

# Federated Continual Learning for Text Classification via Selective Inter-client Transfer

Yatin Chaudhary<sup>1,2</sup>, Pranav Rai<sup>1,2</sup>, Matthias Schubert<sup>2</sup>, Hinrich Schütze<sup>2</sup>, Pankaj Gupta<sup>1</sup>

<sup>1</sup>DRIMCo GmbH, Munich, Germany | <sup>2</sup>University of Munich (LMU), Munich, Germany  
{firstname.lastname}@drimco.net

## Abstract

In this work, we combine the two paradigms: Federated Learning (FL) and Continual Learning (CL) for text classification task in cloud-edge continuum. The objective of Federated Continual Learning (FCL) is to improve deep learning models over life time at each client by (relevant and efficient) knowledge transfer without sharing data. Here, we address challenges in minimizing inter-client interference while knowledge sharing due to heterogeneous tasks across clients in FCL setup. In doing so, we propose a novel framework, *Federated Selective Inter-client Transfer* (FedSeIT) which selectively combines model parameters of foreign clients. To further maximize knowledge transfer, we assess domain overlap and select informative tasks from the sequence of historical tasks at each foreign client while preserving privacy. Evaluating against the baselines, we show improved performance, a gain of (average) 12.4% in text classification over a sequence of tasks using five datasets from diverse domains. To the best of our knowledge, this is the first work that applies FCL to NLP.

## 1 Introduction

**Federated Learning** (Yurochkin et al., 2019; Li et al., 2020; Zhang et al., 2020; Karimireddy et al., 2020; Caldas et al., 2018) in Edge Computing<sup>1</sup> (Wang et al., 2019) has gain attraction in recent years due to (a) data privacy and sovereignty- especially imposed by government regulations (GDPR, CCPA etc.), and (b) the need for sharing knowledge across edge (client) devices such as mobile phones, automobiles, wearable gadgets, etc. while maintaining data localization. Federated Learning (FL) is a privacy-preserving machine learning (ML) technique that enables collaborative training of ML models by sharing model parameters across distributed clients through a central server - without

sharing their data. In doing so, a central server aggregates model parameters from each participating client and then distribute the aggregated parameters, where ML models at each client are optimized using them - achieving inter-client transfer learning. In this direction, the recent works such as FedAvg (McMahan et al., 2017), FedProx (Li et al., 2020), FedCurv (Shoham et al., 2019) have introduced parameter aggregation techniques and shown improved learning at local clients - augmented by the parameters of foreign clients.

On the other hand, the edge devices generate a continuous stream of data where the data distribution can drift over time; therefore, the need for Continual Learning like humans do. **Continual Learning** (CL) (Thrun, 1995; Kumar and Daume III, 2012; Kirkpatrick et al., 2017; Schwarz et al., 2018; Gupta et al., 2020) empowers deep learning models to continually accumulate knowledge from a sequence of tasks - reusing historical knowledge while minimizing catastrophic forgetting (drift in learning of the historical tasks) over life time.

**Federated Continual Learning (FCL):** This work investigates the combination of the two paradigms of ML: Federated Learning and Continual Learning with an objective to model a sequence of tasks over time at each client via inter-client transfer learning while preserving privacy and addressing heterogeneity of tasks across clients. There are two key challenges of FCL: (1) *catastrophic forgetting*, and (2) *inter-client interference* due to heterogeneity of tasks (domains) at clients. At central server, FedAvg (McMahan et al., 2017) aggregates-averages model parameters from each client without considering inter-client interference. To address this, FedWeIT (Yoon et al., 2021) approach performs FCL by sharing *task-generic* (via dense base parameters) and *task-specific* (via task-adaptive parameters) knowledge across clients. In doing so, at the server, they aggregate the dense base parameters however, no aggregation of the

<sup>1</sup>extends cloud computing services closer to data sources

task-adaptive parameters, and then broadcast both the types of parameters. See further details in Figure 2 and section 2.2. FedWeIT, the first approach in FCL, investigates computer vision tasks (e.g., image classification), however the technique has limitations in aligning domains of foreign clients while augmented learning at each local client using task-adaptive parameters - that are often misaligned with local model parameters in parameter space (McMahan et al., 2017) due to heterogeneity in tasks. Therefore, a simple weighted additive composition technique does not address inter-client interference and determine domain relevance in foreign clients while performing transfer learning.

**Contributions:** To the best of our knowledge, this is the first work that applies FCL to NLP task (text classification). At each local client, to maximize the inter-client transfer learning and minimize inter-client interference, we propose a novel approach, *Federated Selective Inter-client Transfer* (FedSeIT) that aligns domains of the foreign task-adaptive parameters via projection in the augmented transfer learning. To exploit the effectiveness of domain-relevance in handling a number of foreign clients, we further extend FedSeIT by a novel task selection strategy, *Selective Inter-client Transfer* (SIT) that efficiently selects the relevant task-adaptive parameters from the historical tasks of (many) foreign clients - assessing domain overlap at the global server using encoded data representations while preserving privacy. We evaluate our proposed approaches: FedSeIT and SIT for Text Classification task in FCL setup using five NLP datasets from diverse domains and show that they outperforms existing methods. Our main contributions are as follows:

(1) We have introduced Federated Continual Learning paradigm to NLP task of text classification that collaboratively learns deep learning models at distributed clients through a global server while maintaining data localisation and continually learn over a sequence of tasks over life time - minimizing catastrophic forgetting, minimizing inter-client interference and maximizing inter-client knowledge transfer.

(2) We have presented novel techniques: FedSeIT and SIT that align domains and select relevant task-adaptive parameters of the foreign clients while augmented transfer learning at each client via a global server. Evaluating against the baselines, we have demonstrated improved performance, a

Notation	Description
$c, s$	current client, global server
$t, r$	current task, current round
$C, T, R$	total number of clients, tasks, rounds
$\mathcal{T}_c^t$	training dataset for task $t$ of client $c$
$\theta_c^t$	model parameter set for task $t$ of client $c$
$\theta_G$	global aggregated server parameter
$x_i^t, y_i^t$	input document, label pair in $\mathcal{T}_c^t$
$\mathbf{z}, \hat{y}$	CNN output dense vector; predicted label
$\mathbf{B}_c^t, \mathbf{A}_c^t$	local base and task-adaptive parameters
$\alpha_c^t, \mathbf{m}_c^t$	scalar attention parameters, mask parameters
$\oplus$	concatenation operation
$\mathbf{W}_c^t, \mathbf{W}_f^t$	projection matrices: alignment and augmentation
$K$	number of parameters selected for transfer in SIT
$D, \mathbb{D}$	word embedding dimension, dataset
$L_d, L_c^t$	unique labels in dataset $d$ , each task dataset $\mathcal{T}_c^t$
$\mathcal{F}$	filter size of convolution layer
$\mathcal{N}^{\mathcal{F}}$	number of filters in convolution layer
$\lambda_1, \lambda_2$	hyperparameters: sparsity, catastrophic forgetting

Table 1: Description of the notations used in this work where matrices and vectors are denoted by uppercase and lowercase bold characters respectively.

gain of (average) 12.4% in text classification over a sequence of tasks using 5 datasets. Implementation of FedSeIT is available at <https://github.com/RaiPranav/FCL-FedSeIT> (See appendix C and D).

## 2 Methodology

### 2.1 Federated Continual Learning

Consider a global server  $s$  and  $C$  distributed clients, such that each client  $c_c \in \{c_1, \dots, c_C\}$  learns a local ML model on its privately accessible sequence of tasks  $\{1, \dots, t, \dots, T\}$  with datasets  $\mathcal{T}_c \equiv \{\mathcal{T}_c^1, \dots, \mathcal{T}_c^t, \dots, \mathcal{T}_c^T\}$ , where  $\mathcal{T}_c^t = \{x_i^t, y_i^t\}_{i=1}^{N^t}$  is a labeled dataset for  $t_{th}$  task consisting of  $N^t$  pairs of documents  $x_i^t$  and their corresponding labels  $y_i^t$ . Please note that there is no relation among the datasets  $\{\mathcal{T}_c^t\}_{c=1}^C$  for task  $t$  across all clients. Now, in each training round  $r \in \{1, \dots, R\}$  for task  $t$ , the training within FCL setup can be broken down into three steps:

**Continual learning at client:** Each client  $c_c$  effectively optimizes its model parameters  $\theta_c^{t(r)}$  (for task  $t$ ) using task dataset  $\mathcal{T}_c^t$  in a continual learning setting such that: (a) it minimizes *catastrophic forgetting* of past tasks, and (b) it boosts learning on the current task using the knowledge accumulated from the past tasks.

**Parameter aggregation at server:** After training on task  $t$ , each client  $c_c$  transmits updated model parameters  $\theta_c^{t(r)}$  to the server  $s$  and server aggregates them into the global parameter  $\theta_G$  to

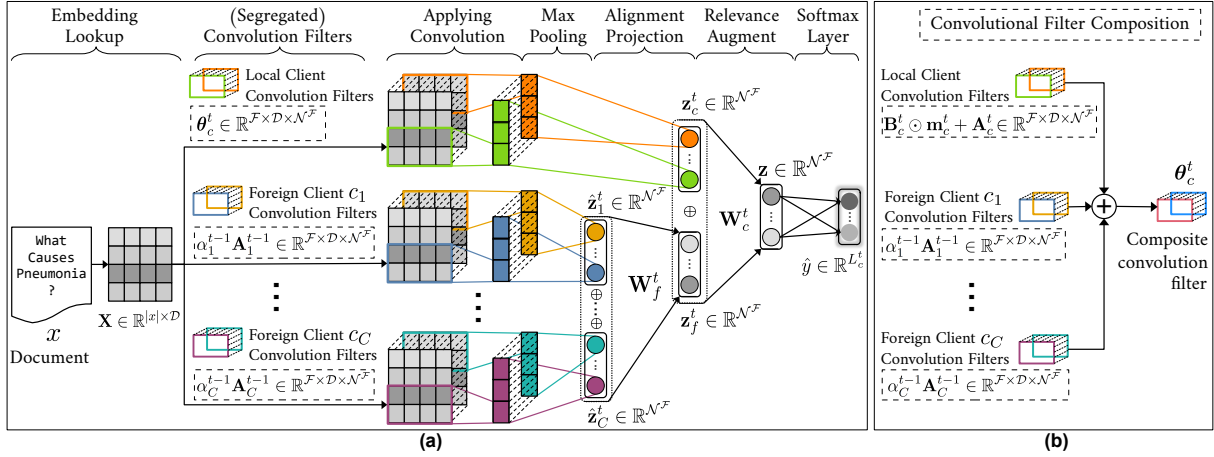


Figure 1: (a) Illustration of the proposed FedSeIT framework where, task-adaptive parameters of foreign clients are segregated and domain-aligned for selective utilization. *HowToRead*: Note the coloring scheme in convolution filters of local and foreign clients and their application in convolution. (b) Weighted additive filter composition performed in the baseline model: FedWeIT. Note the composite  $\theta_c^t$  vs segregated convolution filters of FedSeIT.

accumulate the knowledge across all clients.

**Inter-client knowledge transfer:** Server transmits global aggregated parameter  $\theta_G$  to all participating clients for inter-client knowledge transfer in the next training round  $r + 1$ .

**Challenges:** However, there are two main sources of *inter-client interference* within FCL setup: (1) using a single global parameter  $\theta_G$  during parameter aggregation at server to capture the cross-client knowledge (Yoon et al., 2021) due to model parameters trained on irrelevant foreign client tasks, and (2) *non-alignment* of the foreign client model parameters given the heterogeneous task domains across clients. This leads to the hindrance of the local model training at each client by updating its parameters in erroneous directions, thus resulting in: (a) *catastrophic forgetting* of the client’s historical tasks, and (b) sub-optimal learning of client’s current task. For brevity, we will omit notation of round  $r$  from further equations and mathematical formulation except algorithms.

## 2.2 Federated Selective Inter-client Transfer

To tackle the above-mentioned challenges, we propose Federated Selective Inter-client Transfer (FedSeIT) framework which aims to minimize *inter-client interference* and communication cost while maximizing inter-client knowledge transfer in FCL paradigm. Motivated by Yoon et al. (2021), FedSeIT model decomposes each client’s model parameters  $\theta_c^t$  into a set of three different parameters: (1) dense *local base parameters*  $\mathbf{B}_c^t$  which captures and accumulates the *task-generic knowledge*

across client’s private task sequence  $\mathcal{T}_c$ , (2) sparse *task-adaptive parameters*  $\mathbf{A}_c^t$  which captures the *task-specific knowledge* for each task in  $\mathcal{T}_c$ , and (3) sparse *mask parameters*  $\mathbf{m}_c^t$  which allow client model to selectively utilize the global knowledge. For each client  $c_c$ ,  $\mathbf{B}_c^t$  is randomly initialized only once before training on the first task and shared throughout the task sequence  $\mathcal{T}_c$ , while a new  $\mathbf{A}_c^t$  and  $\mathbf{m}_c^t$  parameters are initialized for each task  $t$ . At the global server, we have global parameter  $\theta_G$  which accumulates task-generic knowledge across all clients i.e., global knowledge, by aggregating local base parameters sent from all clients. Finally, for each client  $c_c$  and task  $t$ , the model parameters  $\theta_c^t$  can be described as:

$$\begin{aligned} \mathbf{B}_c^t &\leftarrow \theta_G \\ \theta_c^t &= \mathbf{B}_c^t \odot \mathbf{m}_c^t + \mathbf{A}_c^t \end{aligned} \quad (1)$$

where, each client initializes  $\mathbf{B}_c^t$  using  $\theta_G$  received from the server containing global knowledge, before training on task  $t$ , to enable inter-client knowledge transfer. Therefore, the first term signifies selective utilization of global knowledge using the mask parameter  $\mathbf{m}_c^t$ , which restricts the impact of *inter-client interference* during server aggregation. Due to additive decomposition of parameters, the second term  $\mathbf{A}_c^t$  captures task specific knowledge.

Another key benefit of parameter decomposition is that by accessing task-adaptive parameters  $\mathbf{A}_c^t$  of the past tasks from foreign clients, a client can selectively utilize task-specific knowledge of the relevant tasks, thus further minimizing *inter-client interference* and maximizing knowledge transfer.

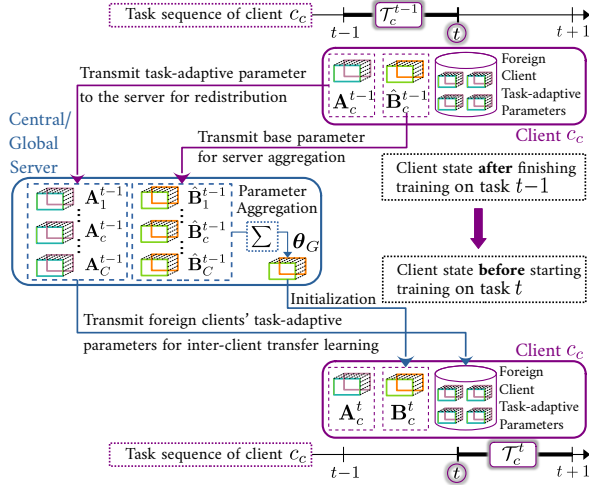


Figure 2: Inter-client Transfer Learning in FCL: (top) broadcasting client-to-server model parameters of client  $c_c$  after training the task  $t - 1$ ; (middle) parameter aggregation at the server; and (bottom) reception of the parameters at the client  $c_c$  before training the task  $t$

Therefore, before training on task  $t$  in FedSeIT, global server enables distribution of task-adaptive parameters of the past tasks from all clients i.e.,  $\{\mathbf{A}_c^{t-1}\}_{c=1}^C$ , to each client. However, as already discussed earlier, given the heterogeneous task domains across clients, the parameters learned during training at foreign clients are often *non-aligned* in the parameter space. So, FedSeIT addresses this issue by treating each foreign task-adaptive parameter separately and align their local model output vectors via projection to augment the relevant knowledge during learning on the current task. Also, for each foreign task-adaptive parameter, we introduce an attention parameter  $\alpha$  to further control the impact of *inter-client interference*. Figure 2 illustrates the parameter transmissions between each client and the server before training on task  $t$ .

**Client activity:** Before training on task  $t$ , each client  $c_c$  receives the global parameter  $\theta_G$  and  $C$  foreign clients' task-adaptive parameters  $\{\mathbf{A}_c^{t-1}\}_{c=1}^C$  from the server. Then, each client partially updates its base parameter  $\mathbf{B}_c^t$  with the non-zero entries of  $\theta_G$  i.e.,  $\mathbf{B}_c^t(n) = \theta_G(n)$  where  $n$  is the nonzero element of  $\theta_G$ . After optimizing local model parameters for task  $t$  in the continual learning setup, the sparsified base parameter  $\hat{\mathbf{B}}_c^t = \mathbf{B}_c^t \odot \mathbf{m}_c^t$  and the task-adaptive parameter  $\mathbf{A}_c^t$  are transmitted to the global server  $s$ .

**Server activity:** After training on task  $t$ , the global server receives local base parameters

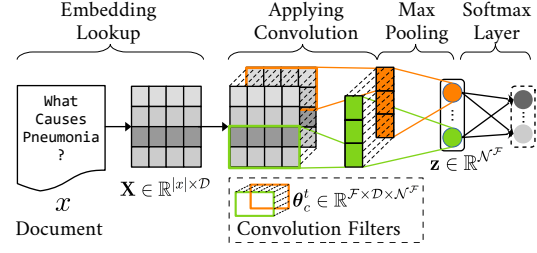


Figure 3: A detailed illustration of CNN client model used in FedSeIT framework.

$\{\hat{\mathbf{B}}_c^t\}_{c=1}^C$  and task-adaptive parameters  $\{\mathbf{A}_c^t\}_{c=1}^C$  from all clients. To capture global knowledge, the server aggregates base parameters  $\{\hat{\mathbf{B}}_c^t\}_{c=1}^C$  into the global parameter  $\theta_G$ . Then, before training on the next task  $t + 1$ , server transmits the task-adaptive parameters  $\{\mathbf{A}_c^t\}_{c=1}^C$  along with the global parameter  $\theta_G$  to each client for inter-client knowledge transfer via selective utilization.

**Method:** In this work, we use Convolutional Neural Network (CNN) (Kim, 2014), illustrated in Figure 3, as the *continual learning* client model in *federated continual learning* setting. So, the client model parameters  $\mathbf{B}_c^t, \mathbf{A}_c^t, \theta_c^t \in \mathbb{R}^{\mathcal{F} \times \mathcal{D} \times \mathcal{N}^{\mathcal{F}}}$  are a set of convolutional filters where,  $\mathcal{F}$  is the filter size,  $\mathcal{D}$  is the word embedding dimension and  $\mathcal{N}^{\mathcal{F}}$  is the count of filters. See appendix B for detailed model description.

Consider training for task  $t$  at client  $c_c$  using dataset  $\mathcal{T}_c^t \equiv \{x_i^t, y_i^t\}_{i=1}^{N^t}$ , the FedSeIT framework segregates client's local model parameters  $\theta_c^t$  from the foreign client's task-adaptive parameters  $\{\mathbf{A}_c^{t-1}\}_{c=1}^C$ . As illustrated in Figure 1(a), for an input document  $x \in \mathcal{T}_c^t$  an embedding matrix  $\mathbf{X} \in \mathbb{R}^{|x| \times \mathcal{D}}$  is generated via embedding lookup and then using  $\mathbf{X}$  the CNN model computes: (1) one client dense vector  $\mathbf{z}_c^t \in \mathbb{R}^{\mathcal{N}^{\mathcal{F}}}$  using local client parameters  $\theta_c^t$ , and (2)  $C$  foreign dense vectors  $\hat{\mathbf{z}}_i^t \in \mathbb{R}^{\mathcal{N}^{\mathcal{F}}}$  using foreign clients' task-adaptive parameters  $\{\mathbf{A}_c^{t-1}\}_{c=1}^C$  as:

$$\begin{aligned} \mathbf{z}_c^t &= \text{CNN}(x, \theta_c^t) \\ \hat{\mathbf{z}}_i^t &= \text{CNN}(x, \alpha_i^{t-1} \mathbf{A}_i^{t-1}) \end{aligned} \quad (2)$$

where,  $i \in \{1, \dots, C\}$ ,  $|x|$  is the count of tokens in  $x$ ,  $\alpha$  is the attention parameter and CNN is the convolution & max-pooling function. Then we align the foreign parameters in parameter space by concatenating and projecting all foreign dense vectors  $\{\hat{\mathbf{z}}_i^t\}$  to get a single foreign vector  $\mathbf{z}_f^t \in \mathbb{R}^{\mathcal{N}^{\mathcal{F}}}$ . Finally, we selectively augment the relevant knowledge from foreign clients by concatenating

and projecting  $\mathbf{z}_c^t, \mathbf{z}_f^t$  for prediction of label  $\hat{y}$  as:

$$\begin{aligned}\mathbf{z}_f^t &= \mathbf{W}_f^t \{\hat{\mathbf{z}}_1^t \oplus \dots \oplus \hat{\mathbf{z}}_C^t\} \\ \mathbf{z} &= \mathbf{W}_c^t \{\mathbf{z}_c^t \oplus \mathbf{z}_f^t\} \\ \hat{y} &= \text{Softmax}(\mathbf{z})\end{aligned}\quad (3)$$

where  $\mathbf{W}_f^t \in \mathbb{R}^{C\mathcal{N}^{\mathcal{F}} \times \mathcal{N}^{\mathcal{F}}}$ ,  $\mathbf{W}_c^t \in \mathbb{R}^{2\mathcal{N}^{\mathcal{F}} \times \mathcal{N}^{\mathcal{F}}}$  are the projection matrices,  $\mathbf{z} \in \mathbb{R}^{\mathcal{N}^{\mathcal{F}}}$  is the augmented vector,  $\text{Softmax}$  is a softmax layer,  $\hat{y} \in \mathbb{R}^{L_c^t}$  is the model prediction and  $L_c^t$  is the unique label set in  $\mathcal{T}_c^t$ . By segregation of foreign parameters via concatenation and projection of the CNN outputs, FedSeIT enables local client model to align foreign parameters in feature space and augment knowledge from the relevant foreign tasks. In doing so, FedSeIT effectively circumvents the *inter-client interference* due to heterogeneous task domains while maximizing inter-client transfers.

**Comparison with FedWeIT:** As illustrated in Figure 1(b), in contrast to our approach, in the baseline FedWeIT framework each client performs inter-client knowledge transfer via weighted composition of foreign task-adaptive parameters i.e., convolution filters, along with local parameters as:

$$\boldsymbol{\theta}_c^t = \mathbf{B}_c^t \odot \mathbf{m}_c^t + \mathbf{A}_c^t + \sum_{i \in C} \alpha_i^{t-1} \mathbf{A}_i^{t-1} \quad (4)$$

However, due to heterogeneity of task domains across clients, the simple additive aggregation of foreign client parameters with local parameters leads to *inter-client interference* and sub-optimal learning of the local task. Unlike FedWeIT (where the attention  $\alpha$  parameters decide the relevance of foreign client parameters), the segregation of foreign client parameters in our proposed FedSeIT method enables the local model to selective augment relevant knowledge from foreign clients' task-adaptive parameters via projection.

**Training:** For task  $t$  at each client  $c_c$ , the FCL optimization objective for local client model can be decomposed into three components as:

$$\begin{aligned}\underset{\mathbf{A}_c^t, \mathbf{B}_c^t, \mathbf{m}_c^t}{\text{minimize}} & \sum_{i=1}^{N_t} \mathcal{L}(\hat{y}_i^t, y_i^t) \\ & + \lambda_1 \mathcal{L}_{sp}([\mathbf{m}_c^t, \mathbf{A}_c^{1:t}]) \\ & + \lambda_2 \sum_{i=1}^{t-1} \|\Delta \mathbf{B}_c^t \odot \mathbf{m}_c^i - \Delta \mathbf{A}_c^i\|_2^2\end{aligned}\quad (5)$$

where,  $\hat{y}_i^t$  and  $y_i^t$  are the predicted and true label respectively for input  $x_i^t$  and  $N_t$  is the number of

---

### Algorithm 1 Proposed FedSeIT Framework

---

**Input:** Task datasets  $\{\mathcal{T}_c^{1:t}\}_{c=1}^C$ , Global parameter  $\boldsymbol{\theta}_G$   
**Output:**  $\{\mathbf{B}_c^t, \mathbf{m}_c^{1:t}, \alpha_c^{1:t}, \mathbf{A}_c^{1:t}, \mathbf{W}_c^{1:t}, \mathbf{W}_f^{1:t}\}_{c=1}^C$

- 1: Server initializes  $\boldsymbol{\theta}_G$
- 2: Each client  $c \in C \equiv \{1, \dots, C\}$  connects with server
- 3: **for** task  $t = 1, \dots, T$  **do**
- 4:     **for** round  $r = 1, \dots, R$  **do**
- 5:         Server transmits global parameter  $\boldsymbol{\theta}_G$  to all  $c \in C$
- 6:         Each client  $c \in C$  initializes  $\mathbf{B}_c^t$  using  $\boldsymbol{\theta}_G$
- 7:         **if**  $r = 1$  **then**
- 8:             Each client  $c \in C$  initializes  $\mathbf{A}_c^t, \mathbf{m}_c^t$
- 9:         **if**  $r = 1$  **and**  $t \neq 1$  **then**
- 10:             Server transmits  $\{\mathbf{A}_c^{(t-1, R)}\}_{c=1}^C$  to all  $c \in C$
- 11:             Prepare model parameters:  $\boldsymbol{\theta}_c^t \leftarrow \mathbf{B}_c^t \odot \mathbf{m}_c^t + \mathbf{A}_c^t$
- 12:             Each client  $c \in C$  minimizes equation 5 using equations 1,2,3 to learn task  $t$  in continual learning setup
- 13:             Each client  $c \in C$  transmits  $\hat{\mathbf{B}}_c^{(t, r)}$  to server
- 14:              $\boldsymbol{\theta}_G \leftarrow \text{Agg}(\{\hat{\mathbf{B}}_c^{(t, r)}\}_{c \in C}) = \frac{1}{|C|} \sum_{c=1}^C \hat{\mathbf{B}}_c^{(t, r)}$
- 15:             Server distributes  $\boldsymbol{\theta}_G$  to all clients  $c \in C$
- 16:             **if**  $r = R$  **and** SIT is enabled **then**
- 17:                 Each client computes  $\hat{\mathcal{T}}_c^t$  using equation 6
- 18:                 Each client transmits  $\{\hat{\mathcal{T}}_c^t, \mathbf{A}_c^{(t, R)}\}$  to server
- 19:                 Server computes task-task similarity using  $\hat{\mathcal{T}}_c^t$
- 20:                 Server selects top- $K$  parameters  $\forall c \in C$
- 21:                 Server distributes  $\{\mathbf{A}_c^{(k, R)}\}_{k=1}^K$  to all  $c \in C$
- 22:             **if**  $r = R$  **and** SIT is not enabled **then**
- 23:                 Each client transmits  $\mathbf{A}_c^{(t, R)}$  to server
- 24:                 Server distributes  $\{\mathbf{A}_c^{(t, R)}\}_{c=1}^C$  to all  $c \in C$

---

documents. The first term is the model training objective for current task  $t$ . The second term is a sparsity objective to induce sparsity in the mask  $\mathbf{m}_c^t$  and task specific parameters  $\mathbf{A}_c^t$  for efficient server-client communication, where  $\lambda_1$  is a hyper-parameter to regulate sparsity. The final term is the *continual learning* regularization (Kirkpatrick et al., 2017) objective to minimize *catastrophic forgetting* by controlling the drift in parameters learned from the past tasks. Here,  $\Delta \mathbf{B}_c^t$  is the change in  $\mathbf{B}_c^t$  between the current and previous task i.e.,  $\Delta \mathbf{B}_c^t = \mathbf{B}_c^t - \mathbf{B}_c^{t-1}$ ,  $\Delta \mathbf{A}_c^i$  is the change in task-adaptive parameters for task  $i$  between the current and previous time-step and  $\lambda_2$  is the hyper-parameter to regulate sparsity. The task specific parameters  $\mathbf{A}_c^{1:t-1}$  of the past tasks are updated to balance the change in  $\mathbf{B}_c^t$  i.e.,  $\Delta \mathbf{B}_c^t$ , which minimizes drift in the solutions learned for the past tasks. For higher values of  $\lambda_2$ , the training objective set high penalty on forgetting of the previous tasks. Algorithm 1 describes the FedSeIT framework where, the server aggregation function  $\text{Agg}$  is agnostic to the choice of available methods like FedAvg, FedProx etc. In FedSeIT, we use FedAvg.

**Dense Layer Sharing (DLS):** Unlike the convolution filters of CNN model that captures the transferable n-gram patterns in the data, the dense layer

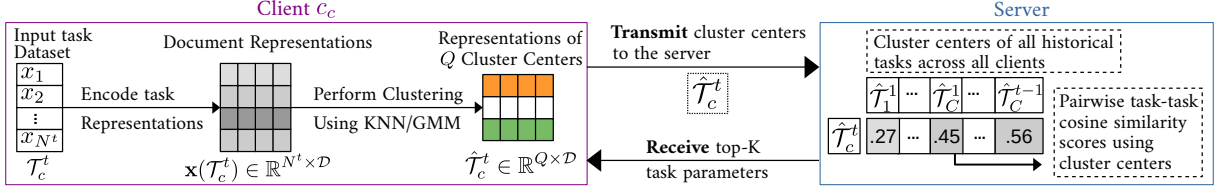


Figure 4: Detailed illustration of the proposed SIT method, where foreign task-adaptive parameters are selected based on task domain-relevance to maximize inter-client transfer learning and minimize communication cost.

parameters  $\{\mathbf{W}_c^t, \mathbf{W}_f^t\}$  capture fine-grained alignment information based on the selection and ordering of unique output labels  $y$  and foreign client parameters  $\{\mathbf{A}_c^{t-1}\}_{c=1}^C$ . In heterogeneity, the server aggregation and distribution of dense layer parameters introduce sub-optimal initialization point for training of future tasks across clients. Therefore, in FedSeIT, we do not share projection layer parameters with server by default. In doing so, we also increase client privacy in response to adversarial attacks. To validate our hypothesis, we evaluate a variant of our proposed model **FedSeIT+DLS**, where we share the dense projection layer parameters  $\{\mathbf{W}_c^t, \mathbf{W}_f^t\}$  in FedSeIT framework.

### 2.3 Selective Inter-client Transfer (SIT)

In FedSeIT, before training on task  $t$ , each client receives  $C$  foreign task-adaptive parameters  $\{\mathbf{A}_1^{t-1}, \dots, \mathbf{A}_C^{t-1}\}$  of the previous task  $t-1$  from each client via server  $s$  for inter-client transfer learning. However, given task heterogeneity, the previous task parameters might be irrelevant for learning current task which could lead to *inter-client interference*. To resolve this, we could transmit parameters of all historical tasks from all foreign clients i.e.,  $\{\mathbf{A}_1^1, \dots, \mathbf{A}_C^{t-1}\}$ , to minimize interference by finding relevant parameters. But, this could lead to burgeoning of computational complexity and communication cost increasing with each completed task. Therefore, to tackle this, we propose Selective Inter-client Transfer (SIT) method, which uses encoded task representations to efficiently explore all historical tasks across foreign clients via domain overlap, and selects the relevant parameters to minimize *inter-client interference* and maximize knowledge transfer.

**Client:** For each task  $t$ , before training, each client  $c_c$  generates the encoded vector representation  $\mathbf{x}(x) \in \mathbb{R}^D$  for each document  $x$  in task dataset  $\mathcal{T}_c^t$  via embedding lookup and averaging. After that, using K-Nearest Neighbor (KNN) or Gaussian Mixture Model (GMM) algorithm, each

client performs clustering on the encoded vector representations as follows:

$$\mathbf{x}(x) = \frac{1}{|x|} \sum_{i=1}^{|x|} \text{EmbeddingLookup}(x_i, \mathbf{E}) \quad (6)$$

$$\hat{\mathcal{T}}_c^t = \text{Clustering}(\{\mathbf{x}(x) | \forall x \in \mathcal{T}_c^t\})$$

where,  $\hat{\mathcal{T}}_c^t \in \mathbb{R}^{Q \times D}$  are the representations of the cluster centers,  $\text{Clustering} \in \{\text{KNN}, \text{GMM}\}$ ,  $Q$  denotes the number of cluster centers and  $\mathbf{E}$  is a pre-trained word embedding repository like Word2Vec (Mikolov et al., 2013). Ultimately, each client transmits  $\hat{\mathcal{T}}_c^t$  to the global server.

**Server:** Once the server receives representations of cluster centers  $\hat{\mathcal{T}}_c^t$  of task  $t$  from client  $c_c$ , it computes pairwise task-task domain overlap using average cosine-similarity score between the current client task  $\hat{\mathcal{T}}_c^t$  and each historical task across all clients  $\{\hat{\mathcal{T}}_1^1, \dots, \hat{\mathcal{T}}_C^{t-1}\}$ . The server then selects and transmits top- $K$  relevant (high similarity) parameters to the client for inter-client transfer learning, where  $K$  is a hyperparameter. Therefore, in scenarios where task history is long ( $t > 10$ ) and/or clients are too many ( $C > 10$ ), SIT can keep the computational complexity of client model constant while minimizing *inter-client interference* and maximizing knowledge transfer. To test this, we apply SIT in FedSeIT framework with  $K \in \{3, 5\}$ .

**Comparison with FedWeIT:** Such an approach of parameter selection by assessing domain relevance is missing in baseline FedWeIT framework, where to control the computational complexity and communication cost, each client transmits task-adaptive parameters of only the previous task to the server. However, as already discussed, these parameters could be irrelevant for learning current task, thus resulting in *inter-client interference*.

## 3 Experiments and Analysis

To demonstrate the effectiveness of our proposed FedSeIT framework, we perform evaluation on Text Classification task using five datasets from

Datasets	Baseline	Without SIT	With SIT ( $K = 3$ )	With SIT ( $K = 5$ )	Without SIT	With SIT ( $K = 3$ )	With SIT ( $K = 5$ )	Gain (%)	
	FedWeIT	FedSeIT	FedSeIT	FedSeIT	FedSeIT +DLS	FedSeIT +DLS	FedSeIT +DLS		
$\lambda_2 = 1.0$	<b>R8</b>	79.1 $\pm$ 2.5	90.5 $\pm$ 0.6	90.5 $\pm$ 0.5	<b>90.6 <math>\pm</math> 0.7</b>	83.1 $\pm$ 5.0	82.3 $\pm$ 5.9	82.4 $\pm$ 3.5	$\uparrow$ 14.5
	<b>TMN</b>	84.2 $\pm$ 0.9	88.2 $\pm$ 0.2	88.3 $\pm$ 0.1	<b>88.6 <math>\pm</math> 0.3</b>	84.7 $\pm$ 2.4	84.3 $\pm$ 1.9	83.6 $\pm$ 1.3	$\uparrow$ 5.2
	<b>TREC6</b>	78.2 $\pm$ 1.9	<b>83.6 <math>\pm</math> 0.6</b>	83.6 $\pm$ 0.5	83.3 $\pm$ 0.2	78.2 $\pm$ 2.1	78.3 $\pm$ 1.5	77.7 $\pm$ 1.8	$\uparrow$ 6.9
	<b>TREC50</b>	85.1 $\pm$ 1.1	88.4 $\pm$ 2.5	88.5 $\pm$ 2.4	<b>88.7 <math>\pm</math> 1.9</b>	86.4 $\pm$ 1.1	86.3 $\pm$ 1.5	85.2 $\pm$ 0.2	$\uparrow$ 4.2
	<b>SUBJ</b>	86.5 $\pm$ 0.9	88.5 $\pm$ 0.6	88.6 $\pm$ 0.6	<b>89.5 <math>\pm</math> 0.3</b>	88.9 $\pm$ 0.5	89.7 $\pm$ 1.7	89.0 $\pm$ 0.9	$\uparrow$ 3.5
$\lambda_2 = 0.1$	<b>R8</b>	68.1 $\pm$ 10.4	<b>89.9 <math>\pm</math> 0.7</b>	89.9 $\pm$ 1.1	89.7 $\pm$ 0.8	60.7 $\pm$ 14.5	61.9 $\pm$ 14.9	65.3 $\pm$ 10.6	$\uparrow$ 32
	<b>TMN</b>	80.9 $\pm$ 0.6	<b>88.5 <math>\pm</math> 0.1</b>	88.4 $\pm$ 0.07	88.3 $\pm$ 0.7	72.8 $\pm$ 9.1	73.3 $\pm$ 9.8	71.1 $\pm$ 8.5	$\uparrow$ 9.4
	<b>TREC6</b>	77.1 $\pm$ 3.6	84.7 $\pm$ 1.1	<b>84.9 <math>\pm</math> 1.3</b>	84.7 $\pm$ 1.2	78.3 $\pm$ 1.6	78.9 $\pm$ 1.3	76.8 $\pm$ 3.0	$\uparrow$ 10.1
	<b>TREC50</b>	76.1 $\pm$ 0.9	<b>81.2 <math>\pm</math> 3.5</b>	81.2 $\pm$ 2.9	80.9 $\pm$ 2.1	69.8 $\pm$ 3.9	69.4 $\pm$ 4.7	68.9 $\pm$ 1.2	$\uparrow$ 6.7
	<b>SUBJ</b>	85.6 $\pm$ 0.6	89.1 $\pm$ 0.4	88.8 $\pm$ 0.1	<b>90.4 <math>\pm</math> 1.5</b>	88.1 $\pm$ 1.3	90.2 $\pm$ 1.8	84.2 $\pm$ 2.6	$\uparrow$ 5.6

Table 2: Comparison of our proposed FedSeIT framework (with and without SIT) against FedWeIT baseline model using Task-averaged Test Accuracy (TTA) scores for two different values of  $\lambda_2 \in \{1.0, 0.1\}$ . Best score for each dataset (row) is shown in **bold** and **Gain (%)** denotes Bold vs FedWeIT.

diverse domains and present our qualitative and quantitative analysis in FCL setup.

**Datasets:** We present experimental evaluation results on Reuters8 (R8), Tag My News (TMN), TREC6, TREC50 and Subjectivity (SUBJ) datasets. Here, R8 and TMN datasets belong to the *News* domain, TREC6 and TREC50 belong to *Question Classification* domain and SUBJ is a *movie reviews* dataset with binary labels. Please see Appendix A for more details regarding datasets.

**Experimental setup:** We follow Yoon et al. (2021) for our FCL experimental setup where we use CNN based Text Classification model (Kim, 2014) as the local client model. For all experiments, we use three clients i.e.,  $C = 3$ , five tasks per client i.e.,  $T = 5$ , 10 rounds per task i.e.,  $R = 10$ , with 50 epochs in each round,  $\lambda_2 \in \{0.1, 1.0\}$  and  $K \in \{3, 5\}$  for Selective Inter-client Transfer (SIT). Please see Appendix C for detailed hyperparameter settings for all of our experiments. To run the experiments in FCL setup, we need to generate task datasets  $\mathcal{T}_c^t$  for each task  $t$ . Consider a dataset  $\mathbb{D}_d$  with a unique label set  $L_d \equiv \{L_1, \dots, L_{N_d}\}$ , where  $N_d$  is the count of unique labels. Now, for each task dataset  $\mathcal{T}_c^t$ , we randomly pick a fixed number of unique labels  $L_c^t \subseteq L_d$ , where the count of unique task labels  $|L_c^t| = 4$  is fixed for all tasks across all clients except subjectivity, which has 2 unique labels shared by all tasks. If label  $L_1$  gets selected for 3 tasks, then we follow the *non-iid splitting strategy* which simply divides the documents

labeled with  $L_1$  i.e.,  $\mathbb{D}_d(L_1)$ , into three mutually exclusive and equal parts, thus ensuring heterogeneity. We use this strategy to split the training and validation sets. However, to create the test set for each task dataset  $\mathcal{T}_c^t$ , we select all of the documents labeled with  $L_c^t$  in the complete test dataset i.e.,  $\{\mathbb{D}_d^{test}(L_i) | L_i \in L_c^t\}$  without splitting. As this work focuses on the new challenges which arise due to combination of FL and CL paradigms in FCL setup, we compare our work with related FCL methods and not with standalone FL, CL methods.

**Baseline:** As the only existing work in FCL domain, FedWeIT has shown significantly superior performance compared to naïve FCL methods that is why we adopt FedWeIT as the baseline method.

**Evaluation Metric:** In each experiment, once the training is finished for all  $C$  clients, we freeze the model parameters for all tasks and compute *Micro-averaged Accuracy* (MAA) score for each of the  $T$  past tasks of all  $C$  clients. Finally, we average  $C \times T$  MAA scores to compute the final Task-averaged Test Accuracy (TTA) score to compare our proposed model with baseline. For each experiment, we report an average TTA score of 3 runs using 3 different seed values for ordering of tasks at each client.

### 3.1 Results: Comparison to baseline

Table 2 shows final TTA scores after completion of all tasks across all clients on five Text Classification datasets. We find the following observations:

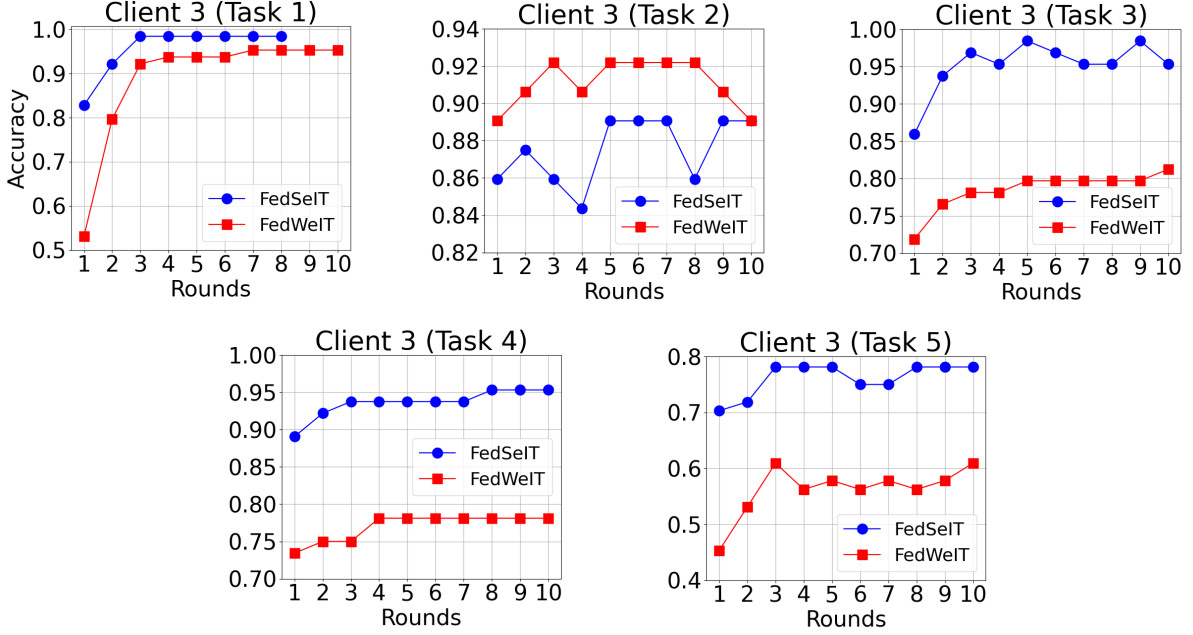


Figure 5: Test set accuracy scores for all 5 tasks of Client-3 in FedSeIT framework using Reuters8 dataset ( $\lambda_2 = 1.0$ ). Each data point denotes the test accuracy score at the end of each training round for 10 rounds of 50 epochs.

(1) For all the datasets, our proposed model FedSeIT consistently outperforms the baseline method FedWeIT. For instance, for the data R8, the classification accuracy is 79.1% vs 90.5% from FedWeIT and FedSeIT (without SIT), respectively. Overall (column 3 vs column 2), on an average over the five datasets, FedSeIT model outperforms FedWeIT by 6.4% and 12.4% for  $\lambda_2 = 1.0$  (higher penalty on catastrophic forgetting) and  $\lambda_2 = 0.1$  (lower penalty) respectively. We then observe that FedSeIT also applies to sparse-setting, for example, small (R8) vs large (TMN) corpora. See Table A for the data statistics. The results suggest that the selective utilization and domain-alignment of task-adaptive parameters at local clients prevent inter-client interference and maximise transfer learning.

(2) To demonstrate the effectiveness of Selective Inter-client Transfer (SIT) approach by limiting the number of foreign parameters ( $K = 3$  or 5) at local client while augment-learning, Table 2 shows a comparison between FedSeIT models (trained without and with SIT) and FedWeIT baseline. For all the datasets, it is observed that the test scores of FedSeIT trained with SIT ( $K = 3$ ) are similar to FedSeIT trained without SIT ( $C = 3$ ) while having the same computational expense; however, reducing the number of parameters when the length of sequence of historical tasks grows at the clients.

Additionally, we observe an improved perfor-

mance in the classification by increasing  $K = 5$ , i.e., extending the window-size of the number of foreign parameters (considered in augmented-learning) from a sequence of historical tasks in continual learning. For instance, on an average, the performance gain of FedSeIT vs FedWeIT (column 5 and 2) is: 6.8% and 12.5% for  $\lambda_2 = 1.0$  and  $\lambda_2 = 0.1$ , respectively. This suggests that the more relevant and domain-aligned foreign parameters boost the augmented-learning at each local client, i.e, by selecting relevant foreign parameters from all historical tasks of foreign clients using SIT.

(3) Next, we investigate the application of dense-layer-sharing (DLS) in FedSeIT, in order to compare with FedSeIT (without DLS) and FedWeIT. For all the datasets, note that the FedSeIT outperforms FedSeIT+DLS, validating our hypothesis that using exclusive dense layer parameters at local client boots domain-alignment and identification of relevant foreign parameters. Overall an average over the five datasets, FedSeIT outperforms FedSeIT+DLS (column 3 vs 6) by 4.4% and 19.1% for  $\lambda_2=1.0$  and  $\lambda_2=0.1$ , respectively.

In summary, evaluating against the baseline FedWeIT approach, the proposed FedSeIT have shown improved performance, a average gain of **12.4%** in text classification over a sequence of tasks using the five datasets from diverse domains.



### 3.2 Ablation: Learning over training rounds

Here, we demonstrate the performance of text classification at a local client (3rd client) for a sequence of five tasks at each training round (10 rounds). Figure 5 shows the test set accuracy scores (at the end of each training round) for all five tasks of Client-3 using R8 dataset. In 4/5 tasks, the test accuracy score of the FedSeIT model at the end of round 1 is noticeably higher than the FedWeIT approach. Interestingly, in task 2, we find that FedWeIT outperforms FedSeIT over rounds, however converge at the same accuracy in the final round.

In essence, the proposed method FedSeIT in FCL setup have shown that the alignment and relevance of foreign tasks parameters at each client (for all the tasks at each model training round) minimise inter-client interference and improve inter-client transfer learning without the dense-layer-sharing.

## 4 Conclusion and Future Work

We have applied Federated Continual Learning to text classification for heterogeneous tasks and addressed the challenges of inter-client interference and domain-alignment in model parameters of local vs foreign clients while minimizing catastrophic forgetting over a sequence of tasks. We have presented two novel techniques: FedSeIT and SIT that improves local client augmented-learning by assessing domain overlap and selecting informative tasks from the sequence of historical tasks of each foreign client while preserving privacy. Furthermore, the novel selection strategy using SIT determines relevant foreign tasks from the complete historical tasks of all foreign clients by assessing domain overlap while preserving privacy. We have evaluated the proposed approaches using five text classification data sets and shown a gain (average) of 12.4% over the baseline.

Although we have applied FedSeIT framework to the document-level text classification task, we can further apply the proposed framework to additional NLP tasks. Inspired by continual Topic Modeling (Gupta et al., 2020) at document-level and continual Named Entity Recognition (Monaikul et al., 2021) at token-level classification, we can further extend these existing works with the proposed FedSeIT framework in federated settings.

### Limitations

In FCL paradigm, the computation complexity of augmented-learning at a client increases when the

number of foreign clients grows exponentially. In future work, it is an interesting research direction to explore hierarchical federated learning techniques (Abad et al., 2020) to limit the number of foreign client parameters injected into augmented-learning (applying convolution filters and projections in CNN of FedSeIT) at a local client. Additionally, due to limited compute on edge devices such as mobiles, wearable devices, sensors, etc., the application of FCL in the cloud-edge continuum is still in early days that requires distillation and pruning of large ML models such as CNN for text classification - as presented in this paper.

### Acknowledgements

This research was supported (also executed at) by DRIMCo GmbH, Munich, Germany. We are also grateful to anonymous reviewers for their constructive comments and suggestions.

### References

- Mehdi Salehi Heydar Abad, Emre Ozfatura, Deniz Gunduz, and Ozgur Ercetin. 2020. Hierarchical federated learning across heterogeneous cellular networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8866–8870. IEEE.
- Sebastian Caldas, Virginia Smith, and Ameet Talwalkar. 2018. Federated kernelized multi-task learning. In *Proc. SysML Conf.*, pages 1–3.
- Pankaj Gupta, Yatin Chaudhary, Thomas Runkler, and Hinrich Schuetze. 2020. Neural topic modeling with continual lifelong learning. In *International Conference on Machine Learning*, pages 3907–3917. PMLR.
- Sai Praneeth Karimireddy, Satyen Kale, Mehryar Mohri, Sashank Reddi, Sebastian Stich, and Ananda Theertha Suresh. 2020. Scaffold: Stochastic controlled averaging for federated learning. In *International Conference on Machine Learning*, pages 5132–5143. PMLR.
- Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751.
- James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114.

- Abhishek Kumar and Hal Daume III. 2012. Learning task grouping and overlap in multi-task learning. *arXiv preprint arXiv:1206.6417*.
- 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.
- Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Natawut Monaikul, Giuseppe Castellucci, Simone Filice, and Oleg Rokhlenko. 2021. Continual learning for named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 13570–13577.
- Jonathan Schwarz, Wojciech Czarnecki, Jolena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. 2018. Progress & compress: A scalable framework for continual learning. In *International Conference on Machine Learning*, pages 4528–4537. PMLR.
- Neta Shoham, Tomer Avidor, Aviv Keren, Nadav Israel, Daniel Benditkis, Liron Mor-Yosef, and Itai Zeitak. 2019. Overcoming forgetting in federated learning on non-iid data. *CoRR*, abs/1910.07796.
- Sebastian Thrun. 1995. A lifelong learning perspective for mobile robot control. In *Intelligent robots and systems*, pages 201–214. Elsevier.
- Xiaofei Wang, Yiwen Han, Chenyang Wang, Qiyang Zhao, Xu Chen, and Min Chen. 2019. In-edge ai: Intelligentizing mobile edge computing, caching and communication by federated learning. *IEEE Network*.
- Jaehong Yoon, Wonyong Jeong, Giwoong Lee, Eunho Yang, and Sung Ju Hwang. 2021. Federated continual learning with weighted inter-client transfer. In *ICML*, pages 12073–12086. PMLR.
- Mikhail Yurochkin, Mayank Agarwal, Soumya Ghosh, Kristjan Greenewald, Nghia Hoang, and Yasaman Khazaeni. 2019. Bayesian nonparametric federated learning of neural networks. In *International Conference on Machine Learning*, pages 7252–7261. PMLR.
- Fengda Zhang, Kun Kuang, Zhaoyang You, Tao Shen, Jun Xiao, Yin Zhang, Chao Wu, Yueting Zhuang, and Xiaolin Li. 2020. Federated unsupervised representation learning. *arXiv preprint arXiv:2010.08982*.

## A Datasets

Table 3 shows detailed data statistics of five labeled datasets used to evaluate our proposed FedSeIT framework using Text Classification task: Reuters8 (R8) and Tag My News (TMN) datasets belong to the *News* domain, TREC6 and TREC50 belong to the *Question Classification* domain and Subjectivity (SUBJ) is a *movie reviews* dataset with binary labels.

## B Local client model

In FedSeIT we use CNN (Kim, 2014) model for Text Classification as the local client model. The CNN model is made up of convolution (Conv) layers and fully connected (FC) layers. The CNN model parameters can be described as follows: (1) For Conv layers:  $\mathbf{B}_c^t, \mathbf{A}_c^t, \boldsymbol{\theta}_c^t \in \{\mathbb{R}^{\mathcal{F}_l \times \mathcal{D} \times \mathcal{N}_l^{\mathcal{F}}}\}_{l=1}^{L_{Conv}}$  are a set of convolutional filters where,  $l$  is the layer indicator,  $L_{Conv}$  is the total number of Conv layers,  $\mathcal{F}_l$  is the filter size for layer  $l$ ,  $\mathcal{D}$  is the input word embedding dimension and  $\mathcal{N}_l^{\mathcal{F}}$  is the count of filters in layer  $l$ , and (2) For FC layers:  $\mathbf{B}_c^t, \mathbf{A}_c^t, \boldsymbol{\theta}_c^t \in \{\mathbb{R}^{\mathcal{I}_l \times \mathcal{O}_l}\}_{l=1}^{L_{FC}}$  are a set of parameter matrices where,  $l$  is the layer indicator,  $L_{FC}$  is the total number of FC layers and  $\mathcal{I}_l, \mathcal{O}_l$  are the input, output dimensions for layer  $l$ . For all layers,  $\mathbf{m}_c^t$  is the masking vector matching the output dimension of  $\mathbf{B}_c^t$ . As illustrated in Figure 3, for an input document  $x$ , the CNN model performs three different tasks: (1) performing word embedding lookup of  $x$  to generate an input matrix  $\mathbf{X} \in \mathbb{R}^{|x| \times \mathcal{D}}$ , (2) applying convolutional filters and max-pooling over  $\mathbf{X}$  to generate an intermediate dense vector  $\mathbf{z} \in \mathbb{R}^{\mathcal{N}^{\mathcal{F}}}$ , and (3) applying softmax layer on  $\mathbf{z}$  to predict the label for input  $x$ .

## C Experimental Setup

Table 4 shows the settings of all hyperparameters used in the experimental setup to evaluate our proposed FedSeIT framework on Text Classification task using 5 datasets.

## D Reproducibility: Code

To run the experiments and reproduce the scores reported in the paper content, we have provided the implementation of our proposed FedSeIT framework at <https://github.com/RaiPranav/FCL-FedSeIT>. Information regarding model training and data pre-processing is provided in the README file.

Datasets	#Train docs	#Test docs	Maximum length*	Mean length*	Median length*	Num Classes
<b>Reuters8</b>	5485	2189	964	102	64	8
<b>TMN</b>	24816	7779	48	18	18	7
<b>TREC6</b>	5451	499	37	10	9	6
<b>TREC50</b>	5451	499	37	10	9	50
<b>SUBJ</b>	7999	1999	120	24	22	2

Table 3: Detailed statistics of the datasets used for evaluation of our proposed FedSeIT framework on Text Classification task. Here, \* indicates that length is equal to the count of word tokens.

Hyperparameter	Value	Description
$\lambda_2$	{0.1, 1.0}	Higher value minimizes catastrophic forgetting
$K$	{3, 5}	Number of tasks sent to each client from server after assessing domain overlap
$L_{Conv}$	3	Number of convolution layers
$\mathcal{F}$	{3, 4, 5}	Kernel Sizes of convolution filters
$\mathcal{N}^{\mathcal{F}}$	128	Number of convolution filters in each convolution layer
Stride	1	Stride when applying convolution filter
Dilation Rate	1	Dilation rate when applying convolution filter
Padding	Valid	Padding when applying convolution filter
Dropout	0.3	-
Learning rate	1e-4	-
Activation Function	ReLU	-
Batch Size	64	-
$\lambda_1$	1e-3	Regularization parameter for sparsity constraint
Random Seeds (for task allocation and parameter initialization)	{1, 2}	-
$C$	3	Number of clients
$T$	5	Tasks picked per client
$R$	10	Number of Rounds
Number of epochs per training round	50	Early stopping due to convergence is possible
Early stopping patience	3	-
Random Seeds (for task generation)	42	-
Clustering Algorithm for Selective Inter-client Transfer (SIT)	KNN	-
$Q$	200	Count of cluster centers to extract
$\mathcal{D}$	300	Word embedding dimension

Table 4: Hyperparameter settings used in the experimental evaluation setup of our proposed FedSeIT framework using Text Classification task.