FedSum: Data-Efficient Federated Learning Under Data Scarcity Scenario For Text Summarization

Zhiyong Ma[1*], **Zhengping Li**[1*], **Yuanjie Shi**[2], **Jian Chen**[1†]

[1]South China University of Technology
[2]Washington State University
seallen97@mail.scut.edu.cn, lievan20022@gmail.com, yuanjie.shi@wsu.edu, ellachen@scut.edu.cn

Abstract

Text summarization task extracts salient information from a large amount of text for productivity enhancement. However, most existing methods heavily rely on training models from ample and centrally stored data which is infeasible to collect in practice, due to privacy concerns and data scarcity nature under several settings (e.g., edge computing or cold starting). The main challenge lies in constructing the privacy-preserving and well-behaved summarization model under the data scarcity scenario, where the data scarcity nature will lead to the knowledge shortage of the model while magnifying the impact of data bias, causing performance degeneration. To tackle this challenge, previous studies attempt to complement samples or improve the efficiency of data. The former is usually associated with high computing costs or has a large dependence on empirical settings, while the latter might not effective due to the lack of consideration of data bias. In this work, we propose FedSum which extends the standard FL framework from depth and breadth to further extract prime and diversified knowledge from limited resources for text summarization. For depth extension, we introduce a Data Partition method to cooperatively recognize the prime samples that are more significant and unbiased, and the Data skip mechanism is introduced to help the model further focus on those prime samples during the local training process. For breadth extension, FedSum extends the source of knowledge and develops the summarization model by extracting knowledge from the data samples, hidden spaces, and globally received parameters. Extensive experiments on four benchmark datasets verify the promising improvement of FedSum compared to baselines, and show its generalizability, scalability, and robustness .

Code — https://github.com/Li-Evan/FedSum

Introduction

The amount of text data has grown explosively in various domains, such as journalism, medicine, and entertainment. For productivity enhancement, the summarization model, namely summarizer, compresses textual content into shorter versions while retaining key concepts from input content. To construct a summarizer, most existing literature focused on the centralized manner with ample data. However, the data from individuals or institutions are generally private and sensitive, prohibiting access from the public. This makes the collection of text data in a central location infeasible. Federated Learning (FL) (McMahan et al. 2017) provides a paradigm to utilize information and construct a shared model securely. Nevertheless, when the FL system is plagued by data scarcity (e.g., under edge computing or cold starting settings, where each FL client trains a common and basic summarization model with over 10M parameters with less than 500 samples), it is intractable for the summarizer to get adequate knowledge (Lin et al. 2022) and vulnerable to data bias, such as Leading Bias (Zhong et al. 2019; Ko et al. 2021), resulting in performance degradation. Then a natural question is: *How to derive a well-behaved summarizer in data scarcity FL?*

Generally, data scarcity leads to the knowledge shortage of the summarizer, which is the main reason for degeneration (Cai et al. 2023b). To tackle this challenge, complementing samples (Zhang et al. 2018; Cubuk et al. 2019) or improving the efficiency of data (Zhuang et al. 2020; Li et al. 2021) are two common-used strategies. The former creates samples by applying various transformations to the original data (Yoon et al. 2021; Zhou and Konukoglu 2023), which generally is computation-costed and heavily relies on the hyperparameters setting (Xu, Lin, and Wang 2023; Duan et al. 2023). The latter focuses on exploiting limited resources (Collins et al. 2021; Lu et al. 2023), such as utilizing the distance between measured prototypes and exploring parameters. Although these methods further explore the limited data, the improvement is not remarkable in the practice, due to the neglect of data bias.

To find an effective solution for the degeneration, we propose to maximize the data efficiency in model training. We study the strategy of sample weighting, and mining hidden knowledge from different aspects. We propose FedSum which extends the standard FL framework from depth and breadth to further extract prime and diversified knowledge for constructing the summarizer, as illustrated in Fig. 1.

For depth extension, inspired by hard sample mining and bias elimination (Chen et al. 2023a), we introduce the Data Partition method to recognize the prime samples, and adjust the weight of data by the proposed Data skip mechanism,

*These authors contributed equally.
† Corresponding Author.

to further mining prime knowledge. Specifically, we refer to the samples with more unbiased and higher loss samples as prime samples, while pronounced data bias and low supervised loss as normal samples. Since normal samples account for a large proportion in the dataset (Zhong et al. 2019), the summarizer will easily ignore the prime sample, degrading the generalization (Xing, Xiao, and Carenini 2021), especially under data scarcity. To address it, FedSum dynamically discards part of normal samples based on the training progress, but not all normal samples like the common solutions (Lin et al. 2018), preventing further degradation.

For breadth extension, inspired by Multi-Task learning (Marfoq et al. 2021; Cai et al. 2024), we extend the source of knowledge in training. FedSum not only learns from data but also from hidden spaces and parameters. Since the prototype is generally regarded as the carrier of semantic information in hidden space (Li et al. 2020a), we leverage different prototypes to build prototype loss, improving the generalization and discrimination of features. Then, FedSum constructs the semantic portraits for FL clients by their specific prototypes, and measures the semantic distances between them to maintain the Portrait Distance Table (PDT) on the server. Take the PDT as a guideline, each client can supplement the insufficient semantic knowledge of their representation model by training with the globally received classification heads.

Finally, we evaluate our method on benchmark datasets, showing that FedSum provides a promising total improvement over baselines in ROUGE metrics (0.15% at least and 29.9% at most) and exhibits generalization under various heterogeneity (fluctuation in ROUGE metrics $\leq$ 2.6% on CNNDM and $\leq$ 0.3% on PubMed), scalability about data quantity over FL system, and robustness to leading bias (that the position distribution of FedSum's prediction is more even and closer to the Oracle). Our main contributions are concluded as follows:

- We propose FedSum, a privacy-preserved text summarization framework that maximizes the data efficiency by mining knowledge from samples, hidden spaces, and received parameters for the challenge of data scarcity.
- To mitigate the negative effect of data bias magnified by data scarcity, we propose the Data Partition method and the Data skip mechanism for further mining the prime knowledge in model training.
- Extensive experimental results on benchmark datasets verify the improvement of FedSum and verify its generalization for various heterogeneous conditions, its scalability in data quantity, and its robustness to leading bias.

Background

Extractive Text Summarization

Since extracting the key idea from abounding information is valuable in various scenes, many works have been proposed in this track (Zhang, Liu, and Zhang 2023; Park et al. 2024). Inspired by the success of BERT, a summarizer with an enlightening pattern has been proposed (Liu and Lapata 2019), namely BERTSUM. It leverages BERT to represent the input content and then recognizes the most salient sentences by classification modules. Following this pattern, a series of works modify the model architecture to further explore the semantic and structural information (Cohan et al. 2020; Bi et al. 2021). Another branch of work builds contrastive frameworks to reinforce BERTSUM by re-ranking the model result (Zhong et al. 2020) or modifying the learning object (Liu and Liu 2021).

Federated Learning

Federated Learning is a rising paradigm in privacy-preserving. A surge of works explore diverse FL applications in NLP (Liu et al. 2021; Du et al. 2023). Although FL has strength in protecting privacy, it suffers from degradation problems, caused by heterogeneity and data scarcity. To alleviate the degradation due to heterogeneity, FedProx (Li et al. 2020b) introduces a proximal term to restrain the local drift. SCAFFOLD (Karimireddy et al. 2020) learns personalized control variates referring directions of global updates to guide the local training. FedNova (Wang et al. 2020) introduces a normalizing weight to eliminate the objective inconsistency and stabilize convergence. In the data scarcity scenario, like the cold start scenario, models trained by these methods might have difficulty acquiring sufficient knowledge. One intuitive idea is extracting knowledge from hidden space (Huang et al. 2023; Zhang et al. 2024). FedProto (Tan et al. 2022) extends the concept of prototype learning to FL, reaching feature-wise local alignment with the global prototype. Another branch of methods tries to leverage the extensive knowledge from parameters. FedDC (Kamp, Fischer, and Vreeken 2021) interleaving model aggregation and permutation steps. During a permutation step, it redistributes local models across clients through the server. These above methods mitigate the negative effects of data heterogeneity and scarcity, but not always behave outstanding in the summarization task, due to the neglect of data bias.

Task Definition and Notations

Given a document (data sample) $d = \{u_1, \cdots, u_S\}$ with S sentences from dataset $D = \{d_1, \cdots, d_n\}$, each sentence is assigned a result $\hat{\phi} \in [0, 1]$ through the model (summarizer), representing the probability that the sentence should be extracted. General extractive summarizer $f(\Omega, \cdot)$ parametered by $\Omega = \{\theta, \omega\}$ can be divided into the representation model $r(\theta, \cdot)$ and the classification head $h(\omega, \cdot)$, where $f(\Omega, \cdot) = h(\omega, r(\theta, \cdot))$. Define $\ell(\widehat{y}_\Omega, y) = CE(\widehat{y}_\Omega, y)$ as the cross entropy loss function measured on the inference $\widehat{y}_\Omega = f(\Omega, d) = [\hat{\phi}_1, \cdots, \hat{\phi}_S]$ of document d and ground-true label sequence $y = [\phi_1, \cdots, \phi_S]$, where $\phi_{(\cdot)} \in \{0, 1\}$.

Given a client id k, we denote $\widehat{Y}_{\Omega_k}$ and Y_k as its prediction set and ground-true set, respectively. $D_{(k,F)}$ and π_k are full dataset and its distribution. $D_{(k,\gamma)}$ denotes the γ subset of $D_{(k,F)}$. In FL literature, it is a heterogeneous setting if $\pi_p \neq \pi_q$ for $p \neq q$. C and K stand for the whole client set and its size, α is the active ratio. C_α and $\{C_\alpha\}$ denote the subset of active clients and its indexes. T and E are the number of communications and local epochs. $\mathcal{B}$ denotes data batch. B and η are the batch size and the learning rate. n_k and $\tau_k =$

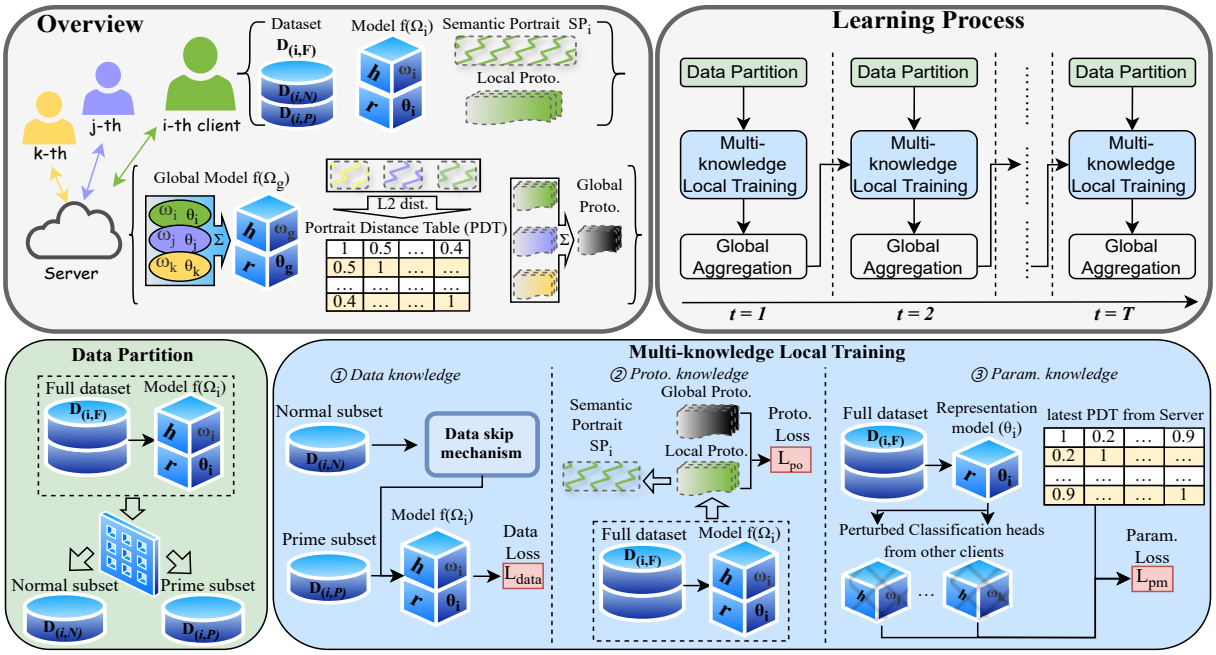

Figure 1: To mitigate the degradation caused by data scarcity, FedSum extends the standard FL framework from depth and breadth. For depth extension, FedSum distinguishes the prime and normal data by the Data Partition method and adjusts the weight of normal samples through the Data skip mechanism, promoting the summarizer to further focus on the more significant and general pattern. For breadth extension, FedSum tries to maximize the efficiency of data by extending the source of knowledge through Multi-knowledge Local Training, which learns from data, hidden space, and globally received parameters.

$\lceil n_k/B \rceil$ stand for the number of local data samples from k-th client and its number of local updates. The total number of samples is $n = \sum_{k=1}^{K} n_k$. In FL, one aims to learn a model that minimizes the empirical loss $\widehat{L}(\Omega)$ as follows:

$$\min_{\Omega} \widehat{L}(\Omega) := \sum_{k=1}^{K} \frac{n_k}{n} \widehat{L}_k(\Omega_k, D_{(k,F)}). \tag{1}$$

where $\hat{L}_k(\Omega_k, D_{(k,F)}) := \sum_{d_i}^{D_{(k,F)}} \frac{1}{n_k} \ell(f(\Omega_k, d_i), y_i)$.

Method

The data scarcity leads to the knowledge shortage problem, affecting the performance of the summarizer. To address this problem by maximizing the data efficiency, we propose FedSum with two innovative modules: Data Partition and Multi-knowledge Local Training, for depth and breadth extensions, as concluded in Alg.1 and Alg. 2. The workflow is below.

Firstly, the server initializes the parameter. During the iteration of communication, the server obtains the communication list and broadcasts the parameter to each active client (see line 1 to 4 in Alg. 1). After that, each client performs the Data Partition based on the deviation of samples' bias level and supervised significance between local and global (see line 4 to 5 in Alg. 2). Then each client performs Multi-knowledge Local Training. During the exploration of data, the Data skip mechanism discards partial normal samples (see line 6 to 11 in Alg. 2). After the first communication round, based on the semantic portrait distance between users, FedSum also extracted semantic knowledge from parameters to update the representation model (see line 12 to 21 in Alg. 2). When local training epoch is done, FedSum tries to optimize the feature representations in generalization and discrimination by measuring different prototypes and proto loss (see line 22 to 24 in Alg. 2). After Multi-knowledge Local Training, the server updates the PDT and corresponding parameters according to the uploaded resources (see line 5 to 13 in Alg. 1). Finally, the server aggregates different resources to complete one iteration (see line 14 to 16 in Alg. 1). In short, FedSum optimizes the representation module by minimizing $L_{data}+L_{po}+L_{pm}$, while optimizing the classification head by minimizing L_{data}.

Data Partition

Similar to recognizing noisy samples (Li et al. 2024), data can be allocated to normal and prime subsets according to their significance for better exploration. In ML, the sample with large training loss is generally considered as a prime sample (Wang et al. 2023), which can reflect more guidance about the task. Considering the characteristics of ExtSum, that the extractive summarization can be decomposed to the classification of multiple sentences (Bidoki, Moosavi, and Fakhrahmad 2022), we propose that the criterion for the prime sample should be detailed down to the sentence level. Since the amount of key sentences is much less than the ordinary, the loss of sentences labeled with "1" is more valuable, which can precisely reflect the dilemma in classification. Meanwhile, the label distribution can intuitively reflect

Algorithm 1: FedSum Server.

Require: T, K, α, η, γ

Initialize $\theta_g^{(0)}, \omega_g^{(0)}$, set $\bar{\epsilon}^{(0)}$ and $\bar{Q}^{(0)}$ as 0
for $t = 1$ to T **do**
  $C_\alpha \leftarrow$ Client Selection(K, α)
  Sever Broadcast(C_α, $\theta^{(t-1)}$, $\omega^{(t-1)}$, $\bar{\epsilon}^{(t-1)}$, $\bar{Q}^{(t-1)}$)
  for $i \in \{C_\alpha\}$ **in parallel do**
    $Q_{(i,j)}^{(e,t)}, \epsilon_{(i,j)}^{(e,t)}, \theta_i^{(t-1)}, \omega_i^{(t-1)}, P_{(i,c)}^{(\xi,t-1)}, SP_i \leftarrow$ FedSum Client ($\bar{\epsilon}^{(t-1)}$, $\bar{Q}^{(t-1)}$, t, PK_i, $\bar{P}_c^{(\xi,t)}$)
  end for
  Client Upload(C_α)
  for $i \in \{C_\alpha\}$ **do** *# Update PDT by $\|\cdot\|_2$ and $tanh(\cdot)$*
    $PDT[i][p] = tanh(\|SP_i - SP_p\|_2)$
    $\tilde{\omega}_i^{(t-1)}$ = Dropout($\omega_i^{(t-1)}$, γ) *# Bernoulli Noise*
    PK_i = {(PDT[i][p], $\tilde{\omega}_i^{(t-1)}$) | $p \in \{C_\alpha\} \setminus \{i\}$}
  end for
  $\theta_g^{(t)} = \sum_i^{\{C_\alpha\}} \frac{n_i}{n} \cdot \theta_i^{(t-1)}, \omega_g^{(t)} = \sum_i^{\{C_\alpha\}} \frac{n_i}{n} \cdot \omega_i^{(t-1)}$
  $\bar{\epsilon}^{(t)}, \bar{Q}^{(t)} \leftarrow$ Eq. 2, $\bar{P}_c^{(\xi,t)} = \sum_i^{\{C_\alpha\}} \frac{n_i}{n} \cdot P_{(i,c)}^{(\xi,t)}$
end for
return $\theta_g^{(T)}, \omega^{(T)}$

Algorithm 2: FedSum Client (i-th client).

Require: t, PK_i, $\bar{P}_c^{(\xi,t)}$

for $e = 1$ to E **do**
  for $j = 1$ to τ_i **do**
    $\mathcal{B}_j \leftarrow$ Sampling ($D_{(i,F)}$), $LD \leftarrow$ Label($\mathcal{B}_j$)
    $H_j = r(\theta_i^{(t-1)}, \mathcal{B}_j)$, $LM = \ell(h(\omega_i^{(t-1)}, H_j), LD)$
    $Q_{(i,j)}^{(e,t)}, \epsilon_{(i,j)}^{(e,t)}, D_{(i,N)}, D_{(i,P)} \leftarrow$ Data Partition(LD, LM, m, λ, $\bar{\epsilon}^{(t-1)}$, $\bar{Q}^{(t-1)}$, $D_{(i,N)}$, $D_{(i,P)}$)
    $\hat{\rho} \leftarrow$ Sampling ($U(0,1)$), $\rho_{(i,j)} = \frac{t}{T} \cdot \frac{e}{E} \cdot \frac{j}{\tau_i}$
    $L_{data} = \sum_b^B \sum_c^m \frac{LM[b][c]}{B}$ *# Data knowledge*
    if $\mathcal{B}_j \in D_{(i,N)}$ and $\hat{\rho} \leq \rho_{(i,j)}$ **then**
      $L_{data} = 0$ *# Data skip mechanism*
    end if
    $L_\theta := L_\theta + L_{data}$, $L_\omega := L_\omega + L_{data}$
    if $t \neq 1$ **then** *# Param. knowledge*
      for $(PDT[i][p], \tilde{\omega}_i^{(t-1)})$ in PK_i **do**
        $\zeta_p = PDT[i][p]$
        $L_{pm} = \zeta_p \cdot \sum_b^B \sum_c^m \frac{\ell(h(\tilde{\omega}_p^{(t-1)}, H_j), LD)[b][c]}{B}$
        $L_\theta := L_\theta + L_{pm}$
      end for
    end if
    $\theta_i^{(t-1)} := \theta_i^{(t-1)} - \eta \nabla L_\theta$, $\omega_i^{(t-1)} := \omega_i^{(t-1)} - \eta \nabla L_\omega$
  end for
end for
$P_{(i,c)}^{(\xi,t-1)} \leftarrow$ Eq. 3, $P_{(i,p)}^{(\xi,t-1)} \leftarrow$ Eq.4, $SP_i \leftarrow$ Eq. 6.
if $t \neq 1$ **then** *# Proto. knowledge*
  $L_{po} \leftarrow$ Eq. 5, $L_\theta = L_{po}$, $\theta_i^{(t-1)} := \theta_i^{(t-1)} - \eta \nabla L_\theta$
end if
return $Q_{(i,j)}^{(e,t)}, \epsilon_{(i,j)}^{(e,t)}, \theta_i^{(t-1)}, \omega_i^{(t-1)}, P_{(i,c)}^{(\xi,t-1)}, SP_i$

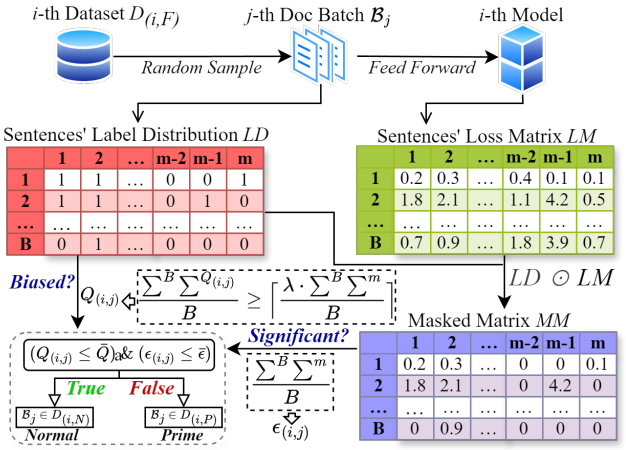

Figure 2: The workflow of Data Partition method. m is the max sentence index in data batch. More unbiased and higher-loss samples will be allocated to the prime subsets.

the degree of leading bias in the data batch. Thus we refer to the samples with pronounced leading bias and low supervised loss as normal samples, while more unbiased and higher loss samples as prime samples. We can recognize the prime samples based on the measurements from the label distribution and loss matrix. For privacy constrain, the data partition should be carried out locally. However, since the private samples for each client are limited, the judgment only based on local information might be biased (Tang et al. 2021). Thus, we propose a collaborative Data Partition algorithm in Alg. 3 , which partitions local data regarding the globally bias and significant level, as Fig. 2 shown. The detail of Alg. 3 can be found in the appendix.

Specifically, $Q_{(i,j)}^{(e,t)}$ index the position where the percentage of key sentences in j-th data batch $\mathcal{B}_j$ over λ, reflecting the degree of leading bias in the sentences' Label Distribution LD. The LD and sentences' Loss Matrix LM, measured by i-th model Φ_i and $\mathcal{B}_j$, are utilized to construct the Masked Matrix MM by Hadamard product. The average loss $\epsilon_{(i,j)}^{(e,t)}$ over MM at e-th epoch in t-th communication round reflects the significance of $\mathcal{B}_j$. Comparing the local statistics, $Q_{(i,j)}^{(e,t)}$ and $\epsilon_{(i,j)}^{(e,t)}$, with the global thresholds $\bar{\epsilon}^{(t-1)}$ and $\bar{Q}^{(t-1)}$ from the last communication, the data partition method can divide $\mathcal{B}_j$ into either normal $D_{(i,N)}$ or prime subset $D_{(i,P)}$, where:

$$\bar{\epsilon}^{(t)} = \sum_i^{\{C_\alpha\}} \sum_e^E \sum_j^{\tau_i} \frac{n_i}{n} \cdot \frac{\epsilon_{(i,j)}^{(e,t)}}{E \cdot \tau_i}, \quad \bar{Q}^{(t)} = \sum_i^{\{C_\alpha\}} \sum_e^E \sum_j^{\tau_i} \frac{n_i}{n} \cdot \frac{Q_{(i,j)}^{(e,t)}}{E \cdot \tau_i}. \tag{2}$$

Multi-knowledge Local Training

For the lack of knowledge problem caused by data scarcity, FedSum maximizes data efficiency by mining diversified knowledge from three aspects: data, hidden space, and received parameters. For better-exploring data, FedSum extends the standard local training with the Data skip mechanism to discard partial normal samples, promoting the model further focus on the prime samples. To extract knowledge from hidden space, FedSum takes the divergence between local and global prototypes as regularization constraints to promote the generalization and discrimination of feature representation. Mining the missing semantic knowledge from globally received parameters, FedSum takes the semantic portrait distances as a guideline to train the representation model with perturbed classification heads.

Data knowledge and Data skip mechanism. Since the proportion of normal samples is more than the proportion of prime, it's common for the model to ignore those prime data, threatening the generalization (Kawaguchi and Lu 2020; Xing, Xiao, and Carenini 2021). To improve performance, a common solution in ML studies is to discard normal data (Mindermann et al. 2022; Kaddour et al. 2024). Due to data scarcity, FedSum does not discard all normal samples to avoid exacerbating degradation. Instead, we propose the Data skip mechanism to skip normal samples according to the skipping probability $\rho_{(i,j)}$ which increases with training progress. As the training progresses, the summarizer will tend to learn more normal patterns than the prime. To enhance the generalization, we skip partial normal samples at the later stage, so that the summarizer can more focus on the samples with greater guidance value. Given j-th batch $\mathcal{B}_j$ from i-th client belongs to normal subset $D_{(i,N)}$, the skipping probability $\rho_{(i,j)} = \frac{t}{T} \cdot \frac{e}{E} \cdot \frac{j}{\tau_i}$ is determined by the progress in communication, epoch, and local update. Randomly sampling $\hat{\rho}$ from uniform distribution $U(0,1)$, if $\hat{\rho} \leq \rho_{(i,j)}$, the $\mathcal{B}_j$ be skipped and $L_{data} = 0$. Otherwise, the $\mathcal{B}_j$ be retained and $L_{data} = \sum_{(d,y)}^{\mathcal{B}_{(j)}} \sum_{c}^{m} \frac{\ell(f(\Omega_i,d)[c],y)}{B}$.

Proto. knowledge. To extract knowledge from hidden space, we take the prototype as the carrier. Following the definition of the class prototype presented in literature (Tan et al. 2022; Chen et al. 2020, 2023b), we denote the class ξ prototype of $D_{(i,F)}$ in t-th communication round as $P_{(i,c)}^{(\xi,t)}$, which take the average over hidden representation of sentences belong to class $\xi \in \{0,1\}$:

$$P_{(i,c)}^{(\xi,t)} = \sum_{\{d,[\phi_1,\cdot,\phi_S]\}}^{D_{(i,F)}} \sum_{s=1}^{S} \frac{r(\theta_i^t,d)[s] \cdot \mathbb{1}(\phi_s = \xi)}{n_{(i,c)}^{\xi}}, \quad (3)$$

where $n_{(i,c)}^{\xi} = \sum_{d}^{D_{(i,F)}} \sum_{s=1}^{S} \mathbb{1}(\phi_s = \xi)$, and $\mathbb{1}(\cdot)$ denote the indicator function. Extending the concept of class prototype, we measure the prediction prototype $P_{(i,p)}^{(\xi,t)}$ by the average over the hidden representation, in which the soft predictions of these representations are predicted as class ξ:

$$P_{(i,p)}^{(\xi,t)} = \sum_{d}^{D_{(i,F)}} \sum_{s=1}^{S} \frac{r(\theta_i^t,d)[s] \cdot \mathbb{1}(\hat{\phi}_s = \xi)}{\hat{n}_{(i,p)}^{\xi}}, \quad (4)$$

where $\hat{n}_{(i,p)}^{\xi} = \sum_{d}^{D_{(i,F)}} \sum_{s=1}^{S} \mathbb{1}(\hat{\phi}_s = \xi)$. Leveraging these prototypes, we build the prototype loss L_{po} with two terms. One term aims to mitigate representation differences between the local and global prototypes, improving the generalization of features. Another aims to discriminate features, promoting the model to fit better decision boundaries:

$$L_{po} = \underbrace{\sum^{\xi \in \{0,1\}} \|\bar{P}_c^{(\xi,t-1)} - P_{(i,c)}^{(\xi,t)}\|_2}_{Generalizable} \underbrace{- \|P_{(i,p)}^{(1,t)} - P_{(i,p)}^{(0,t)}\|_2}_{Discriminative}, \quad (5)$$

where $\bar{P}_c^{(\xi,t-1)}$ is aggregated as $\bar{P}_c^{(\xi,t)} = \sum_{i}^{\{C_\alpha\}} \frac{n_i}{n} \cdot P_{(i,c)}^{(\xi,t)}$. After prototype measurement, each client can construct their semantic portrait SP_i by concatenating class prototypes:

$$SP_i = Concat(P_{(i,c)}^{(0,t-1)}, P_{(i,c)}^{(1,t-1)}). \quad (6)$$

The semantic portrait of each participant will be uploaded under privacy constrain, and utilized to maintain the Portrait Distance Table (PDT) by measuring the L2 distance between each portrait and scaling by $tanh$ function in the FL server, where the PDT can be initialized randomly or according to the initial semantic portrait provided by clients.

Param. knowledge. The semantic knowledge contained in the local model is closely related to the client's data distribution. Different models might have different semantic knowledge deficiencies in practice. To fix the semantic deficiencies in the local model, FedSum extract knowledge from received parameters. FedSum uses PDT to guide the mining process about received parameters and build the parameter lost L_{pm}, due to nature of PDT in intuitively reflecting the degree of semantic gap between different clients. Besides, to avoid catastrophic forgetting and protect privacy, the received classification modules are perturbated by Bernoulli noise controlled by hyperparameter γ in advance on server. For measuring L_{pm}, we feed the representation of data batch H_j to the received p-th client's classification modules $\tilde{\omega}_p^{(t-1)}$, then weights the training loss with $\zeta_p = PDT[i][p]$:

$$L_{pm} = \zeta_p \cdot \sum_{b}^{B} \sum_{c}^{m} \frac{\ell(h(\tilde{\omega}_p^{(t-1)}, H_j), LD)[b][c]}{B} \quad (7)$$

Experiments

Experimental Setup

Baselines and Measurement. We investigate the milestone model, BERTSUM, in FL experiments. Several representative algorithms are adopted as baselines: FedAvg (McMahan et al. 2017), FedProx (Li et al. 2020b), SCAFFOLD (Karimireddy et al. 2020), FedNova (Wang et al. 2020), FedProto (Tan et al. 2022), and FedDC (Kamp, Fischer, and Vreeken 2021). To observe the effect of global aggregation, we also adopt the Separate method, where each client only trains their model locally. To measure the quality

of the output summary, we apply the commonly used overlap metric ROUGE (Lin 2004). The ROUGE-F1 and ROUGE-Recall metrics are uniformly denoted as R-F(1/2/L) and R-R(1/2/L) respectively. We report the most optimal testing records (with better R-R-1 values) among different communication rounds for each algorithm. Since the performance fluctuation caused by various random factors is inevitable, we report the average of the results over 3 repeated experiments to avoid unfair comparison in practice.

Datasets and Distributions. We built different test beds on common benchmark datasets, such as CNN/DailyMail (Nallapati, Zhai, and Zhou 2017), WikiHow(Koupaee and Wang 2018), Reddit(Kim, Kim, and Kim 2019), and PubMed(Cohan et al. 2018). To simulate the data scarcity scenarios, only 2K training samples can be accessed by the FL system. To simulate the non-IID setting, we construct the quantity skew following (Li et al. 2022; Cai et al. 2023a) and control the heterogeneity levels through Dir. Smaller Dir indicates a more imbalanced scenario about clients' data quantity, and $Dir = +\infty$ represents the uniform case.

Experiment Result

We summarize the main experimental results in Tables 1 to 2. Due to the limitation of pages, we report the experimental results of two datasets in the main body. More detailed results about four datasets are supplemented in the Appendix. Overall, FedSum obtains significant improvement in performance over baselines. The most obvious improvements compared to FedAvg are $1.6\%/1.31\%/1.52\%$ in R-F and $2.18\%/1.82\%/2.05\%$ in R-R on CNNM. Under data heterogeneity, comparing FedSum with the optimal baseline, there exists 0.15% total improvement at least, and 29.9% total improvement at most over ROUGE metrics on two datasets. These results demonstrate the effectiveness of FedSum.

CNNDM	Data Heterogeneity	
	R-F(1/2/L)	R-R(1/2/L)
Separate	32.21/11.52/25.38	39.7/14.8/31.35
FedAvg	34.11/12.87/27.17	42.8/16.92/34.15
FedProx	35.32/13.85/28.44	44.64/18.38/35.99
Scaffold	30.96/10.33/24.13	37.76/13.13/29.5
FedDC	32.91/11.87/25.98	40.82/15.43/32.29
FedNova	33.39/12.47/26.48	41.23/16.07/32.77
FedProto	31.04/10.54/24.18	37.72/13.34/29.47
FedSum	**35.71/14.18/28.69**	**44.98/18.74/36.2**

Table 1: Experimental results of different algorithms under the data heterogeneous setting ($Dir = 0.1$) on CNNDM.

PubMed	Data Heterogeneity	
	R-F(1/2/L)	R-R(1/2/L)
Separate	31.01/10.76/27.8	32.73/11.13/29.32
FedAvg	31.1/10.81/27.9	32.85/11.18/29.44
FedProx	31.29/11.0/28.07	33.05/11.38/29.63
Scaffold	31.08/10.79/27.87	32.83/11.19/29.42
FedDC	31.08/10.79/27.87	32.83/11.19/29.42
FedNova	30.76/10.51/27.55	32.43/10.86/29.01
FedProto	30.72/10.41/27.51	32.41/10.75/28.99
FedSum	**31.48/11.2/28.26**	**33.3/11.62/29.86**

Table 2: Experimental results of different algorithms under the data heterogeneous setting ($Dir = 0.1$) on PubMed.

Generalization for heterogeneous condition. A ROUGE comparison under different heterogeneous levels is summarized in Table 3. The performance of FedSum fluctuates slightly ($\leq 2.6\%$ on CNNDM) as Dir increases. When Dir is 8, the FedSum ranks second with small gaps from the highest ($\leq 1.5\%$), verifying the generalization of FedSum under different heterogeneous.

CNNDM	R-R(1/2/L)	
	$Dir = 0.1$	$Dir = 0.5$
Separate	39.7/14.8/31.35	41.56/16.05/33.02
FedAvg	42.8/16.92/34.15	40.81/15.75/32.39
FedProx	44.64/18.38/35.99	38.36/13.33/29.96
Scaffold	37.76/13.13/29.5	39.76/14.85/31.4
FedDC	40.82/15.43/32.29	38.17/13.3/29.82
FedNova	41.23/16.07/32.77	38.48/13.34/29.92
FedProto	37.72/13.34/29.47	37.8/13.38/29.53
FedSum	**44.98/18.74/36.2**	**42.3/16.59/33.71**
	$Dir = 1$	$Dir = 8$
Separate	39.34/14.43/31.0	37.53/12.99/29.23
FedAvg	41.62/16.15/33.07	42.66/17.07/34.17
FedProx	42.3/16.5/33.7	42.95/17.13/34.35
Scaffold	39.47/14.64/31.21	41.93/16.5/33.51
FedDC	42.61/16.94/33.99	**45.63/19.17/36.86**
FedNova	40.57/15.53/32.13	41.41/16.19/32.93
FedProto	37.62/13.22/29.37	37.84/13.42/29.59
FedSum	**44.68/18.4/35.96**	44.42/18.01/35.62

Table 3: Experimental results with different heterogeneous settings on CNNDM. In general, the performance of FedSum outperforms baselines in most heterogeneous cases.

Robustness to leading bias. To verify the efficacy of tackling the degeneration caused by leading bias, we compare the position of the exacted sentences in the summaries generated between FedSum, FedAvg, and Oracle, following the setting of Liu and Lapata (2019), where the Oracle stands for the ground-true summaries. As shown in Fig. 3, the distribution of the extracted proportion about FedSum is more similar to the same of Oracle than FedAvg. The extracted result of FedAvg contains more leading bias than FedSum, shown by a more right-skewed histogram for FedSum.

Scalability in data quantity. To explore the scalability of FedSum under varying degrees of data scarcity over the FL system, the performance trends of FedSum and FedAvg with different data quantities are illustrated in Fig. 4. Generally, the performance of FedSum becomes better as the data quantity enlarges, demonstrating the scalability in data quantity.

Effect of λ. Hyperparameter λ controls how tolerable the Data Partition method is to data leading bias. Smaller λ leads

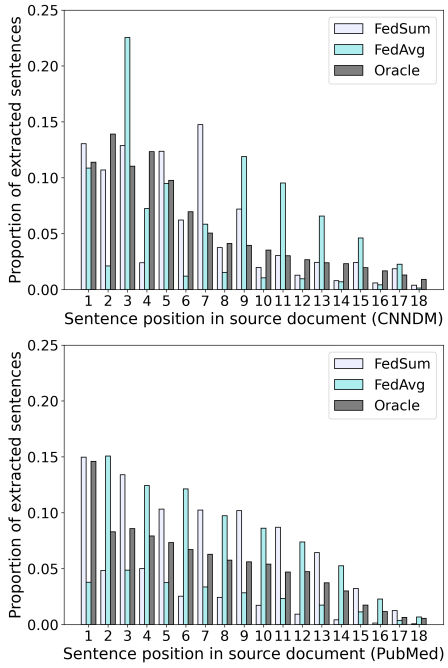

Figure 3: The proportion of extracted sentences according to their position in the original document. Measuring FedSum and FedAvg in the total absolute deviation to Oracle, the deviation of FedAvg is around 69.31% higher than FedSum. and 15.91% in PubMed.

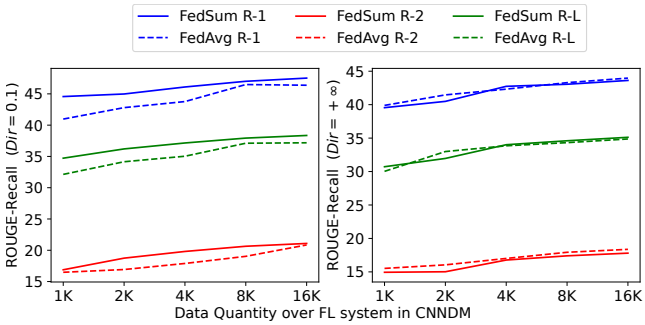

Figure 4: The performance of two methods with different data quantities. In the heterogeneous setting, three solid lines (FedSum) behave better than the dotted lines with the same colors (FedAvg). Under uniform, FedSum is inferior to FedAvg when the quantity is less than 4K, but the gaps between two methods become narrowed as the quantity increases.

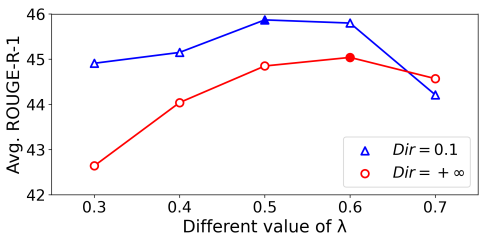

Figure 5: ROUGE scores of FedSum with different λ in CNNDM. The ROUGE is optimal when λ equals around 0.5.

to smaller $Q^{(e,t)}_{(i,j)}$ for looser restrictions. We adjust λ in heterogeneity and uniform scenarios on CNNDM, as shown in Fig. 5. As λ increases from 0.3 to 0.6, the restrictions on leading bias become stronger, and ROUGE go better. When λ over 0.6, the restrictions of leading bias in dataset are too strict and cause degeneration. The above results demonstrate the efficacy in mitigating the negative effect of leading bias.

Ablation. To evaluate the effect of two extensions, we take FedAvg as the reference and measure the improvement of our methods, as shown in Table 4. For depth extensions, our method obtains at least 1.01% improvement in all metrics. With depth and breadth extension, FedSum exhibits a more obvious improvement, which shows 1.31% at least and 2.18% at most, showing the superiority of two extensions.

CNNDM	Data Heterogeneity		Improvement
FedAvg	R-F	34.11/12.87/27.17	-/-/-
	R-R	42.8/16.92/34.15	-/-/-
+ Depth Extension	R-F	35.21/13.88/28.4	+1.1/+1.01/+1.23
	R-R	44.42/18.35/35.86	+1.62/+1.43/+1.71
+ Two Extensions	**R-F**	**35.71/14.18/28.69**	**+1.6/+1.31/+1.52**
	R-R	**44.98/18.74/36.2**	**+2.18/+1.82/+2.05**

Table 4: Performances of FedAvg with depth and breadth extensions in heterogeneity setting ($Dir = 0.1$) on CNNDM.

Conclusion and Limitation

In this paper, we explore the text summarization task in FL, which is more realistic and private than related methods built upon the centralized storage. Besides, data scarcity generally arouses performance degeneration and magnifies the negative effect of leading bias. To address these challenges, we propose FedSum to maximize the data efficiency and construct the summarizer. FedSum demonstrates its promising improvement over baselines on benchmark datasets, while exhibiting its generalization in tackling the intricacies of data heterogeneity. We further verify the scalability of FedSum with various data quantities and the efficacy in mitigating the negative effect of data bias. Future investigations can build from this foundation to examine other capabilities of the model under data scarcity FL, like stability and fairness.

Further experiments are needed to verify the adaptability of our framework for other NLP tasks(e.g. machine translation and sentiment analysis). Besides, our evaluation methodology is based on averages across three experimental runs, which could be enhanced through more rigorous statistical analysis. We leave these jobs for our future work.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (Grant No. 62376099) and Natural Science Foundation of Guangdong Province (Grant No. 2024A1515010989).

References

Bi, K.; Jha, R.; Croft, B.; and Celikyilmaz, A. 2021. AREDSUM: Adaptive Redundancy-Aware Iterative Sen-

tence Ranking for Extractive Document Summarization. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 281–291. Online: Association for Computational Linguistics.

Bidoki, M.; Moosavi, M. R.; and Fakhrahmad, M. 2022. A semantic approach to extractive multi-document summarization: Applying sentence expansion for tuning of conceptual densities - ScienceDirect. *Information Processing Management*, 57(6).

Cai, D.; Wu, Y.; Wang, S.; Lin, F. X.; and Xu, M. 2023a. Efficient federated learning for modern nlp. In *Proceedings of the 29th Annual International Conference on Mobile Computing and Networking*, 1–16.

Cai, D.; Wu, Y.; Yuan, H.; Wang, S.; Lin, F. X.; and Xu, M. 2023b. Towards Practical Few-shot Federated NLP. In *Proceedings of the 3rd Workshop on Machine Learning and Systems*, EuroMLSys '23. ACM.

Cai, Z.; Shi, Y.; Huang, W.; and Wang, J. 2024. Fed-CO _{2}: Cooperation of Online and Offline Models for Severe Data Heterogeneity in Federated Learning. *Advances in Neural Information Processing Systems*, 36.

Chen, C.; Chen, Z.; Zhou, Y.; and Kailkhura, B. 2020. Fedcluster: Boosting the convergence of federated learning via cluster-cycling. In *2020 IEEE International Conference on Big Data (Big Data)*, 5017–5026. IEEE.

Chen, J.; Dong, H.; Wang, X.; Feng, F.; Wang, M.; and He, X. 2023a. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, 41(3): 1–39.

Chen, S.; Hou, W.; Hong, Z.; Ding, X.; Song, Y.; You, X.; Liu, T.; and Zhang, K. 2023b. Evolving Semantic Prototype Improves Generative Zero-Shot Learning. arXiv:2306.06931.

Cohan, A.; Dernoncourt, F.; Kim, D. S.; Bui, T.; Kim, S.; Chang, W.; and Goharian, N. 2018. A Discourse-Aware Attention Model for Abstractive Summarization of Long Documents. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 615–621. New Orleans, Louisiana: Association for Computational Linguistics.

Cohan, A.; Feldman, S.; Beltagy, I.; Downey, D.; and Weld, D. S. 2020. SPECTER: Document-level Representation Learning using Citation-informed Transformers. Association for Computational Linguistics.

Collins, L.; Hassani, H.; Mokhtari, A.; and Shakkottai, S. 2021. Exploiting Shared Representations for Personalized Federated Learning. In Meila, M.; and Zhang, T., eds., *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, 2089–2099. PMLR.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2019. AutoAugment: Learning Augmentation Policies from Data. arXiv:1805.09501.

Du, Y.; Zhang, Z.; Wu, B.; Liu, L.; Xu, T.; and Chen, E. 2023. Federated Nearest Neighbor Machine Translation. arXiv:2302.12211.

Duan, S.; Liu, C.; Han, P.; He, T.; Xu, Y.; and Deng, Q. 2023. Fed-TDA: Federated Tabular Data Augmentation on Non-IID Data. arXiv:2211.13116.

Huang, W.; Ye, M.; Shi, Z.; Li, H.; and Du, B. 2023. Rethinking federated learning with domain shift: A prototype view. In *In 2023 IEEE CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 16312–16322.

Kaddour, J.; Key, O.; Nawrot, P.; Minervini, P.; and Kusner, M. J. 2024. No train no gain: Revisiting efficient training algorithms for transformer-based language models. *Advances in Neural Information Processing Systems*, 36.

Kamp, M.; Fischer, J.; and Vreeken, J. 2021. Federated learning from small datasets. *arXiv preprint arXiv:2110.03469*.

Karimireddy, S. P.; Kale, S.; Mohri, M.; Reddi, S.; Stich, S.; and Suresh, A. T. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International conference on machine learning*, 5132–5143. PMLR.

Kawaguchi, K.; and Lu, H. 2020. Ordered SGD: A New Stochastic Optimization Framework for Empirical Risk Minimization. arXiv:1907.04371.

Kim, B.; Kim, H.; and Kim, G. 2019. Abstractive Summarization of Reddit Posts with Multi-level Memory Networks.

Ko, M.; Lee, J.; Kim, H.; Kim, G.; and Kang, J. 2021. Look at the First Sentence: Position Bias in Question Answering. arXiv:2004.14602.

Koupaee, M.; and Wang, W. Y. 2018. Wikihow: A large scale text summarization dataset. *arXiv preprint arXiv:1810.09305*.

Li, J.; Li, G.; Cheng, H.; Liao, Z.; and Yu, Y. 2024. FedDiv: Collaborative Noise Filtering for Federated Learning with Noisy Labels. arXiv:2312.12263.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. H. 2020a. Prototypical Contrastive Learning of Unsupervised Representations. arXiv:2005.04966.

Li, J.; Zhou, P.; Xiong, C.; and Hoi, S. C. H. 2021. Prototypical Contrastive Learning of Unsupervised Representations. arXiv:2005.04966.

Li, Q.; Diao, Y.; Chen, Q.; and He, B. 2022. Federated Learning on Non-IID Data Silos: An Experimental Study. In *IEEE International Conference on Data Engineering*.

Li, T.; Sahu, A. K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; and Smith, V. 2020b. Federated Optimization in Heterogeneous Networks. In Dhillon, I.; Papailiopoulos, D.; and Sze, V., eds., *Proceedings of Machine Learning and Systems*, volume 2, 429–450.

Lin, B. Y.; He, C.; Zeng, Z.; Wang, H.; Huang, Y.; Dupuy, C.; Gupta, R.; Soltanolkotabi, M.; Ren, X.; and Avestimehr, S. 2022. FedNLP: Benchmarking Federated Learning Methods for Natural Language Processing Tasks. arXiv:2104.08815.

Lin, C.-Y. 2004. ROUGE: A Package for Automatic Evaluation of Summaries. In *Text Summarization Branches Out*, 74–81. Barcelona, Spain: Association for Computational Linguistics.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2018. Focal Loss for Dense Object Detection. arXiv:1708.02002.

Liu, M.; Ho, S.; Wang, M.; Gao, L.; Jin, Y.; and Zhang, H. 2021. Federated Learning Meets Natural Language Processing: A Survey. arXiv:2107.12603.

Liu, Y.; and Lapata, M. 2019. Text Summarization with Pretrained Encoders. 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)*, 3730–3740. Hong Kong, China: Association for Computational Linguistics.

Liu, Y.; and Liu, P. 2021. SimCLS: A Simple Framework for Contrastive Learning of Abstractive Summarization.

Lu, W.; Hu, X.; Wang, J.; and Xie, X. 2023. FedCLIP: Fast Generalization and Personalization for CLIP in Federated Learning. arXiv:2302.13485.

Marfoq, O.; Neglia, G.; Bellet, A.; Kameni, L.; and Vidal, R. 2021. Federated Multi-Task Learning under a Mixture of Distributions. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 15434–15447. Curran Associates, Inc.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; and y Arcas, B. A. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, 1273–1282. PMLR.

Mindermann, S.; Brauner, J. M.; Razzak, M. T.; Sharma, M.; Kirsch, A.; Xu, W.; Höltgen, B.; Gomez, A. N.; Morisot, A.; Farquhar, S.; et al. 2022. Prioritized training on points that are learnable, worth learning, and not yet learnt. In *International Conference on Machine Learning*, 15630–15649. PMLR.

Nallapati, R.; Zhai, F.; and Zhou, B. 2017. Summarunner: A recurrent neural network based sequence model for extractive summarization of documents. In *Proceedings of the AAAI conference on artificial intelligence*, volume 31.

Park, S.; Kim, K.; Seo, J.; and Lee, J. 2024. Unsupervised Extractive Dialogue Summarization in Hyperdimensional Space. In *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE.

Tan, Y.; Long, G.; Liu, L.; Zhou, T.; Lu, Q.; Jiang, J.; and Zhang, C. 2022. Fedproto: Federated prototype learning across heterogeneous clients. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, 8432–8440.

Tang, M.; Ning, X.; Wang, Y.; Sun, J.; Wang, Y.; Li, H. H.; and Chen, Y. 2021. FedCor: Correlation-Based Active Client Selection Strategy for Heterogeneous Federated Learning. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10092–10101.

Wang, H.; Song, K.; Fan, J.; Wang, Y.; Xie, J.; and Zhang, Z. 2023. Hard Patches Mining for Masked Image Modeling. arXiv:2304.05919.

Wang, J.; Liu, Q.; Liang, H.; Joshi, G.; and Poor, H. V. 2020. Tackling the objective inconsistency problem in heterogeneous federated optimization. *Advances in neural information processing systems*, 33: 7611–7623.

Xing, L.; Xiao, W.; and Carenini, G. 2021. Demoting the Lead Bias in News Summarization via Alternating Adversarial Learning. arXiv:2105.14241.

Xu, Y.-Y.; Lin, C.-S.; and Wang, Y.-C. F. 2023. Bias-eliminating augmentation learning for debiased federated learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 20442–20452.

Yoon, T.; Shin, S.; Hwang, S. J.; and Yang, E. 2021. FedMix: Approximation of Mixup under Mean Augmented Federated Learning. arXiv:2107.00233.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. arXiv:1710.09412.

Zhang, H.; Liu, X.; and Zhang, J. 2023. DiffuSum: Generation Enhanced Extractive Summarization with Diffusion.

Zhang, J.; Liu, Y.; Hua, Y.; and Cao, J. 2024. FedTGP: Trainable Global Prototypes with Adaptive-Margin-Enhanced Contrastive Learning for Data and Model Heterogeneity in Federated Learning. *arXiv preprint arXiv:2401.03230*.

Zhong, M.; Liu, P.; Chen, Y.; Wang, D.; Qiu, X.; and Huang, X. 2020. Extractive Summarization as Text Matching. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 6197–6208. Online: Association for Computational Linguistics.

Zhong, M.; Wang, D.; Liu, P.; Qiu, X.; and Huang, X. 2019. A Closer Look at Data Bias in Neural Extractive Summarization Models. arXiv:1909.13705.

Zhou, T.; and Konukoglu, E. 2023. FedFA: Federated Feature Augmentation. arXiv:2301.12995.

Zhuang, F.; Qi, Z.; Duan, K.; Xi, D.; Zhu, Y.; Zhu, H.; Xiong, H.; and He, Q. 2020. A Comprehensive Survey on Transfer Learning. arXiv:1911.02685.