FedAvg are comparable with each other.

Y-axis: evaluation metric of the task. Higher the better.
X-axis: # of iterations in the algorithm
Q2: How do different non-IID partitions of influence FL performance?

- Non-IID data partition (smaller alpha) creates more challenges for FL methods to perform
- Non-IID data partition (smaller alpha) also makes the FL algorithm less stable
- Uniform and quantity-skew partitions are less challenging to learn

Figure 5: Testing FedOPT with DistilBERT for 20News under different data partition strategies.
Q3: How does freezing of Transformers influence the FL performance?

Prediction

Classifier

Transformer Layer 6 (DistilBERT)

Transformer Layer 2

Transformer Layer 1

[CLS] I like to draw [SEP]

which layers are frozen and NOT used for weight-uploading/downloading.

Frozen Layers

E
E+L₀
E+L₀₋₁
E+L₀₋₂
E+L₀₋₃
E+L₀₋₄
E+L₀₋₅

Figure 6: Testing FedOPT with DistilBERT for 20News under different frozen layers.
Q4: Are compact model DistilBERT adequate for FL+NLP?

- BERT-base is 2x larger than DistilBERT
- DistilBERT is a more cost-effective choice.
- It’s reasonable to do experiments with DistilBERT as the curve is similar to BERT-base.

Figure 7: FedOPT for 20News with different LMs.
Task 1 [heterogenous FedNLP]: Current FL methods focus on the case where all local models are of the same architecture and model size. This is inflexible and can cause problems when users have different devices.

*How should we perform FL when clients are using different BERT-style architecture?*

Task 2 [privacy]: Concerns about Transformer LMs that can memorize private information.

*Can we quantitively measure such data leak? Can we design methods to prevent this?*