Federated Model Decomposition with Private Vocabulary for Text Classification

Zhuo Zhang[♡♠] **Xiangjing Hu**[♡] **Lizhen Qu**[♣*]
Qifan Wang[♢] **Zenglin Xu**[♡♠]
[♡]Harbin Institute of Technology (Shenzhen), China
[♠]Peng Cheng Lab, Shenzhen, China
[♣]Monash University, Melbourne, Australia [♢]Meta AI, CA, USA
{iezhuo17,starry.hxj}@gmail.com Lizhen.Qu@monash.edu
wqfcr@fb.com xuzenglin@hit.edu.cn

Abstract

With the necessity of privacy protection, it becomes increasingly vital to train deep neural models in a federated learning manner for natural language processing (NLP) tasks. However, recent studies show eavesdroppers (i.e., dishonest servers) can still reconstruct the private input in federated learning (FL). Such a data reconstruction attack relies on the mappings between vocabulary and associated word embedding in NLP tasks, which are unfortunately less studied in current FL methods. In this paper, we propose a fedrated model decomposition method that protects the privacy of vocabularies, shorted as FEDEVOCAB. In FEDEVOCAB, each participant keeps the local embedding layer in the local device and detaches the local embedding parameters from federated aggregation. However, it is challenging to train an accurate NLP model when the private mappings are unknown and vary across participants in a cross-device FL setting. To address this problem, we further propose an adaptive updating technique to improve the performance of local models. Experimental results show that FEDEVOCAB maintains competitive performance and provides better privacy-preserving capacity compared to status quo methods.

1 Introduction

Privacy-sensitive Natural Language Processing (NLP) applications, such as personal virtual assistants (Chen et al., 2017), online medical diagnosis (Hakak et al., 2020) and mobile keyboards (Ji et al., 2019), often have sensitive user data stored in local devices to protect user privacy. To ensure service quality, they also need a large amount of such data for training the corresponding machine learning models. However, it would breach user privacy or data protection law, e.g., GDPR, to apply conventional training methods by putting sensitive data in a centralized place.

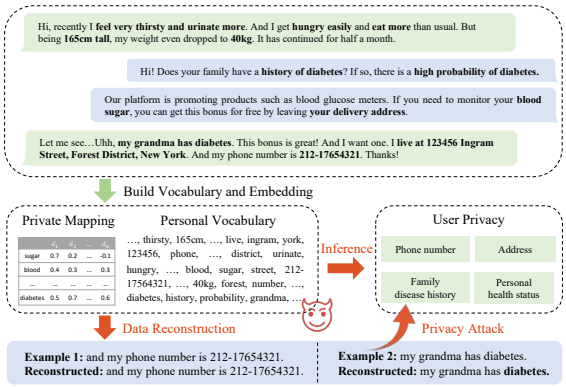

Figure 1: User-device Online Medical Conversation. The doctor records the user's health conditions, demographic information and contact details. The sensitive information of the user is made **bold**. The sensitive information can either be constructed from model updates and private mappings by e.g. data reconstruction attack, or be inferred from the local vocabulary when the local data is small.

To resolve the dilemma, federated learning (FL) methods are proposed to collaboratively learn a global model in a distributed manner without requiring sensitive data to leave local devices (McMahan et al., 2017a). In the FL framework, each local device downloads the current model, updates the model using local data, and subsequently sends model updates to a central server for improving the current model. The process is repeated until certain criteria are met. Due to improved privacy protection, there is a fast-growing interest in applying FL to NLP applications (Ge et al., 2020; Sui et al., 2020; Liu et al., 2021).

However, recent studies show that it is still possible to reconstruct user data from model updates sent to servers (Boenisch et al., 2021). Such attack methods, e.g. Deep Leakage from Gradient (DLG) (Zhu and Han, 2020), need to know the mappings between words and their embeddings. Eavesdroppers may be able to infer sensitive information by knowing which words are used in local devices, as shown in Figure 1. Hence, privacy-

*Co-corresponding author

preserving models should protect the mappings between sensitive words and their embeddings. However, it is challenging to train NLP models using current FL methods if such mappings are unknown and vary across devices. Despite potential privacy risks, prior works largely neglect the importance of protecting private vocabularies and the associated word embedding layers in local devices in the FL framework.

To address the above problems, we propose a fedrated model decomposition algorithm that protects private vocabularies, coined FEDEVOCAB, which keeps the word embedding layer (i.e., local parameters) of a deep neural network in local devices and only collaboratively updates the remaining components of the model (i.e., global parameters). Because the word embedding layer does not participate in federated aggregations and a user device may not be always-available in each federated training round in cross-device FL settings (Kairouz et al., 2021), the local parameters may be poorly coupled with those global parameters, resulting in performance degeneration (Reddi et al., 2020). To alleviate this issue, we introduce an *adaptive updating* procedure into FEDEVOCAB, which learns effective local parameters for resolving the performance drops.

To sum up, our contributions are two-fold:

- We propose a novel federated learning algorithm (FEDEVOCAB for short) to strengthen the protection of user privacy by detaching word embedding layers from federated aggregations and adaptively updating local parameters in order to further improve performance in the challenging yet practical cross-device FL setting.

- We conduct extensive experiments on three corpora to evaluate FL methods with regard to privacy protection, model utility and communication efficiency. Our method significantly outperforms the state-of-the-art FL methods in terms of privacy protection and communication efficiency, while still achieving comparable test accuracies on text classification regardless if BiLSTM (Graves et al., 2005) or DistilBERT (Sanh et al., 2019) serves as the backbone model. Compared with the state-of-the-art FL method, our method can reduce token-wise recovery by 80.1% on average, improve local model performance by about 1.3% on average, and reduce communication cost by 83 times at most.

2 Background and related work

Federated learning. Federated learning (McMahan et al., 2017a) (FL) is a privacy-enhancing distributed machine learning paradigm with individual user's data preserved locally. In FL, there has a central server and numerous user devices where the server is responsible for aggregating model parameters or gradients from users' local training. At the beginning, the service randomly selects multiple users from large-scale devices in each communication round. The selected device computes and uploads the model's parameters or gradients to the service provider that aggregates them to update the global model. The above process is repeated for multiple times until model converges. With the user data remaining on local device, FL has widely been applied to privacy-sensitive NLP tasks (Liu et al., 2021; Ge et al., 2020; Sui et al., 2020).

However, recent studies have contested the privacy-preserving ability of FL (Zhu and Han, 2020; Geiping et al., 2020; Boenisch et al., 2021; Wei et al., 2020). Zhu and Han (2020) first show how to recover the user's input text from the uploaded gradients. Boenisch et al. (2021) propose a perfect attack to recover input text based on sent gradients with near-zero costs. In these works, the mappings between words and their embeddings are critical to recovering text data due to the discrete nature of text. Unfortunately, previous studies have focused on how to protect shared gradients while largely ignoring this vital mapping challenge.

A line of work has introduced differential privacy (DP) (Dwork et al., 2014) or homomorphic encryption (Gentry, 2009) into the federated training pipeline to ensure user privacy. FL with DP is a general technique to protect user data by injecting controlled noise to shared gradients (McMahan et al., 2017b; Zhu et al., 2020). Nevertheless, it falls into the dilemma of low utility (Basu et al., 2021; Zhu and Han, 2020), and is still uncertain how much privacy can be preserved in real-world applications (Huang et al., 2020). FL with encryption method is another way to secure federated learning (Bonawitz et al., 2017). It is several orders of magnitude slower than the unencrypted equivalent, which is impractical for deep learning. Recently, Huang et al. (2020) proposed TextHide based on instance-encoding to preserve the model's utility,

which shows promising results against gradients matching attack. However, Xie and Hong (2021) have experimentally shown that TextHide cannot provide rigorous privacy guarantee. Orthogonal to previous work, our work proposes a novel method to ensure user data by protecting word embedding layers and the associated private vocabularies in local devices.

Partially local federated learning. Our work is also related to recent partially local federated learning. Liang et al. (2020) propose LG-FedAvg, which jointly learns compact local representations on each device and a global model across devices. Although LG-FedAvg is very similar to our method, the differences are as follows: 1) LG-FedAvg is tested on a few users and small data sets[1], and does not use pre-trained models (such as GloVe (Pennington et al., 2014) or DistilBERT (Devlin et al., 2018)); 2) LG-FedAvg assumes users are stateful or always-available, which is not undesirable at scale in cross-device settings (Singhal et al., 2021); 3) LG-FedAvg partitions local and global parameters unstructuredly, which may lead to performance degradation. Singhal et al. (2021) presents the state-of-the-art partially local FL method called FedRecon, which trains sensitive user-specific parameters locally and other parameters globally. FedRecon only protects mappings between sparsely sensitive words and embeddings, yet it performs poorly against data reconstruction attacks and incurs large communication overhead compared with FEDEVOCAB. Distinguished from previous studies, our work conceals all mappings between vocabulary and word embedding in the stateless federated training process, and handles the trade-off between the model's utility and privacy protection.

3 Method

3.1 Overview

Figure 2 depicts an overview of the proposed FEDEVOCAB. Our work mainly considers the classification model because text classification is a fundamental task in NLP and one of the fields widely used in FL (Zhu et al., 2020). As shown in Figure 2 (a), the federated NLP model consists of a word embedding layer, an encoder, and a classification layer.

To protect models from the data reconstruction attack while preserving the model's utility, we propose an efficient method named FEDEVOCAB, which protects the private mappings between words and their embeddings. To simulate realistic applications, we consider a more challenging yet practical cross-device FL setting where participant users are unstably, e.g., user may drop out, and the data on each device are not independent and identically distributed (Non-IID).

In the following, we describe the two key ideas of FEDEVOCAB. The first one is private mappings protection (Sec. 3.2), which is critical to defend data reconstruction attack. The second is the adaptive updating (Sec. 3.3) to minimize the performance drops in cross-device FL setting. The pseudocode of overall training processing of FEDEVOCAB is illustrated in Algorithm 1.

Algorithm 1: Training process of FEDEVOCAB

Parameters:
User set $\mathcal{N}$; Communication round $\mathcal{T}$; Epoch number $\mathcal{E}$; Learning rate η; The global parameters $\mathcal{W}_s$; The local embedding parameters $\mathcal{W}_p^k$ and the local dataset $\mathcal{D}_k$ of the k-th user; Adaptive updating algorithm $\mathbf{A}$; User-device update algorithm $\mathbf{U}$;

Initialize $\mathcal{W}_s$ on the server and $\mathcal{W}_p^k$ on each user in $\mathcal{N}$
for *each communication round* $t = 1$ *to* $\mathcal{T}$ **do**
  $\mathcal{N}^t \leftarrow$ (randomly sample K users from $\mathcal{N}$)
  for *each user* $k \in \mathcal{N}^t$ ***in parallel*** **do**
    $\mathcal{W}_s^{k,t} \leftarrow \text{UserLocalUpdate}(k, \mathcal{W}_s^{t-1})$
    send $\mathcal{W}_s^{k,t}$ to the server
  end
  Perform federated aggregation by Eq. 1
end
UserLocalUpdate $(k, \mathcal{W}_s)$**:**
  $\mathcal{W}^k \leftarrow$ (assemble $\mathcal{W}_p^k$ and $\mathcal{W}_s$)
  $\mathcal{W}_p^k \leftarrow \mathbf{A}(\mathcal{D}_k, \mathcal{W}^k)$
  for *epoch* $e = 1$ *to* $\mathcal{E}$ **do**
    $\mathcal{W}^k \leftarrow \mathbf{U}(\mathcal{D}_k, \mathcal{W}_p^k, \mathcal{W}_s)$
  end
  return $\mathcal{W}_s^k$

3.2 Private mappings protection

In this section, we elaborate on the core algorithm of FEDEVOCAB. When the eavesdroppers perform the data reconstruction attack, it is necessary to know the mappings between words and their embeddings due to the discrete nature of the text data. Based on this, it is more privacy-preserving to locally store word embedding layers and associated vocabularies than sharing them. To this end, we decouple vocabulary and word embedding from the global NLP model. As shown in Figure 2 (a), users can construct their personalized vocabulary and embedding layer and keep them on the local device. In

[1]The original paper (Liang et al., 2020) designs mobile text data set containing 572 samples across 14 participants.

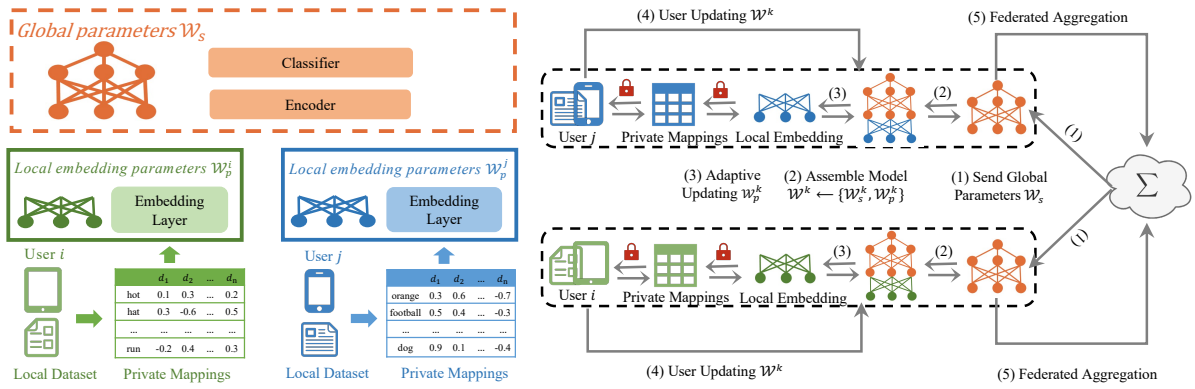

Figure 2: The overview of FEDEVOCAB method. (a) demonstrates the protection of private mappings in each local device. (b) illustrates the training process of FEDEVOCAB.

this way, FEDEVOCAB can protect the important mappings between vocabularies and embedding layers against data reconstruction attacks. Such vocabularies are personalized w.r.t. to the local texts, e.g., reflecting users' preference of wording, in contrast, global models typically have a fixed vocabulary shared among all users.

We then show how to inject private mappings protection into the federated learning pipeline. Figure 2 (b) describes the federated training process using FEDEVOCAB. Our method considers a standard setting in the FL framework that there is a central server responsible for coordinating users of interest for local training and distributing the global parameters. In each federated communication round, there are K users to be selected from all $\mathcal{N}$ users for training the FL model. We do not assume user devices are always online, which is realistic in many real-world scenarios. For instance, mobile devices may drop out (shut down) at anytime. For each selected k-th user, FEDEVOCAB decompose federated NLP model's parameters $\mathcal{W}^k = \{\mathcal{W}_p^k, \mathcal{W}_s\}$ into local embedding parameters $\mathcal{W}_p^k$ and global parameters $\mathcal{W}_s$. Unlike the canonical FL method knowing all $\mathcal{W}^k$ details, the server only accesses the global parameters and distributes $\mathcal{W}_s$ to selected users. The optimization for the server with the global parameters is:

$$\min_{w_s} \mathcal{L}(\mathcal{W}_s) = \min_{w_s} \sum_{k=1}^{K} p_k \mathcal{L}_k \left(\mathcal{W}_s, \mathcal{W}_p^k\right) \quad (1)$$

where $\mathcal{L}_k$ is the local training objective of k-th user, p_k is the weight of the k-th user such that $p_k = \frac{n_k}{\sum_{i=1}^{K} n_i}$ and n_k is amount of k-th user's dataset. Suppose that the k-th user has a supervised data set $\mathcal{D}_k = \{(x_j^k, y_j^k)\}_{j=1}^{n_k}$, and the model $G(\mathcal{W}^k, x_j^k) : \mathbb{R}^d \rightarrow \mathcal{Y}$ maps inputs $x_j^k \in \mathbb{R}^d$ to predicted label. The k-th user's local training objective $\mathcal{L}_k$ is defined by:

$$\mathcal{L}_k(\mathcal{W}_s, \mathcal{W}_p^k) = \frac{1}{n_k} \sum_{j=1}^{n_k} l\Big(G\big((\mathcal{W}_s, \mathcal{W}_p^k), x_j^k\big), y_j^k\Big) \quad (2)$$

where l is the local loss function. Next, we present how FEDEVOCAB trains such decomposed NLP model in each federated communication round, including the central server update and the user-device update.

Central server update. The server distributes global parameters to every user of interest at the beginning of each federated communication round. Then it monitors the collection of updated parameters sent by each user. After receiving the global parameters of all selected users, the server performs the federated aggregation and updates the global parameters by Eq. 1.

User-device update. When the selected users download the global parameters, they assemble a whole model with distributed global parameters and local embedding parameters. Then, selected users train the assembled model with local private data by $\mathcal{W}^k = \mathcal{W}^k - \eta \frac{\partial \mathcal{L}_k}{\partial \mathcal{W}^k}$ where η is the learning rate. After local training, the k-th user sends its updated global parameters $\mathcal{W}_s^k$ to the central server for federated aggregation.

The training process described above is repeated until specific criteria are met.

3.3 Adaptive Updating

It is realistic to assume that a user device does not participate in every round of training. Hence, the parameters of local word embeddings may well be outdated and are incompatible with the newly received global parameters. The prior studies show that it leads to deteriorated model performance (Reddi et al., 2020; Singhal et al., 2021). The challenge is thus to learn local parameters compatible with the new global parameters efficiently.

Motivated by gradient-based alternating minimization, we introduce a simple but effective adaptive updating strategy into the local training process. Specifically, once receiving new global parameters, FEDEVOCAB performs one local training epoch to adapt the local embedding parameters to the global parameters and freezes the global module during the process. Considering the limited computing resources of user devices, it is sufficient to reuse the same optimization method, which is used for updating all model parameters after this step. FEDEVOCAB can also adopt other gradient-based alternating minimization techniques (Singhal et al., 2021; Zhu and Sun, 2021), which we will explore in the future.

4 Experiments

In this section, we demonstrate how our method 1) effectively defends against data reconstruction attacks by protecting private mappings between vocabularies and word embeddings, 2) preserves the model's utility with achieving competitive performance, 3) efficiently reduces communicated parameters by keeping embedding layer in local devices. In addition, we also show adaptive updating plays a critical role in challenging yet practical cross-device FL setting.

4.1 Experimental Setup

Datasets and Non-IID Partitions. Following Lin et al. (2021), we conduct experiments on there classification datasets: 20News (Lang, 1995), AG News (Zhang et al., 2015), and SST-2 (Socher et al., 2013). These public datasets serve as benchmarks in Lin et al. (2021) to verify the proposed FL method. In order to evaluate our method in a realistic and challenging setting, we consider the Non-IID data partitioning throughout the experiments. In particular, instead of uniformly sampling the datasets, we partition the datasets by using the Dirichlet distribution as the class priors. We sample $\mathcal{D} \sim Dir(\alpha)$ and allocate data $\mathcal{D}_k$ to k-th user. α determines the degree of Non-IID, and a smaller α generates a high label distribution shift. Following Lin et al. (2021) configuration, we set $\alpha = 1.0$ as default and set the number of cross-device users as 100 for all datasets. The statistics of these datasets are in Table 1.

Dataset	# Train	# Test	# Labels
20News	11.3k (113)	7.5k	20
AG News	120k (1200)	7.6k	4
SST-2	67k (670)	1.8k	2

Table 1: Dataset Specifications. The number in parenthesis is the size of each user-device training data.

Models. We primarily evaluate two prevalent NLP models in our experiments: BiLSTM (Graves et al., 2005) and DistilBERT (Sanh et al., 2019). These models are also widely used to mimic realistic federated NLP applications (Sui et al., 2020; Lin et al., 2021; Huang et al., 2020) when the user's computational capabilities and bandwidth are restricted. We have also evaluated our methods with bigger models, such as BERT-Base, and the results are shown in B.1. More details of model hyperparameter tuning are given in Appendix A.1.

Baselines. To comprehensively evaluate the performance of FEDEVOCAB, we compare FEDEVOCAB against five baselines with different models on various datasets. **Local-only** refers to training model only using local data on each user device without collaborations between other users. In the FL family, we compare two classic and global FL methods: **FedAvg** (McMahan et al., 2017a) is the benchmark FL method that collaboratively trains a global FL model across users, and **FedProx** (Li et al., 2020) excels at handling heterogeneity in federated learning by using L_2 regularization to limit local model updates to be closer to the global model for more stable and accurate convergence. We provide two partially local FL methods which also decompose the model parameters into global parameters and local parameters: **LG-FedAvg** (Liang et al., 2020) jointly learns compact local representations on each device and a global model across all devices. **FedRecon** (Singhal et al., 2021) is the state-of-the-art FL method that trains sensitive user-specific parameters locally and other parameters federated. See Appendix A.2 for details on each baseline method implementations.

Method	DistilBERT	BiLSTM
FedAvg	88.2	87.6
FedRecon	2.4	2.1
FEDEVOCAB	**1.3**	**1.2**

Table 2: Comparison of FEDEVOCAB and baselines on the auxiliary private dataset against private tokens leakage. We report privacy tokens leakage ratio (%). Lower is better. FEDEVOCAB shows firmly privacy-preserving capacity for privacy-sensitive tokens.

4.2 Privacy Experiments

We first evaluate FEDEVOCAB against the gradient-based data reconstruction attack, which imposes a severe challenge to FL. Gradient-based data reconstruction attack is first proposed by Zhu and Han (2020) and can effectively recover users' private data. They assume eavesdroppers get access to the complete model details and are able to intercept gradients in the FL process. With this assumption, the eavesdroppers can obtain both the training inputs through the DLG optimization algorithm. See Appendix A.3 for details.

Baselines and Metrics. We consider the FL methods performing well in the utility experiments as baselines (see Sec. 4.3). In particular, we choose the classic global FL method (FedAvg) and state-of-the-art partially local FL method (FedRecon) and evaluate them for privacy protection. As the aim of attackers is to recover user text in auxiliary private datasets, we evaluate FEDEVOCAB and baselines in terms of *precision* (the average percentage of recovered words in the target texts), *recall* (the average percentage of words in the target texts are predicted) and *F1 score* which is the harmonic mean between precision and recall. Given that only private token embeddings are locally trained in FedRecon, we also exploit *privacy tokens leakage ratio* (PTLR) to evaluate each method's ability to protect privacy-sensitive tokens. More details of this attack and the auxiliary private dataset are provided in Appendix A.3.

Results. Table 2 and Figure 3 show the privacy protection results. These results demonstrate that FEDEVOCAB *can consistently outperform previous methods in defending against the data reconstruction attack.*

Compared with FedRecon, eavesdroppers cannot recover users' local data with our method by achieving almost zero recall and precision. FEDEVOCAB effectively protects privacy through making the mappings between the local device's vocabulary and their word embeddings private. In contrast, FedRecon shares most of these essential mappings, which leads to high recall and precision. As shown in Table 2, FedRecon has achieved similar performance results regarding privacy tokens protection. These results demonstrate that privacy-preserving user mappings can protect against data reconstruction attacks efficiently. More case studies can be found in Appendix B.3.

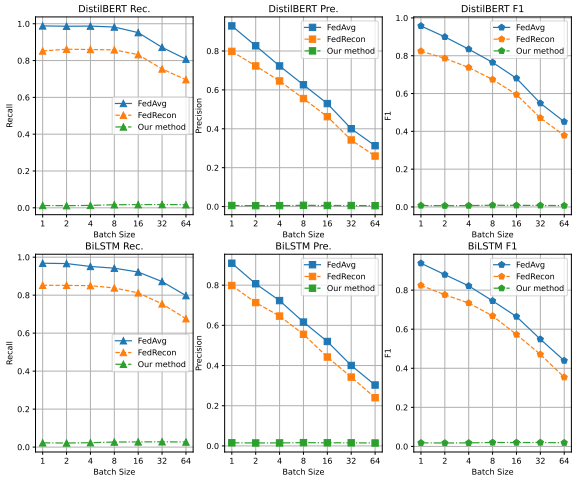

Figure 3: The data reconstruction attack results on the auxiliary private dataset. The *Pre.* denotes precision and *Rec.* denotes recall (lower is better for all metrics). FEDEVOCAB shows significantly effective defense against data reconstruction attacks.

The canonical FL method FedAvg shows frustrating results of privacy attacks, especially in sensitive tokens. It is not a surprise that FEDEVOCAB is substantially better than FedAvg in attack metrics and PTLR because FedAvg discloses the private vocabulary and associated word embedding in training processing.

In Figure 3, we show the privacy-preserving capacity of each FL method with different batch sizes. We can see that increasing batch size makes the recovery more difficult. The plausible reason behind this is that there are more variables to solve during DLG optimization in large batch size. This result is also in accordance with the experimental results in Zhu and Han (2020), which suggests that increasing the training batch size is a good defense strategy. However, there is a trade-off with limited user-device computing resources (i.e., mobile) and large batches. Notably, our method's privacy-preserving capability is not affected by the batch size, providing a practical defense strategy for users with restricted computing resources.

Methods	20News		AG News		SST-2		AVG.	
	Global	Local	Global	Local	Global	Local	Global	Local
Local	8.2	59.4	32.5	77.8	51.1	82.5	30.6	73.2
FedAvg	73.6	78.4	87.2	92.2	92.3	95.0	84.4	88.5
FedProx	73.5	78.3	**88.4**	92.9	92.2	94.9	84.7	88.7
LG-FedAvg	53.5	80.0	63.2	89.8	63.8	85.5	60.2	85.1
FedRecon	**75.5**	82.0	87.1	94.1	**93.0**	**95.8**	**85.2**	90.6
FEDEVOCAB	75.2	**86.8**	87.8	**95.5**	92.5	95.6	**85.2**	**92.6**

Table 3: Comparison of FEDEVOCAB with baselines using DistilBERT. We report the test accuracy (%) under local test and global test. Compared with best baseline, FEDEVOCAB achieves competitive performance, winning or tying in AVG.

Methods	20News		AG News		SST-2		AVG.	
	Global	Local	Global	Local	Global	Local	Global	Local
Local	7.3	42.2	34.0	64.6	54.7	83.7	32.0	63.5
FedAvg	35.4	56.3	84.1	91.6	83.0	89.6	67.5	79.2
FedProx	35.6	56.3	84.3	90.1	82.8	89.2	67.6	78.5
LG-FedAvg	14.4	53.4	48.1	88.7	59.1	85.6	40.5	75.9
FedRecon	38.3	57.3	86.0	92.0	85.4	90.2	69.9	79.8
FEDEVOCAB	**40.3**	**58.6**	**86.9**	**92.8**	**87.8**	**92.1**	**71.7**	**81.2**

Table 4: Comparison of FEDEVOCAB with baselines using BiLSTM. We report the test accuracy (%) under local test and global test. FEDEVOCAB outperforms all baselines under global test and local test.

4.3 Utility Experiments

We measure the utility of our method in terms of test accuracies and compare it with various baselines with both DistilBERT and BiLSTM on three text classification corpora. Our experiments evaluate all methods using two metrics:

1) Global Test (Global): FL methods are evaluated on each test set having the same distribution as the global data distribution. For each method, we report the geometric mean of the test accuracies collected from each local device. The global test results can measure the capability of models to learn global knowledge.

2) Local Test (Local): Each test set on each local device follows the local training data distributions, which vary across devices. For each method, we report the averaged test accuracy from all users. Compared with the global test, the local test is more realistic for real-world NLP applications and it can show performance improvement without centralizing user-sensitive data.

Results. The utility results for FEDEVOCAB and baselines are listed in Table 3 and Table 4. Overall, FEDEVOCAB *achieves competitive performance results on the global test and local test where the performance gap with the best baseline is within 1%.* This competitive result demonstrates the usability of detaching the embedding layer from federated aggregations. From Table 3 and Table 4, we have two key findings:

First, the partially local FL method with adaptive updating outperforms the global FL method. Compared with the global FL, FEDEVOCAB and FedRecon significantly improve test accuracy on the global test and local test. In particular, the largest performance gains for FEDEVOCAB and FedRecon are 8.5% and 4.7% (local test on 20News with DistilBERT in Table 3), respectively. These results show partially local FL method is able to help the local model adapt to the local task and learn global knowledge better in Non-IID data distribution. However, another partially local FL method LG-FedAvg performs poorly on most datasets and models. We conjecture that this is largely due to the fact that 1) LG-FedAvg does not use adaptive updating, resulting in a performance decrease in the cross-device FL scenario; 2) Decoupling local and global parameters in an unstructured way makes it harder to learn, especially on data with the Non-IID distributions.

Second, FL methods significantly outperform the method with only local training. As shown in Table 3 and Table 4, FL methods have better performance compared to the local-only method, especially for users with a small amount of local data and learning global knowledge (the maximum performance gap is 67.3% in the global test of 20News with DistilBERT). Unsurprisingly, the local-only method produces extremely poor results. The rea-

	20News				AG News				SST-2			
	DistilBERT		BiLSTM		DistilBERT		BiLSTM		DistilBERT		BiLSTM	
	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local	Global	Local
Ours	75.2	86.8	40.3	58.6	87.8	95.5	86.9	92.8	92.5	95.6	87.8	92.1
-adap	72.9	78.7	37.1	55.4	86.2	93.3	86.3	91.5	90.7	94.3	85.1	89.8

Table 5: Impact of adaptive updating in FEDEVOCAB.

son is the local-only method exploits only limited training samples on the local device to train the model. Conversely, FL methods significantly improve local model performance despite not having direct access to the user data where centralized training is impossible.

Compared with FedRecon, we find that FEDEVOCAB can obtain better results with BiLSTM and comparable results on DistilBERT. We conjecture that the performance gap is resulted by different ways of local model updating: 1) our method updates the word embedding layer by using all the local data during adaptive updating. FedRecon based on meta-learning (Finn et al., 2017) only uses part of the data to update a part of local word embeddings (i.e., privacy-sensitive word embedding). It is particularly challenging for this method to train on a small volume of data in a local device; 2) Collaboratively training shared word embeddings may make models suffer from instability due to Non-IID data distributions. Table 3 and Table 4 also demonstrate that FedAvg and FedProx get suboptimal performance where they cooperatively train the whole embedding layer using data from Non-IID data distributions.

4.4 Communication efficiency results

Current large-scale NLP models often contain billions of parameters, it is challenging to deploy large models in realistic FL applications due to the insufficient communication speed and low bandwidth (Sui et al., 2020). Therefore, we evaluate the communication efficiency of FEDEVOCAB and the baselines.

From Figure 4 we can tell that, compared with all baselines, FEDEVOCAB *can transmit fewer model parameters, hence it is practical for users with limited bandwidth*. The reduced communication cost of FEDEVOCAB comes from detaching the embedding layers from the federated aggregation. In Figure 4, FEDEVOCAB is more effective in the classic NLP models (e.g., only 1.41MB in BiLSTM). The global FL methods have almost no communication cost decrease compared with the partially local

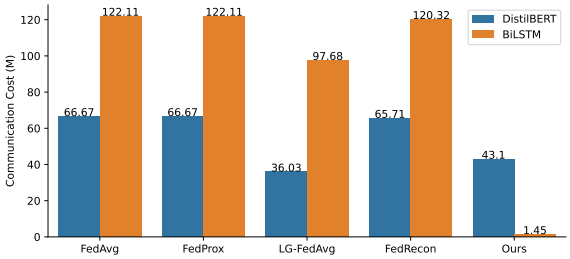

Figure 4: Communication efficiency of FEDEVOCAB and comparison baselines. We report the communicated model's parameters in each federated training round (lower is better). FEDEVOCAB can transmit fewer communication parameters, which is practical for users with limited bandwidth.

FL methods. The decreasing communication cost of FedRecon comes from the size of the privacy word embeddings. However, the privacy-sensitive words in the real world are sparse. Although LG-FedAvg can achieve more flexible traffic reduction, it needs to communicate more parameters considering its performance (see Sec.4.3).

4.5 Contribution of adaptive updating

We investigate the contribution of adaptive updating used in FEDEVOCAB. We conduct ablation studies and show the results by removing adaptive updating (- *adap*) on all datasets. As illustrated in Table 5, FEDEVOCAB with removing adaptive updating obtains worse performance. Therefore, adaptive updating is essential in enhancing FEDEVOCAB, especially for the cross-device FL setting.

5 Conclusion

We have presented FEDEVOCAB, a practical training method for privacy-preserving NLP models. FEDEVOCAB protects the private mappings between local vocabulary and the associated embedding layer by detaching the embedding layer from federated aggregation. In this manner, FEDEVOCAB allows users to personalize their vocabularies and word embedding layers. To tackle the dilemma in cross-device FL, we propose an adaptive updating to minimize performance drops. The privacy and utility experiments show FEDEVOCAB pro-

vides significantly better privacy protection than the baselines while maintaining the utility of models. Moreover, FEDEVOCAB also significantly reduces communication costs than the SOTA FL methods.

Limitations

We show the limitations of FEDEVOCAB in terms of various privacy attacks in FL and additional computing costs in local training.

Privacy attacks. FEDEVOCAB mainly considers the defense against gradient-based data reconstruction attacks and shows significant defense results. However, the attacker is still able to recover the sentence embedding before the encoder because of shared gradients of the global module. Although it is difficult for an attacker to recover the input text without knowing the mappings between words and their embeddings, it may perform membership information attack (Melis et al., 2019) and sensitive attribute information attack (Alnasser et al., 2021), which will also lead to the user privacy disclosure to a certain extent.

From the realistic scenario, the purpose of users' participation in FL is not only to improve the performance of local models, but also to protect their sensitive data from obtaining or detecting. Compared with other mentioned attacks, data reconstruction attack is the primary privacy attacks to be defensed in federated learning. Recently, some work has proposed effective defense against these mentioned attacks, such as adversarial training (Louppe et al., 2017). We think FEDEVOCAB is easy to combine these methods, which we will explore in the future.

In addition, there is no unified privacy protection metric to evaluate the existing privacy protection technologies. The evaluation metrics of differential privacy and homomorphic encryption are not suitable for measuring the privacy protection ability of FEDEVOCAB. This is also the reason why our method is not directly compared with existing privacy protection technologies. With the growing privacy concerns in deep learning, it is an urgent need to explore a general privacy protection evaluation metric.

Additional computing costs. In FEDEVOCAB, we introduced adaptive updating to reduce the performance degradation in the cross device FL scenario. During local training, we use all local data to update the embedded layer and freeze the global module. Although our method is uncomplicated, it also brings additional computing overhead, especially for devices with limited computing power and a large amount of data.

Acknowledgements

We'd like to thank all the anonymous reviewers for their careful readings and valuable comments. This work was funded by Peng Cheng Lab Project().

References

Walaa Alnasser, Ghazaleh Beigi, and Huan Liu. 2021. Privacy preserving text representation learning using bert. In *International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation*, pages 91–100. Springer.

Priyam Basu, Tiasa Singha Roy, Rakshit Naidu, Zumrut Muftuoglu, Sahib Singh, and Fatemehsadat Mireshghallah. 2021. Benchmarking differential privacy and federated learning for bert models. *arXiv preprint arXiv:2106.13973*.

Franziska Boenisch, Adam Dziedzic, Roei Schuster, Ali Shahin Shamsabadi, Ilia Shumailov, and Nicolas Papernot. 2021. When the curious abandon honesty: Federated learning is not private. *arXiv preprint arXiv:2112.02918*.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. 2017. Practical secure aggregation for privacy-preserving machine learning. In *proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191.

Hongshen Chen, Xiaorui Liu, Dawei Yin, and Jiliang Tang. 2017. A survey on dialogue systems: Recent advances and new frontiers. *Acm Sigkdd Explorations Newsletter*, 19(2):25–35.

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

Cynthia Dwork, Aaron Roth, et al. 2014. The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.*, 9(3-4):211–407.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. 2017. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR.

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

Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting gradients-how easy is it to break privacy in federated learning? *Advances in Neural Information Processing Systems*, 33:16937–16947.

Craig Gentry. 2009. Fully homomorphic encryption using ideal lattices. In *Proceedings of the forty-first annual ACM symposium on Theory of computing*, pages 169–178.

Alex Graves, Santiago Fernández, and Jürgen Schmidhuber. 2005. Bidirectional lstm networks for improved phoneme classification and recognition. In *International conference on artificial neural networks*, pages 799–804. Springer.

Saqib Hakak, Suprio Ray, Wazir Zada Khan, and Erik Scheme. 2020. A framework for edge-assisted healthcare data analytics using federated learning. In *2020 IEEE International Conference on Big Data (Big Data)*, pages 3423–3427. IEEE.

Yangsibo Huang, Zhao Song, Danqi Chen, Kai Li, and Sanjeev Arora. 2020. Texthide: Tackling data privacy in language understanding tasks. *arXiv preprint arXiv:2010.06053*.

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.

Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and Trends® in Machine Learning*, 14(1–2):1–210.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Machine Learning Proceedings 1995*, pages 331–339. Elsevier.

Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. 2020. Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2:429–450.

Paul Pu Liang, Terrance Liu, Liu Ziyin, Nicholas B Allen, Randy P Auerbach, David Brent, Ruslan Salakhutdinov, and Louis-Philippe Morency. 2020. Think locally, act globally: Federated learning with local and global representations. *arXiv preprint arXiv:2001.01523*.

Bill Yuchen Lin, Chaoyang He, Zihang Zeng, Hulin Wang, Yufen Huang, Mahdi Soltanolkotabi, Xiang Ren, and Salman Avestimehr. 2021. Fednlp: A research platform for federated learning in natural language processing. *arXiv preprint arXiv:2104.08815*.

Ming Liu, Stella Ho, Mengqi Wang, Longxiang Gao, Yuan Jin, and He Zhang. 2021. Federated learning meets natural language processing: A survey. *arXiv preprint arXiv:2107.12603*.

Gilles Louppe, Michael Kagan, and Kyle Cranmer. 2017. Learning to pivot with adversarial networks. *Advances in neural information processing systems*, 30.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017a. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.

H Brendan McMahan, Daniel Ramage, Kunal Talwar, and Li Zhang. 2017b. Learning differentially private recurrent language models. *arXiv preprint arXiv:1710.06963*.

Luca Melis, Congzheng Song, Emiliano De Cristofaro, and Vitaly Shmatikov. 2019. Exploiting unintended feature leakage in collaborative learning. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 691–706. IEEE.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.

Sashank Reddi, Zachary Charles, Manzil Zaheer, Zachary Garrett, Keith Rush, Jakub Konečnỳ, Sanjiv Kumar, and H Brendan McMahan. 2020. Adaptive federated optimization. *arXiv preprint arXiv:2003.00295*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Karan Singhal, Hakim Sidahmed, Zachary Garrett, Shanshan Wu, John Rush, and Sushant Prakash. 2021. Federated reconstruction: Partially local federated learning. *Advances in Neural Information Processing Systems*, 34.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

Dianbo Sui, Yubo Chen, Jun Zhao, Yantao Jia, Yuantao Xie, and Weijian Sun. 2020. Feded: Federated learning via ensemble distillation for medical relation extraction. In *Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP)*, pages 2118–2128.

Wenqi Wei, Ling Liu, Margaret Loper, Ka-Ho Chow, Mehmet Emre Gursoy, Stacey Truex, and Yanzhao Wu. 2020. A framework for evaluating gradient leakage attacks in federated learning. *arXiv preprint arXiv:2004.10397*.

Shangyu Xie and Yuan Hong. 2021. Reconstruction attack on instance encoding for language understanding. In *In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP'21)*.

Dun Zeng, Siqi Liang, Xiangjing Hu, and Zenglin Xu. 2021. Fedlab: A flexible federated learning framework. *arXiv preprint arXiv:2107.11621*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Ligeng Zhu and Song Han. 2020. Deep leakage from gradients. In *Federated learning*, pages 17–31. Springer.

Xinghua Zhu, Jianzong Wang, Zhenhou Hong, and Jing Xiao. 2020. Empirical studies of institutional federated learning for natural language processing. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 625–634.

Zirui Zhu and Lifeng Sun. 2021. Initialize with mask: For more efficient federated learning. In *International Conference on Multimedia Modeling*, pages 111–120. Springer.

A Experimental Details

A.1 Models Implementations

Our model implementations and each user's local training procedure are based on `FedNLP`'s code[2]. For fair comparison, we adopt the model hyperparameters straight from the default set in here for all baselines and our method. To be more concrete, the one-layer BiLSTM has 300 hidden states and the dropout rate is set to 0.5. Adam is chosen as the optimizer with an initial learning rate of 5e-3. For the transformer-based model, we exploit AdamW as the optimizer and set an initial learning rate of 5e-5 with linear decay. To make a fair comparison, we keep the same embedding layer initialization for all methods where the BiLSTM utilizes pre-trained GloVe (Pennington et al., 2014) while DistilBERT uses the primary pre-trained embedding layer. Our code are available at https://github.com/SMILELab-FL/FedVocab.

A.2 Baseline Setup

All baseline methods are based on `FedLab`'s code[3], which is a lightweight open-source framework (Zeng et al., 2021) for FL simulations. For LG-FedAvg, we tune the interpolation between the local and global model and report the best results. For FedRecon, we follow Singhal et al. (2021) next words prediction experiment where they configure that out-of-vocabulary (OOV) embeddings are local and the rest of the model (including the core vocabulary embeddings) is global. In our experiment, we set sensitive tokens (i.e., digital tokens) in the model's vocabulary as OOV tokens. The communication rounds of each FL method is 100 and one training local epoch for all models. Note that there is no collaborative training for the local-only method. To make fair comparisons, the total number of local training epochs in the local-only method will be greater than that of FL methods. We set local training epochs as 10. We train all methods on an NVIDIA Tesla V100 and report the best results.

A.3 Details of attacks

Auxiliary Dataset. For the auxiliary datatset, we sample 128 sentences from AG News dataset as the target data $\mathcal{D}_{tag}$ and perform data reconstruction attack to get recovered data $\mathcal{D}_{rec}$. To demonstrate the protection of private tokens (such as digital tokens), each sentence in the auxiliary dataset contains at least three digital tokens.

Deep Leakage from Gradients. The *Deep Leakage from Gradients* (DLG) optimization algorithm (Zhu and Han, 2020) shows that sharing the gradients can leak private training data. It starts by randomly initializing a pair of dummy data and labels and performs the usual forward and backward. When getting the user-uploaded real gradients, the attacker computes the l_2-distance between the dummy gradients deriving from the dummy data and the real gradients. And then it back-propagates this loss to update the dummy data. After multiple iterative updates, the attacker can recover the original input data.

However, NLP models need to preprocess discrete words into embeddings which is different from vision tasks where image inputs are continuous values. The mappings between vocabulary and its word embedding is able to inversely map the continuous embedding into the original token. Knowing the mappings is the critical point to recover input text in data reconstruction attacks. Following Zhu and Han (2020), we apply DLG on embedding space and minimize the gradients distance between dummy embeddings and real ones. After optimization finishes, we can get the recovered embeddings and derive original words by finding the closest token in different mappings. For FedAvg, we use the known mappings. The mappings in FedRecon are different from the known mappings only in privacy tokens' embeddings. Because FEDEVOCAB allows users to customize their mappings, we use a mapping that is completely different from the known mapping.

B Extra Results

B.1 The results of bigger model

To verify the effectiveness of the large model, we evaluated our methods with bigger models, such as BERT-Base. The results are shown in Figure 5 and Table 6. We find that FEDEVOCAB can outperform the baseline methods in terms of privacy protection and achieve competitive performance results in terms of utility.

B.2 The effect of different embedding initializations

We present results with different initialization (`Different init`) and the default same initialization (`Same init`) on three benchmark datasets. In

[2]https://github.com/FedML-AI/FedNLP/
[3]https://github.com/SMILELab-FL/FedLab

Methods	20News		AG News		SST-2		AVG.	
	Global	Local	Global	Local	Global	Local	Global	Local
Local	9.2	60.7	37.2	79.7	51.8	83	32.7	74.5
FedAvg	80.1	83.9	88.9	93.2	95.2	96.7	88.1	91.3
FedProx	**80.2**	83.8	**89.7**	93.7	**95.5**	96.7	**88.5**	91.4
LG-FedAvg	57.3	85.1	64.7	89.6	65.7	87.8	62.6	87.5
FedRecon	78.4	87.3	86.0	93.6	94.6	96.9	86.3	92.6
FEDEVOCAB	77.5	**91.5**	89.3	**95.9**	94.2	**96.9**	87.0	**94.8**

Table 6: Comparison of FEDEVOCAB with baselines using BERT-Base.

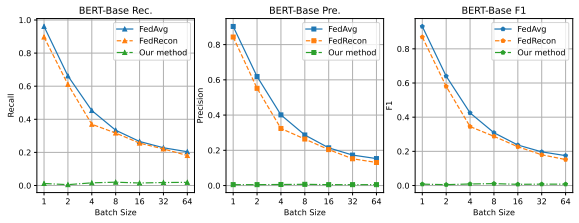

Figure 5: The data reconstruction attack results on the auxiliary private dataset with bigger model BERT-Base.

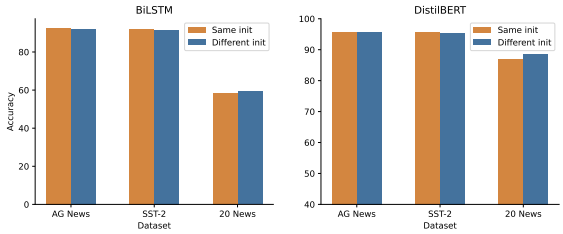

Figure 6: The utility results of different embedding initialization.

the `Different init` setting, each user can choose its embedding initialization from pre-trained embedding initialization set (i.e., GloVe embedding initialization from {6B, 42B, 840B} or DistilBERT embedder initialization from {DistilBERT, BERT-Base, BERT-Large, GPT2}). As a result, users can not only customize the vocabularies but also choose different embedding initialization. The performance results are shown in Figure 6. Compared with `Same init`, FEDEVOCAB can even outperform the models trained with the same initial embeddings in terms of privacy protection (see Figure 3) while achieving comparable performance in terms of utility.

We can observe that a slight performance degradation of `Different init` is consistent for SST-2 and AG News, compared with `Same init`. But we notice that `Different init` can outperform `Same init` on the 20 News dataset. We speculate that the reason is due to the characteristics of the dataset. Compared with SST-2 and AG News, 20 News has fewer data per user and more category labels. As a result, 20New is more challenging in the federated learning setting. `Different init` can help users more personalize their local models and perform better in more challenging settings.

Orig: *SPACE.com - UPDATE: Story first posted 6:49 a.m. ET,* 11 / 16 / 2004
FedAvg: *space. com - update : story first posted* 6 : 49 *a . m . et ,* 11 / 16 / 2004 **FedRecon**: *space. com - update : story first posted* martial *:* paper *a . m . et ,* front / faster / hop . **FEDEVOCAB**: directlyuts broader organise gh sqlel sort fal kyle minority wind attend easy presenting lo presenting esports blogging official beganvery death experimental.

Table 7: Example of data reconstruction attack for DistilBERT. "Orig" denotes the original input sentence and the italicized *token* denotes less privacy-sensitive text. In our experiment, we set digital token as privacy-sensitive token and tokens indicate no recovery by eavesdropper.

Orig: *the st. louis cardinals and pitcher matt morris agree to one-year, $* 2.5*-*million *contract.*
FedAvg: *the st. louis cardinals and pitcher matt morris agree to one - year , $* 2 . 5 *- million contract .* **FedRecon**: *the st. louis cardinals pitcher matt morris agree to one - year , $* lennon . upon *contract .* **FEDEVOCAB**: pops application fusion 20-4420-4429 slice baby room countries a 82785 less for 4

Table 8: Example of data reconstruction attack for BiLSTM. "Orig" denotes the original input sentence and the italicized *token* denotes less privacy-sensitive text.

B.3 Case Study

Table 7 and Table 8 show examples of the data reconstruction attack from AG News dataset. We perform the DLG described in A.3 and set batch size as 1. Compared with baselines, FEDEVOCAB can provide the firmly privacy-preserving ability for the user-device text and privacy-sensitive tokens. FedAvg divulges user privacy tokens and almost all user text data since it discloses the shared model's details and its gradients in the communication process. Although FedRecon can protect private tokens, other input tokens can be reconstructed, which may increase the risk of legal and ethical issues in the real world.