A Federated Learning Approach using Pre-trained BART Models for Text Classification and Generation Tasks

Quang-Vinh Pham
Department of Data Science
TMA Tech Group, Gia Lai, Vietnam
Email: pqvinh@tma.com.vn

Xuan-Tuyen Le
Department of Information Technology
Quy Nhon University, Vietnam
Email: tuyen4554100008@st.qnu.edu.vn

Quang-Hung Le*
Department of Information Technology
Quy Nhon University, Vietnam
Email: lequanghung@qnu.edu.vn

***Abstract*—This paper proposes a novel approach, which integrates Federated Learning (FL) with pre-trained BART models for text classification and generation tasks. In particular, we first design FedAvgBART, a framework that adapts the BART architecture for FL. We then fine-tune the pre-trained BART models using FedAvgBART to support both text classification and generation tasks. Comprehensive experiments performed to evaluate the effectiveness and efficiency of this approach on benchmark datasets using two BART-variants of different sizes, including BART-large and DistilBART. The significant results demonstrate that federated training can surpass centralized fine-tuning in performance. Specifically, BART-large exhibits exceptional proficiency in classification tasks, while DistilBART excels in text generation, offering superior computational efficiency for resource-limited clients. Furthermore, BART-large maintains greater stability across diverse client scales, whereas non-IID data disproportionately affects smaller models, underscoring the robustness of larger architectures. Our implementation is available in this Github repository.[1]**

***Index Terms*—Federated Learning, BART Model, BART-large, DistilBART, Text Generation, Text Classification, Natural Language Processing.**

I. Introduction

Text classification and generation are two fundamental tasks in Natural Language Processing (NLP) with broad applications ranging from sentiment analysis [1], spam detection [2], and news categorization [3] to dialogue systems [4], machine translation [5], [6], and abstractive summarization [7]. While text classification enables automatic labeling of large-scale textual data, text generation empowers downstream applications such as conversational agents and content creation. With the growing availability of sensitive textual data, strict privacy regulations such as the GDPR [8] have raised concerns about centralizing user data in conventional machine learning pipelines. These challenges have motivated the development of FL, a distributed paradigm that enables model training across clients without transferring raw data to a central server, thereby preserving privacy while supporting large-scale learning [9], [10].

At the same time, advances in pre-trained transformer models, including BERT [11], DistilBERT [12], TinyBERT [13], MobileBERT [14], and T5 [4], have significantly improved the performance of NLP tasks. Among them, BART [7] has achieved state-of-the-art results as a powerful sequence-to-sequence model for classification and generation tasks [15]. Integrating FL with BART models (e.g., BART-large and DistilBART) provides a promising solution for privacy preservation and collaborative learning across multiple clients. Despite its strong performance, several challenges remain unresolved. First, applying BART-large directly in FL settings is impractical due to its high computational and communication costs, which limit deployment on resource-constrained clients such as mobile or edge devices [16]. Second, heterogeneous and non-IID data distributions across clients often degrade the convergence and generalization performance of the global model [17]. Third, the trade-off between efficiency and accuracy when leveraging distilled variants such as DistilBART has not been systematically studied in federated environments [18], [19].

To address the aforementioned challenges, in this paper we propose a novel approach, which integrates FL with pre-trained BART models for text classification and generation tasks. The goal of this work is to evaluate the feasibility and performance of these models in federated environments for multi-task NLP applications. Our main contributions are summarized as follows:

1) We introduce FedAvgBART, the first federated adaptation of BART that performs joint training across distributed clients while preserving the encoder–decoder architecture.
2) We analyze the training objective of FedAvgBART under heterogeneous and non-IID conditions, showing

* Corresponding author: Quang-Hung Le (lequanghung@qnu.edu.vn)
[1]https://github.com/vinh1988/FedAvgBART

its ability to balance global model accuracy and local personalization.
3) We conduct extensive experiments on benchmark datasets for both text classification and generation, demonstrating that FedAvgBART achieves competitive performance compared to centralized training.
4) We compare the performance of DistilBART and BART-large in both centralized and federated settings, highlighting the effect of model size on efficiency and accuracy.
5) We investigate the impact of the number of clients and data distribution on model performance, providing insights for future FL applications in NLP.

The rest of the paper is structured as follows. Related work is presented in Section II. Section III gives some brief reviews of background knowledge. Our proposed method is presented in Section IV. Section V analyses and discusses the evaluation results obtained. Finally, conclusions and future work are highlighted in Section VI.

II. Related work

FL concept was first introduced by McMahan et al. [9] in 2017. This field has been applied successfully to domains where data privacy and ownership are critical, such as healthcare [20] and finance [21]. In the field of NLP, FL has primarily been explored for text classification [22], [23]. These studies often leverage pre-trained transformer-based models like BERT [11] and RoBERTa [24], which have shown strong performance even in distributed learning environments [25]. Recently, some frameworks such as FedNLP [26] and FedML [27] provide tools to benchmark FL methods on standard NLP tasks. However, the aforementioned works tend to focus on encoder-only models and classification objectives.

In contrast, text generation in federated settings remains an underexplored area. While language modeling in FL has been explored (e.g., mobile keyboard prediction [28]), few works investigate more sophisticated generative models like BART [7] and its distilled variant, DistilBART [15]. These models have proven effective in a range of centralized NLP tasks such as generation, translation, and comprehension. Their encoder-decoder architecture makes them particularly suitable for both classification (via encoder features) and generation (via decoder outputs), which makes them strong candidates for multi-task learning in FL settings.

To the best of our knowledge, there is no existing research that investigates BART models in federated environments for multi-task NLP applications. In this paper, we aim to explore the federated fine-tuning of pre-trained BART-large and DistilBART models for text classification and generation tasks.

III. Preliminaries

In this section, we provide the necessary background on FL, the BART model, and task formulations, which together form the foundation of our proposed method.

A. Federated Learning

FL is a distributed paradigm in which a set of clients collaboratively train a global model while keeping their private data local. The goal of FL is to optimize the global objective:

$$\min_{w} \mathcal{L}(\cdot) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{L}_n(x_n, y_n; w) \tag{1}$$

where $\mathcal{L}_n(\cdot)$ is the cross-entropy loss on data of client n (x_n, y_n), and the overall objective $\mathcal{L}(w)$ represents the average loss aggregated across all N clients. To better illustrate the optimization process in FL, we now consider a scenario where the system comprises N clients $\{C_1, C_2, \ldots, C_N\}$, each owning a local dataset C_k and maintaining its own model.

$$D_k = \{(x_i^k, y_i^k)\}_{i=1}^{n_k}, \quad \text{where} \quad |D_k| = n_k \tag{2}$$

with the total number of samples $n = \sum_{k=1}^{N} n_k$. Each client maintains a local model with parameters w_t^k at communication round t, where the prediction on input x is denoted as $f(x; w_t^k)$. In contrast, conventional centralized training assumes access to a unified dataset $D_{\text{server}} = \cup_{k=1}^{N} D_k$.

B. BART Model

BART [7] is a sequence-to-sequence pre-trained model designed as a denoising autoencoder. Built upon the transformer encoder–decoder framework [29], the model utilizes a BERT-like bidirectional encoder to capture representations from corrupted text, and a GPT-style autoregressive decoder to regenerate the original document via cross-attention. BART-large consists of 12 encoder and 12 decoder layers (hidden size 1024), with GeLU activation [30]. The pre-training objective minimizes the negative log-likelihood of the original text:

$$\mathcal{L} = -\sum_{t=1}^{T} \log P(x_t \mid x_{<t}, \tilde{X}; w_t) \tag{3}$$

where $X = (x_1, \ldots, x_T)$ is the original sequence, $\tilde{X}$ is the corrupted input, and w_t denotes model parameters. Corruption strategies include token masking, token deletion, text infilling, sentence permutation, and document rotation. With its encoder–decoder design, BART can be fine-tuned for diverse tasks such as sequence classification, token classification, text generation (summarization, question answering), and machine translation.

C. Task Formulations

1) Text Generation: Text generation (also know as decoding) [1] refers to the process of producing tokens conditioned on a given context. Let $X = (x_1, x_2, \ldots, x_S)$, $Y = (y_1, y_2, \ldots, y_T)$ where X is the source context of S tokens and Y is the target sequence of T tokens. The conditional probability of generating Y is

$$P(Y \mid X) = \prod_{t=1}^{T} p(y_t \mid X, y_{<t}) \tag{4}$$

where $y_{<t} = (y_1, \ldots, y_{t-1})$. At each step t, the model $\mathcal{M}$ computes the distribution $p(y_t \mid X, y_{<t})$ over the vocabulary,

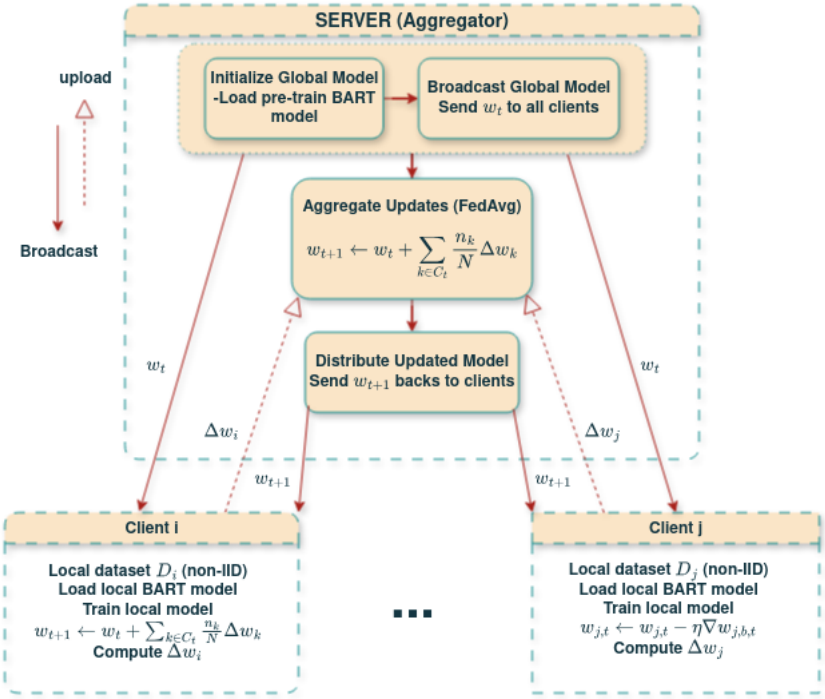

Fig. 1: Overview of the FedAvgBART framework, illustrating the server-client interaction for aggregating model updates in a federated setting.

from which the next token y_t is selected. This iterative autoregressive mechanism [7], [11] enables coherent sequence generation.

2) Text Classification: Let $X = (x_1, x_2, \ldots, x_S)$ denote an input text of length S, and let $Y = \{y_1, \ldots, y_T\}$ be the label set with $|Y| = T$. Each label y_j is represented as a one-hot vector. The goal is to learn a classifier f that estimates the conditional probability $P(y \mid X)$ and assigns the most probable label:

$$P(y \mid X) = f(X), \quad \text{where} \quad y^* = \arg\max_{y \in Y} P(y \mid X) \tag{5}$$

Here, the ground-truth label $y \in Y$ specifies the semantic category of the input.

IV. Methodology

In this section, we first introduce an overview of our FedAvgBART framework. We then explain the training process for the proposed method. At last, fine-tuning FedAvgBART for text classification and generation tasks are discussed.

A. An Overview

Figure 1 presents the FedAvgBART framework, which integrates the BART model into a FL pipeline. The process is divided into three main phases:

1) Broadcast: The central server initializes the global model w_0 by loading a pre-trained BART architecture tailored to the target task. System parameters such as learning rate η and communication rounds are configured. The initialized global model w_t is then broadcast to all participating clients, serving as the starting point for their local models.

2) Upload: Each client C_n loads the BART model and fine-tunes it locally using its private, non-IID dataset D_n. The local training process involves computing gradients and updating the model parameters:

$$w_{t+1}^n \leftarrow w_t - \eta \nabla \mathcal{L}_n(w_t, D_n) \tag{6}$$

where $\mathcal{L}_n$ is the local loss function. After training, the client computes the model update:

$$\Delta w_n = w_{t+1}^n - w_t \tag{7}$$

and uploads Δw_n to the server. This approach allows clients to adapt the BART model to their local data distributions without sharing raw data.

3) Update: The server collects updates $\{\Delta w_1, \Delta w_2, \ldots, \Delta w_N\}$ from all clients and performs FedAvg method [10] to update the global model:

$$w_{t+1} \leftarrow w_t + \sum_{n=1}^{N} \frac{1}{N} \Delta w_n \tag{8}$$

Alternatively, weighted averaging based on dataset sizes $|D_n|$ can be applied. The updated global BART model w_{t+1} is then redistributed to all clients for the next training round. This iterative process continues until convergence or a predefined performance threshold is met.

B. FedAvgBART via Federated Learning

The intuition is that the pre-trained BART, with its encoder-decoder architecture, acts as a unified feature extractor and classifier across clients. Unlike split methods, no parameters

are kept local all are jointly optimized. This can lead to higher communication costs but provides a simpler baseline for homogeneous or mildly heterogeneous settings. The FedAvgBART based training approach for BART pre-training is summarized in Algorithm 1.

Algorithm 1 FedAvgBART Pre-training via FedAvg

Require: Number of communication rounds T, local epochs E, learning rate η, number of clients N.

Ensure: Global model w_T after T communication rounds.

1: **Initialize** global model parameters w_0 (from pre-trained BART).
2: **for** each round $t = 0, 1, \ldots, T-1$ **do**
3: Server selects a subset of clients $\mathcal{S}_t \subseteq \{C_1, \ldots, C_N\}$.
4: **for** each client $C_k \in \mathcal{S}_t$ **in parallel do**
5: Set $w_t^k \leftarrow w_t$.
6: **for** each local epoch $e = 1, \ldots, E$ **do**
7: Compute gradient $\nabla \mathcal{L}_k(w_t^k; D_k)$ on local data D_k.
8: Update parameters:

$$w_t^k \leftarrow w_t^k - \eta \nabla \mathcal{L}_k(w_t^k; D_k)$$

9: Send updated parameters w_t^k to server.
10: Server aggregates updates:

$$w_{t+1} \leftarrow \frac{1}{n} \sum_{k=1}^{N} n_k w_t^k, \quad \text{where} \quad n = \sum_{k=1}^{N} n_k$$

The FedAvgBART algorithm extends the standard FedAvg method to fine-tune the BART model in a distributed and heterogeneous data setting. Initially, the server initializes the global model parameters w_0 from the pre-trained BART. At each communication round t, the server selects a subset of clients to participate in training. Each client C_k downloads the current global model w_t and performs local fine-tuning on its private dataset D_k for E epochs, resulting in updated parameters w_t^k. The updated models are then sent back to the server. The server aggregates the received parameters using a weighted average based on the local dataset sizes n_k of the participating clients:

$$w_{t+1} \leftarrow \frac{1}{n} \sum_{k=1}^{N} n_k w_t^k, \quad \text{where } n = \sum_{k=1}^{N} n_k \tag{9}$$

This process is repeated for T rounds until convergence. The final global model w_T represents the fine-tuned BART model trained in a federated manner, leveraging decentralized and diverse client data while preserving data privacy.

Building on the foundational FedAvgBART pipeline, which enables collaborative fine-tuning across distributed clients, we now delve into the specific adaptations of the BART model for downstream tasks. This ensures that the federated framework leverages BART's strengths in both discriminative and generative paradigms, addressing a range of NLP applications while maintaining data privacy.

C. Fine-tuning FedAvgBART for Text Classification and Generation Tasks

In our FL framework, we fine-tune the pre-trained BART model using the proposed FedAvgBART approach to support both text classification and generation tasks. By leveraging its bidirectional encoder and autoregressive decoder, FedAvgBART effectively adapts to heterogeneous data distributions across clients while maintaining a unified global model. For text classification (e.g., sentiment analysis or topic categorization), the task is reformulated as a sequence-to-sequence generation problem, where the decoder predicts the target label conditioned on the encoded input. For text generation (e.g., summarization or dialogue), the model directly utilizes the encoder to process the input sequence and the decoder to autoregressively generate the corresponding output, aligning naturally with BART's pre-training objective.

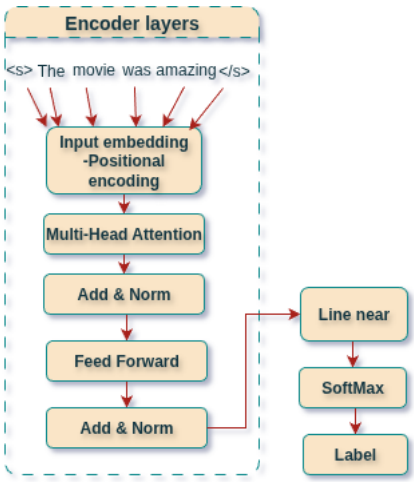

Fig. 2: BART architecture for text classification.

Figure 2 illustrates BART's encoder-based architecture for classification. The tokenized input sequence (e.g., "<s> *The movie was amazing* </s>") is first mapped through an embedding layer with positional encoding, then sequentially processed by multi-head self-attention, residual connections with layer normalization, and a feed-forward network, yielding contextualized hidden representations. These encoder representations are then passed to a linear projection layer, followed by Softmax normalization, to produce the final classification label. Unlike the full seq2seq generation framework, this setup employs only the encoder for representation learning, directly mapping input text into class probabilities. By this way, we can leverage BART's pre-trained bidirectional context modeling for downstream classification tasks, making it effective even on heterogeneous or non-IID data in FL scenarios.

Shifting to generation tasks, BART excels in producing coherent and contextually relevant text outputs, such as summarization, translation, or conditional text completion.

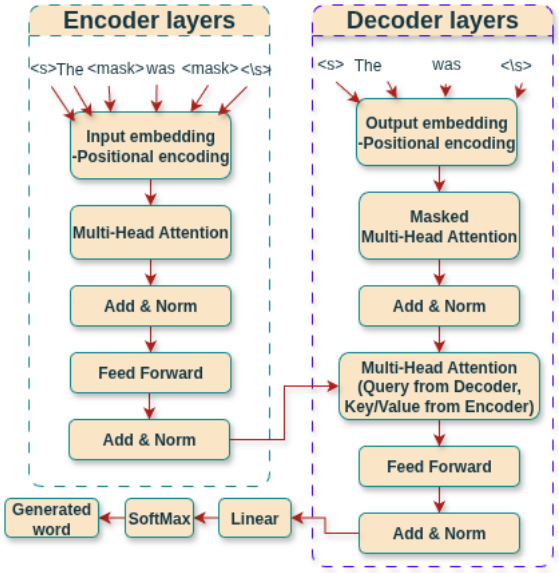

Fig. 3: BART architecture for text generation.

Figure 3 depicts the sequence-to-sequence generative architecture of BART. The encoder processes the tokenized input (e.g., "*<s> The <mask> was <mask> </s>*") through an embedding layer with positional encoding, followed by multi-head self-attention, residual connections with layer normalization, and a feed-forward network, yielding contextualized hidden representations. These encoder representations are transmitted to the decoder, where the partially observed target sequence (e.g., "*<s> The ... was ... </s>*") is first embedded with positional information and passes through masked multi-head self-attention and normalization. Subsequently, the decoder performs cross-attention, attending to the encoder's hidden states, before applying an additional feed-forward transformation and normalization. The decoder outputs are projected through a linear transformation and normalized via Softmax to produce the next token in an autoregressive manner. This enables the encoder to capture bidirectional contextual dependencies, while the decoder generates the output sequence incrementally, token by token.

By integrating these task-specific adaptations into FedAvg-BART, the framework not only handles classification through discriminative generation but also supports pure generative outputs, ensuring robustness across varying data distributions. This dual capability facilitates seamless transitions between tasks in federated environments, with future extensions potentially exploring hybrid classification-generation objectives to further mitigate heterogeneity challenges.

V. Experiments

In this section, we conduct experiments led by the following research questions (RQ):

- RQ1 (model size): What is the impact of model size on the efficiency and performance of federated training of BART-based models (DistilBART and BART-large) for NLP tasks involving classification and generation?
- RQ2 (non-IID data): How does non-IID data distribution across clients affect the performance of DistilBART and BART-large in federated settings for text classification and generation tasks?
- RQ3 (number of clients): How does the number of clients influence the performance of federated fine-tuning of pre-trained models such as DistilBART and BART-large on text classification and generation tasks?

A. Datasets, Tasks, and Models

We used two datasets: the 20News dataset [32] for classification and the CNN/DailyMail dataset [33] for generation. The models used are BART-large [7] and DistilBART [15].

1) 20News Dataset: The 20News dataset [32] is a collection of approximately 20,000 newsgroup documents, partitioned (nearly) evenly across 20 different newsgroups. It is a widely used benchmark for text classification tasks due to its diverse topics and relatively balanced class distribution. Each document belongs to one of 20 categories, such as 'alt.atheism', 'comp.graphics', 'rec.sport.baseball', and 'talk.politics.mideast'. The dataset is preprocessed to remove headers, footers, and quotes, focusing on the core content for classification. In our experiments, we utilize 11,314 training samples and 7,532 validation samples.

2) CNN/DailyMail Dataset: The CNN/DailyMail dataset [33] is a popular benchmark for text summarization and generation tasks. It consists of news articles (from CNN and Daily Mail) paired with multi-sentence summaries written by journalists. The task is to generate a concise and coherent summary given the full news article. We use a subset of this dataset, comprising 10,000 training samples and 5,000 validation samples, to evaluate the text generation capabilities of our models.

B. Evaluation Metrics

We use the following metrics for evaluation:

- Accuracy (Acc): The proportion of correct predictions among the total number of cases examined [31].

$$\text{Acc} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{10}$$

- Precision (Prec), Recall (Rec), and F_1-score: Metrics for classification performance [31].
  - Prec: The ratio of true positives (TP) to all positive predictions (TP + FP).

$$\text{Prec} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{11}$$

  - Rec: The ratio of true positives (TP) to all actual positives (TP + FN).

$$\text{Rec} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{12}$$

  - F_1-score: The harmonic mean of precision and recall.

$$\text{F}_1 = 2 \times \frac{\text{Prec} \times \text{Rec}}{\text{Prec} + \text{Rec}} \tag{13}$$

- ROUGE: A standard metric for evaluating text summarization and generation [34]. We report ROUGE-1 (unigram overlap), ROUGE-2 (bigram overlap), and ROUGE-L (longest common subsequence). Here, *gen* denotes the generated summary, *ref* the reference summary, and *n-gram* a contiguous sequence of n words.
  - ROUGE-N: Measures the recall of *n-gram* overlap between *gen* and *ref*:

$$ROUGE - N = \frac{\sum_{\text{n-gram}} \min(\text{Count}_{\text{gen}}, \text{Count}_{\text{ref}})}{\sum_{\text{n-gram}} \text{Count}_{\text{ref}}} \tag{14}$$

  - ROUGE-L: Based on the length of the longest common subsequence (LCS) between *gen* and *ref*:

$$ROUGE - L = \frac{\text{LCS}(\text{gen}, \text{ref})}{|\text{ref}|} \tag{15}$$

C. Experimental Setup

1) Implementation: All the architectures are implemented in Pytorch. FL simulations are done with the FedAvg algorithm in a cross-device setting. Client data heterogeneity is controlled using Dirichlet partitioning with concentration parameter $\alpha \in \{0.1, 0.5\}$, where a smaller α indicates more non-IID data. Experiments were executed on Nvidia GPUs.

2) Hyper Parameters: The models were fine-tuned for a set number of rounds: 22 for the 20News dataset and 5 for CNN/DailyMail. The number of clients varied from 2 to 10. Performance of BART-large and DistilBART was compared in both centralized and federated settings. For text classification, the number of clients ranged from 2 to 10, while for text generation it ranged from 2 to 5, allowing us to observe the effect on model performance for both tasks. To simulate non-IID data distributions, a Dirichlet distribution with α values of 0.1 (highly non-IID) and 0.5 (more IID-like) was employed, and the results were compared against IID settings.

D. Results and Discussion

1) Impact of Model Size on Federated Training (RQ1): Table I provides detailed results for RQ1 regarding text classification. Our experiments reveal contrasting behaviors between the two architectures under federated training. For classification, BART-large achieves significant improvements with 0.778 accuracy and 0.782 F_1 score in federated training, compared to 0.743 accuracy and 0.746 F_1 in centralized training (+0.04 accuracy, +0.04 F_1). In contrast, DistilBART shows a slight degradation in classification performance, with 0.738 accuracy and 0.738 F_1 in federated settings versus 0.739 accuracy and 0.740 F_1 centralized (-0.002 accuracy, -0.001 F_1). For text generation, both models show substantial improvements across all ROUGE metrics under federated training (Table II). BART-large shows consistent performance across different data distributions, with ROUGE-1 scores of 42.2-43.4, ROUGE-2 scores of 21.1-22.0, and ROUGE-L scores of 30.7-31.7. DistilBART demonstrates stronger overall performance in text generation, particularly in the non-IID setting, with ROUGE-1 scores of 43.0-48.9, ROUGE-2 scores of 20.8-30.5, and ROUGE-L scores of 30.8-38.7. The most significant improvements are seen in the non-IID setting for text generation, where DistilBART's ROUGE-2 score of 30.5 is 38.7% higher than BART-large's ROUGE-2 score of 22.0, highlighting its effectiveness in heterogeneous data environments. Figure 4 shows the F_1 comparison for classification.

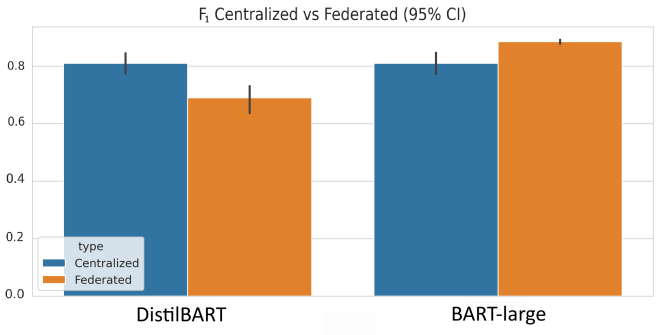

Fig. 4: Performance comparison for classification (F_1-Score).

TABLE I: Classification performance: Centralized vs. Federated.

Model	Centralized		Federated		Improvement	
	Acc	F_1	Acc	F_1	ΔAcc	ΔF_1
BART-large	0.743	0.746	**0.778**	**0.782**	+0.03	+0.04
DistilBART	**0.739**	**0.740**	0.738	0.738	-0.002	-0.001

TABLE II: Text generation performance: Centralized vs. Federated.

Model	Metric	Centralized	Federated	Improv.
BART-large	ROUGE-1	41.5	**42.2**	+0.7
	ROUGE-2	19.5	**20.6**	+1.1
	ROUGE-L	28.7	**29.5**	+0.8
DistilBART	ROUGE-1	40.9	**43.3**	+2.4
	ROUGE-2	18.9	**20.9**	+2.0
	ROUGE-L	28.4	**30.4**	+2.0

2) Influence of Non-IID Data Distribution (RQ2): The influence of non-IID data reveals interesting patterns across different model architectures. For text generation (Table IV), DistilBART shows superior performance in the highly non-IID setting ($\alpha = 0.1$) with ROUGE scores of 48.9/30.5/38.7 (R-1/R-2/R-L), compared to 43.0/20.8/30.8 in the more IID-like setting ($\alpha = 0.5$). This 5.9-point improvement in ROUGE-1 suggests that the distilled model can effectively leverage data diversity. In contrast, BART-large shows a different pattern, with slightly better ROUGE-2 and ROUGE-L scores in the non-IID setting but a 1.2-point drop in ROUGE-1. The models also differ in their loss patterns, with DistilBART achieving lower loss values (0.893 vs 1.113 for BART-large in the non-IID setting), indicating more stable training. These results suggest that the relationship between model size, data distribution, and task performance in FL is complex and warrants further investigation. For classification performance under non-IID conditions, see Tables III and V, and Figure 7 for detailed comparisons.

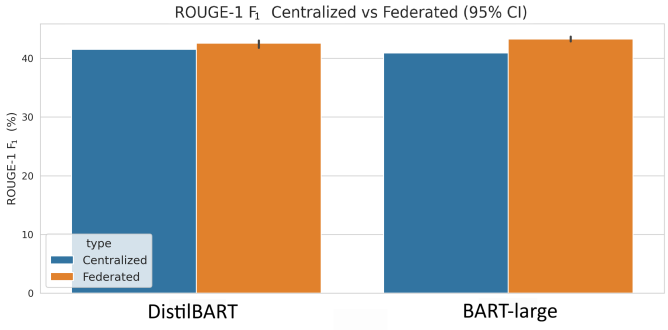

Fig. 5: Performance comparison for text generation (ROUGE-1 Score).

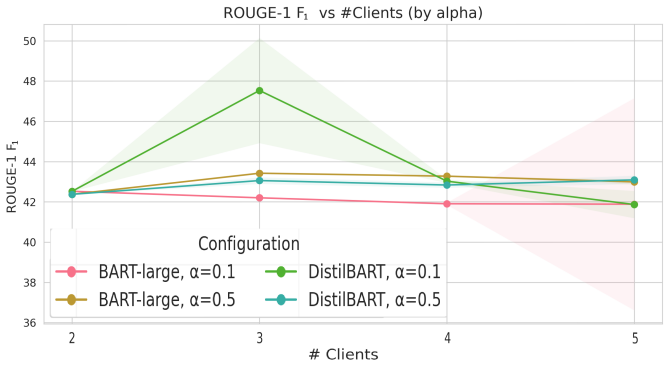

Fig. 6: ROUGE-1 F_1 vs number of clients for DistilBART and BART-large. Smaller α indicates more non-IID data.

TABLE III: BART-large text classification federated best-round results across client counts, separated by data partition parameter α (**bold** = best per column; for Loss, lower is better).

α=0.1 **(non-IID)**						
Clients	Round	Acc	F_1	Prec	Rec	Loss
2	22	0.9677	0.9722	0.9744	0.9715	0.0987
3	22	0.9698	0.9735	0.9753	0.9730	0.0959
4	22	0.9724	0.9515	0.9531	0.9515	0.0825
5	21	0.9697	0.9654	0.9676	0.9662	0.0993
6	21	**0.9756**	0.9807	0.9822	0.9806	**0.0724**
7	21	0.9729	0.9787	0.9793	0.9792	0.0876
8	22	0.9711	0.9681	0.9716	0.9669	0.0873
9	22	0.9660	0.9736	0.9746	0.9734	0.1019
10	22	0.9704	**0.9817**	**0.9839**	**0.9818**	0.1031
α=0.5 **(IID)**						
Clients	Round	Acc	F_1	Prec	Rec	Loss
2	22	0.9658	**0.9723**	**0.9737**	**0.9721**	0.1090
3	21	**0.9679**	0.9423	0.9546	0.9375	**0.1110**
4	22	0.9630	0.9596	0.9611	0.9597	0.1146
5	22	0.9584	0.9707	0.9717	0.9705	0.1366
6	22	0.9581	0.9629	0.9650	0.9629	0.1354
7	22	0.9462	0.9513	0.9562	0.9514	0.1598
8	22	0.9597	0.9626	0.9657	0.9617	0.1344
9	22	0.9407	0.9567	0.9569	0.9581	0.2088
10	20	0.9512	0.9709	0.9711	0.9718	0.1629

Figure 5 illustrates that DistilBART generally outperforms BART-large in ROUGE-1 scores for text generation, especially under non-IID data conditions. DistilBART achieves a peak ROUGE-1 score of 48.9, significantly higher than BART-large's 42.2, suggesting its superior effectiveness in heterogeneous federated environments for generative tasks.

Figure 8 illustrates the per-client data share as 100%-stacked bars for each federation size *N*, where each bar aggregates clients *1..N* as a percentage of total samples, averaged across multiple runs, considering data heterogeneity, communication cost, scalability, and robustness.

3) Effect of Client Population on Federated Fine-Tuning (RQ3): Text generation results show distinct client scaling patterns. As seen in Figure 6, DistilBART achieves strong performance with a peak ROUGE-1 score of 48.9 under non-IID conditions ($\alpha = 0.1$), while BART-large shows more modest performance with 42.2 ROUGE-1. The performance gap between the models is particularly notable in the non-IID setting, where DistilBART demonstrates superior performance across all ROUGE metrics (30.5 ROUGE-2 and 38.7 ROUGE-L) compared to BART-large (22.0 ROUGE-2 and 31.7 ROUGE-L). This suggests that the distilled model may be more effective at handling the complexities of federated text generation tasks, especially in heterogeneous data environments. Table III and V provide detailed results across different client configurations.

The text generation results in Tables VI and VII reveal distinct patterns in how model architectures respond to data heterogeneity and client scaling. For BART-large, performance peaks at 5 clients in the non-IID setting ($\alpha = 0.1$) with ROUGE scores of 44.57/23.28/30.05, while the IID setting ($\alpha = 0.5$) achieves optimal performance at 3 clients with 43.43/21.12/30.66. This suggests that BART-large benefits from increased client diversity in heterogeneous environments but requires fewer coordination points in homogeneous settings.

DistilBART exhibits a more pronounced sensitivity to data distribution, achieving exceptional performance at 3 clients in the non-IID setting with remarkable ROUGE scores of 48.87/30.48/38.72—significantly outperforming BART-large across all metrics. However, in the IID setting, DistilBART's performance converges to similar levels as BART-large (43.39/21.06/30.67 at 3 clients), indicating that the distilled model's advantage emerges specifically from its ability to leverage data heterogeneity. The convergence patterns also differ, with DistilBART requiring fewer rounds in IID settings (round 1 vs round 2-3) but maintaining consistent performance across rounds in non-IID environments, suggesting more stable training dynamics under data heterogeneity.

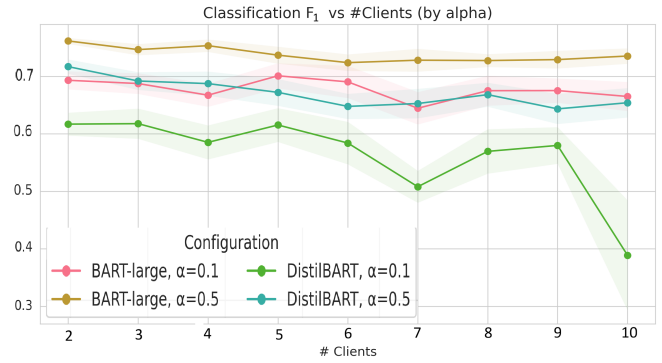

Fig. 7: Classification F_1 vs number of clients. BART-large (IID) is stable, while DistilBART's performance varies with data heterogeneity (α).

TABLE IV: Federated text generation: IID vs. Non-IID comparison. Best-round performance on CNN/DailyMail showing impact of data distribution (**bold** = best per column; for Loss, lower is better).

Model	α	ROUGE-1	ROUGE-2	ROUGE-L	Loss
BART-large	0.1	42.20	**21.98**	31.66	**1.113**
BART-large	0.5	**43.42**	21.13	30.66	2.229
DistilBART	0.1	**48.86**	**30.49**	**38.73**	**0.893**
DistilBART	0.5	43.00	20.84	30.84	1.615

TABLE V: DistilBART text classification federated best-round results across client counts, separated by data partition parameter α (**bold** = best per column; for Loss, lower is better).

α=0.1 **(non-IID)**						
Clients	Round	Acc	F_1	Prec	Rec	Loss
2	20	0.9730	**0.9771**	0.9811	**0.9758**	0.0862
3	19	**0.9765**	0.9732	0.9801	0.9693	**0.0696**
4	21	0.9756	0.9259	0.9914	0.8817	0.0763
5	21	0.9731	0.9157	0.9720	0.8895	0.0906
6	22	0.9763	0.8570	0.9879	0.7651	0.1018
7	22	0.9725	0.8600	**0.9959**	0.7883	0.0739
8	22	0.9732	0.8995	0.9862	0.8430	0.0953
9	22	0.9747	0.9719	0.9907	0.9587	0.0859
10	5	0.9111	0.3324	0.5451	0.2514	0.5470
α=0.5 **(IID)**						
Clients	Round	Acc	F_1	Prec	Rec	Loss
2	22	0.9731	**0.9753**	**0.9908**	0.9633	0.0899
3	22	**0.9749**	0.9673	0.9781	0.9624	**0.0849**
4	20	0.9717	0.9413	0.9751	0.9278	0.0988
5	22	0.9681	0.9630	0.9704	0.9604	0.1062
6	20	0.9629	0.9586	0.9801	0.9464	0.1314
7	22	0.9554	0.9529	0.9689	**0.9676**	0.1486
8	21	0.9668	0.9704	0.9765	0.9676	0.1169
9	22	0.9356	0.9139	0.9568	0.8866	0.2430
10	20	0.9474	0.9264	0.9525	0.9073	0.2056

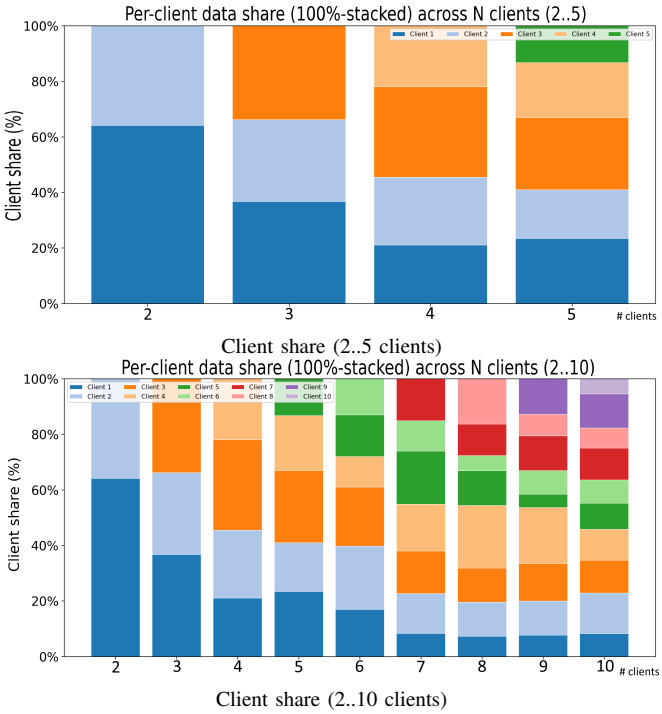

Fig. 8: Per-client data share as 100%-stacked bars for each federation size N, averaged across runs.

TABLE VI: DistilBART text generation federated best-round ROUGE results across client counts, separated by data partition parameter α (**bold** = best per column).

$\alpha = 0.1$ **(non-IID)**				
Clients	Round	ROUGE-1	ROUGE-2	ROUGE-L
2	2	42.5291	20.1340	29.7839
3	2	**48.8694**	**30.4841**	**38.7152**
4	2	43.1349	18.7339	28.6039
5	3	42.3518	21.0916	30.1326
$\alpha = 0.5$ **(IID)**				
Clients	Round	ROUGE-1	ROUGE-2	ROUGE-L
2	1	42.3780	20.0847	29.6427
3	1	**43.3872**	**21.0637**	**30.6698**
4	1	43.1024	20.4452	30.0926
5	3	43.2821	20.8961	30.3861

TABLE VII: BART-large text generation federated best-round ROUGE results across client counts, separated by data partition parameter α (**bold** = best per column).

$\alpha = 0.1$ **(non-IID)**				
Clients	Round	ROUGE-1	ROUGE-2	ROUGE-L
2	2	42.5276	20.1377	29.7835
3	2	42.2042	21.9767	**31.6594**
4	2	41.9328	19.1366	28.3490
5	2	**44.5677**	**23.2808**	30.0505
$\alpha = 0.5$ **(IID)**				
Clients	Round	ROUGE-1	ROUGE-2	ROUGE-L
2	2	42.3843	20.0745	29.6515
3	2	**43.4289**	**21.1218**	**30.6572**
4	2	43.2763	20.6950	30.3264
5	2	43.0532	20.6097	30.1883

VI. Conclusion

In this paper, we proposed a novel approach, which integrates FL with pre-trained BART models for text classification and generation tasks. This is a promising solution for privacy preservation and collaborative learning across multiple clients. We evaluated the effectiveness and efficiency of this approach on benchmark datasets using BART-large and DistilBART models. The experimental results demonstrated that federated training can surpass centralized fine-tuning in performance. We uncover a task-architecture alignment: BART-large exhibits exceptional proficiency in classification tasks, while DistilBART excels in text generation, offering superior computational efficiency for resource-limited clients. Furthermore, BART-large maintains greater stability across diverse client scales, whereas non-IID data disproportionately affects smaller models, underscoring the robustness of larger architectures. These findings redefine the efficiency-robustness trade-off, offering a theoretical and practical foundation for tailoring model architectures to the unique dynamics of federated NLP environments.

In future work, we plan to explore the following four directions. First, investigating more advanced FL algorithms beyond FedAvg, such as FedProx or SCAFFOLD [35], may yield better performance under high data heterogeneity by addressing client drift and improving convergence stability. Second, combining distillation with other model compression techniques like quantization and pruning could further reduce computational and communication overhead while maintaining model performance. Third, applying our framework to other complex NLP tasks and domains, such as biomedical text processing or multilingual scenarios, would demonstrate broader applicability. Finally, incorporating formal privacy guarantees through differential privacy or secure aggregation would strengthen the privacy-preserving aspects of federated fine-tuning for sensitive applications.

References

[1] B. Liu, "Sentiment analysis and opinion mining," *Communications of the ACM*, vol. 64, no. 2, pp. 62–71, 2021.

[2] R. Agarwal, A. Dhoot, S. Kant, V. S. Bisht, H. Malik, M. F. Ansari, A. Afthanorhan, and M. A. Hossaini, "A novel approach for spam detection using natural language processing with AMALS models," *IEEE Access*, vol. 12, pp. 124298–124313, 2024.

[3] M. Hasan, T. Ahmed, M. R. Islam, and M. P. Uddin, "Leveraging textual information for social media news categorization and sentiment analysis," *PLOS ONE*, vol. 19, no. 7, p. e0307027, 2024.

[4] C. Raffel *et al.*, "Exploring the limits of transfer learning with a unified text-to-text transformer," *Journal of Machine Learning Research*, vol. 21, pp. 1–67, 2020.

[5] Quang-Hung, L. & Anh-Cuong, L. Syntactic pattern based Word Alignment for Statistical Machine Translation. *International Journal of Knowledge and Systems Science (IJKSS)*. **5**, 36-45 (2014).

[6] Quang-Hung, L. & Anh-Cuong, L. Improving Word Alignment for Statistical Machine Translation Based on Constraints. *2012 International Conference on Asian Language Processing*. pp. 113-116 (2012).

[7] M. Lewis *et al.*, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proceedings of ACL*, 2020, pp. 7871–7880.

[8] General Data Protection Regulation (GDPR), Accessed: Aug. 31, 2025. [Online]. Available: https://gdpr-info.eu

[9] H. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of AISTATS*, 2017, pp. 1273–1282.

[10] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–19, 2019.

[11] J. Devlin, M. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of deep bidirectional transformers for language understanding," in *Proceedings of NAACL*, 2019, pp. 4171–4186.

[12] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[13] X. Jiao *et al.*, "TinyBERT: Distilling BERT for natural language understanding," in *Findings of EMNLP*, 2020, pp. 4163–4174.

[14] Z. Sun *et al.*, "MobileBERT: a compact task-agnostic BERT for resource-limited devices," in *Proceedings of ACL*, 2020, pp. 2158–2170.

[15] S. Shleifer and A. Rush, "Pre-trained summarization distillation," *arXiv preprint arXiv:2010.13002*, 2020.

[16] J. Xu and K. Lee, "Performance and communication cost of deep neural networks in federated learning environments," *IEEE Transactions on Parallel and Distributed Systems*, 2024.

[17] Y. Zhao, M. Li, Y. Lai, N. Suda, D. Civin, and V. Chandra, "Federated learning with non-IID data," *arXiv preprint arXiv:1806.00582*, 2018.

[18] S. He and Y. Wang, "A survey on federated fine-tuning of large language models," *arXiv preprint arXiv:2507.xxxxx*, 2025.

[19] Z. Chen, L. Xu, and J. Li, "Communication-efficient and tensorized federated fine-tuning of large language models," *arXiv preprint arXiv:2508.xxxxx*, 2025.

[20] I. Dayan *et al.*, "Federated learning for predicting clinical outcomes in patients with COVID-19," *Nature Medicine*, vol. 27, no. 10, pp. 1735–1743, 2021.

[21] G. Long, Y. Tan, J. Jiang, and C. Zhang, "Federated learning for open banking," in *Federated Learning: Privacy and Incentive*, Springer, 2020, pp. 240–254.

[22] J. Lee and S. Park, "Federated Freeze BERT for text classification," *IEEE Transactions on Neural Networks and Learning Systems*, 2024.

[23] M. Lee and J. Cho, "Federated Split BERT for heterogeneous text classification," in *Proceedings of COLING*, 2022.

[24] Y. Liu *et al.*, "RoBERTa: A robustly optimized BERT pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[25] L. Chen and H. Yang, "FedBERT: When federated learning meets pre-training," in *Proceedings of EMNLP*, 2022.

[26] T. Lin, L. Kong, S. Liu, M. Hong, and H. Yang, "FedNLP: Benchmarking federated learning methods for natural language processing tasks," in *Proceedings of the EMNLP Workshop*, 2022.

[27] C. He *et al.*, "FedML: A research library and benchmark for federated machine learning," *arXiv preprint arXiv:2007.13518*, 2020.

[28] A. Hard *et al.*, "Federated learning for mobile keyboard prediction," *arXiv preprint arXiv:1811.03604*, 2018.

[29] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[30] D. Hendrycks and K. Gimpel, "Gaussian error linear units (GELUs)," *arXiv preprint arXiv:1606.08415*, 2016.

[31] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[32] K. Lang, "Newsweeder: Learning to filter netnews," in *Proceedings of the 12th International Conference on Machine Learning*, 1995, pp. 331–339.

[33] K. Hermann *et al.*, "Teaching machines to read and comprehend," in *Advances in Neural Information Processing Systems*, 2015, pp. 1693–1701.

[34] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of ACL Workshop*, 2004, pp. 74–81.

[35] S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic controlled averaging for on-device federated learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.