

Investigating Edge Fine-Tuning of Large Language Models in a Federated Environment

Lorenzo Colombi, Michela Vespa, Francesco Resca, Sara Cavicchi, Edoardo Di Caro, Elena Bellodi, Mauro Tortonesi and Cesare Stefanelli

University of Ferrara, Ferrara, Italy

Abstract

Large Language Models (LLMs) are increasingly adopted in edge applications, with notable advantages compared to cloud-based approaches, such as enhanced privacy, reduced latency, and improved energy efficiency. While deploying pre-trained models at the edge is common, training or fine-tuning them locally poses significant challenges due to data remaining on-device in distributed, heterogeneous environments and the ones related to computational constraints and communication overhead. In this context, a well-established solution is Federated Learning (FL), where clients train models locally, without sharing their data, and a global model is obtained by aggregating their parameters, weighted by the number of examples. This method may be inadequate for LLMs, as it overlooks the varying information content of training examples. In fact, longer sequences often contain more informative structures, offering richer learning signals. To investigate this issue, we evaluate the training performance of LLMs in a federated learning setting using two aggregation methods: standard FedAvg and a token-based variant that weights updates based on the number of tokens processed locally. We conduct experiments using lightweight LLMs, specifically SmollM2, comparing performance using different open-source datasets from the healthcare field. Experimental results demonstrate that token-based FedAvg reaches the performance of standard FedAvg and, in some cases, slightly surpasses it.

Keywords

Large Language Model, Federated Learning, Machine Learning, Edge Computing

1. Introduction

Machine Learning (ML) solutions are being increasingly used to streamline and enhance processes across a wide range of application domains [1, 2, 3]. These include social networks [4], healthcare, energy management [5], and smart manufacturing [6, 7]. However, effectively developing and deploying ML services presents several challenges [8] related to both functional (e.g., model performance, robustness to contextual changes) and non-functional aspects, which include privacy, confidentiality, fairness, and explainability [9].

Within this context, the introduction of Transformer-based Large Language Models (LLMs) [10] has fundamentally transformed the way humans interact with machines and external information. Leveraging the self-attention mechanism, these models excel at capturing intricate semantic relationships within text, enabling them to understand and generate human-like language with remarkable coherence and contextual awareness. Unlike earlier models, LLMs can process extended sequences of text, making them especially effective for tasks such as summarization, translation, configuration generation [11], and conversational Artificial Intelligence (AI).

These models, sometimes consisting of millions or billions of parameters, are typically trained on large text corpora in data centers and are deployed exclusively in cloud environments due to their substantial computational requirements. However, as LLM-powered applications gain popularity across diverse environments, and with the recent advancements in model optimization and the development of smaller architectures by both academia and industry, there is a growing trend toward deploying

ITADATA2025: The 4th Italian Conference on Big Data and Data Science, September 9–11, 2025, Turin, Italy

 lorenzo.colombi@unife.it (L. Colombi)

 0009-0004-9681-9842 (L. Colombi); 0009-0004-4350-8151 (M. Vespa); 0009-0004-9900-717X (F. Resca); 0009-0002-0771-0067 (S. Cavicchi); 0009-0003-3740-5204 (E. Di Caro); 0000-0002-3717-3779 (E. Bellodi); 0000-0002-7417-4455 (M. Tortonesi); 0000-0003-4617-1836 (C. Stefanelli)



© 2025 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

them at the edge [12].

This shift enables more applications, for example, in the healthcare and smart city fields, where aggregating all collected data in a central location is not possible for different reasons (e.g., privacy, lack of connectivity, limited bandwidth) and at the same time preserves privacy, reduces costs, and improves energy efficiency.

Notably, despite many applications only requiring the deployment of a pre-trained model in an edge environment, significant challenges arise when the models must be trained or fine-tuned in such a distributed and heterogeneous setting, where data must remain on-device. A promising and well-studied solution to this issue in Federated Learning (FL), where each client trains its model and then a global model is obtained by the aggregation of the weights [13]. For example, one of the most used aggregation techniques is Federated Averaging (FedAvg) [14], where the global model is the result of the average of the clients' parameters, weighted by the number of examples.

However, this simple aggregation technique could be limiting when applied to LLM fine-tuning, because it does not reflect the information content of each example. In LLMs, longer sequences often encode more complex and informative linguistic structures, providing a richer learning signal. By integrating token counts, other than the number of examples, into the aggregation function, token-based FedAvg variation accounts for this variability, promoting more balanced and performance-aligned model updates. This improved fine-tuning accuracy could be particularly beneficial in scenarios where data is highly heterogeneous, having large variability in example length.

Healthcare is a particularly interesting scenario in which smarter distributed LLMs training methods could have a significant impact. In fact, health applications are characterized by vast amounts of sensitive textual data of highly varying length, such as electronic health records, clinical notes, and patient reports that cannot be shared across hospitals due to privacy regulations and institutional policies - with longer length documents often containing a larger informative content [15].

In this context, FL combined with LLMs offers a powerful solution, enabling collaborative learning across institutions without breaching patient privacy. Hospitals can jointly train language models for tasks such as clinical note analysis or diagnostic support while keeping data strictly on-premise [16]. This setup aligns with stringent regulatory frameworks like General Data Protection Regulation (GDPR) and Health Insurance Portability and Accountability Act (HIPAA), while also benefiting from a pooled knowledge base that enhances model quality and robustness [17].

Small Language Models (SLMs) are particularly advantageous because they can be hosted on existing hospital IT infrastructure, enabling low-latency, offline inference during clinical workflows, all while minimizing reliance on external cloud services. Furthermore, federated collaboration effectively increases the diversity of training data, improving generalization for rare diseases or diverse patient populations that a single institution might not cover alone. Importantly, a federated approach can unlock insights from prestigious medical centers whose data would otherwise remain siloed, thereby boosting predictive performance for all participants [18]. Smaller clinics and hospitals also benefit, as SLMs lower the computational barrier, allowing participation in federated training without the need for extensive hardware resources.

Nonetheless, medical data poses unique challenges due to its heterogeneity. Different hospitals use various Electronic Health Record (EHR) systems, data schemas, and serve diverse patient populations. This variability can cause federated models to converge more slowly or develop biases toward larger institutions' data. Strategies such as weighted averaging and domain-specific fine-tuning are therefore crucial to address these issues and ensure equitable model performance across all participating sites.

In this paper, we evaluate the training performance of LLMs in a federated learning setting using two parameter aggregation methods: the standard FedAvg and a weighted variant that accounts for the number of tokens processed by each client during local training. To address the computational constraints typical of edge devices, we employ Parameter Efficient Fine-Tuning (PEFT) techniques, which enable more efficient fine-tuning of Transformer-based models like LLMs. This token-based weighting accounts for the fact that, in the context of LLMs, larger textual examples often provide richer learning signals than smaller ones. Additionally, to support scalability and heterogeneity in real-world federated environments, our experiments focused on low-parameter LLMs, which could also

be deployed and fine-tuned on edge devices.

We conducted a series of preliminary experiments using open-source datasets from the healthcare field and two versions of the SmollM2 model, a state-of-the-art low-parameter LLM, specifically chosen for use in edge environments without requiring specialized hardware to better simulate real-world conditions.

We measured the performance increase obtained through PEFT in various tasks, utilizing publicly available datasets focusing on a conversational healthcare scenario.

We empirically compared the federated PEFT performance using the aforementioned token-based FedAvg against the standard FedAvg aggregation method, as well as against centralized training. Our initial results show that token-based FedAvg matches—and in some cases slightly exceeds—the performance achieved in both the centralized and FedAvg settings. However, the observed performance gains were marginal, indicating the need for further experimentation, particularly with datasets featuring more diverse example lengths.

The paper is structured as follows: Section 2 reviews background and related work. Section 3 presents the proposed *token-based Federated Averaging* scheme and its aggregation rule. Section 4 reports the experimental evaluation: datasets (Sec. 4.1), environment (Sec. 4.2), and results (Sec. 4.3). Section 5 concludes and outlines directions for future work.

2. Background and Related Works

2.1. Federated Learning

FL is an ML paradigm designed to enable collaborative model training while addressing data silos and preserving data privacy [14]. Since Google introduced the concept in 2017, numerous studies have focused on improving FL in terms of training accuracy, speed, fault tolerance, and efficiency. In addition, novel architectures [19] and aggregation techniques have been proposed. The novelty and advantage of FL lie in its ability to perform local training on edge devices without requiring the transfer of sensitive data, thereby preserving user privacy. Furthermore, FL is particularly beneficial when centralized data collection is impractical, such as in healthcare, smart cities, and Industry 4.0/5.0 applications [20].

In the FL framework, multiple clients (e.g., mobile devices, institutions, or organizations) collaborate with a central server to perform decentralized ML [21]. Each client downloads an initial global model from the server and trains it using its local data. The server then aggregates the model parameters received from all clients to update the global model for the next training round. The most widely adopted aggregation algorithm is FedAvg, which computes a weighted average of the clients' model parameters, where each weight corresponds to the proportion of training samples at that edge location [14]. For simplicity, the following formulation omits the explicit inclusion of the learning rate, as it is applied during each client's local training phase:

$$\mathbf{w}_{\tau+1} = \sum_{k=1}^K \frac{n_k}{N} \mathbf{w}_\tau^{(k)} \quad (1)$$

where:

\mathbf{w}_τ are the global model weights at round τ ,

$\mathbf{w}_\tau^{(k)}$ are the model weights from client k after local training in round τ ,

n_k is the number of local data samples used by client k ,

$N = \sum_{k=1}^K n_k$ is the total number of samples across all K participating clients,

K is the number of clients involved in the current communication round.

Since its initial introduction, several aspects of the FedAvg algorithm, including the objective function, learning rate, and weighting scheme, have been modified to enhance FL performance and robustness. For example, FedProx [22] incorporates a proximal term in the objective function to pull the local model closer to the global model, thereby enhancing stability and providing convergence guarantees. Another notable strategy is FedNova, which addresses non-IID-ness by normalizing and scaling each client's updates based on its local iteration count before updating the global model [22].

2.2. Large Language Models

A novel area of research aims at integrating the characteristics of the FL paradigm with the capabilities of LLMs, ensuring collaborative model training without the need to share private data. LLMs are neural language models built on transformer architectures, usually containing tens to hundreds of billions of parameters and pre-trained on extensive text datasets [23]. LLMs leverage self-attention mechanisms [24] to process entire sequences simultaneously, thereby removing the sequential processing bottleneck, and greatly enhancing scalability and performance in language tasks.

Training LLMs typically includes two main stages: pre-training on extensive text corpora to learn grammar, facts, general knowledge, and language comprehension abilities, followed by task-specific fine-tuning using smaller datasets to adapt the model to specific applications. Nevertheless, fine-tuning LLMs typically requires sharing vast amounts of data, raising significant privacy and regulatory concerns, particularly in sensitive domains such as healthcare, finance, and legal services [25]. A promising approach to preserving sensitive information is shifting LLMs fine-tuning to the edge. Moreover, this transition would reduce reliance on constant connectivity, as edge-deployed LLMs can operate even without communication capabilities. Additionally, performing fine-tuning or inference locally can decrease latency and bandwidth usage, resulting in an improved user experience and faster response times. Another significant advantage is the ability to personalize edge-based LLMs better to reflect individual user preferences and behavior [12]. Moreover, using FL techniques the global model could be updated by aggregating the parameters of the local models. However, this shift is challenged by the massive size and computational demands of these models, along with the inherent limitations of edge devices, since efficient fine-tuning in edge environments still requires reducing model size or optimizing parameter updates to avoid overloading resource-constrained devices, without compromising performance [26].

To address these challenges, many solutions have been developed. The first is PEFT, which updates only a small subset of model parameters while keeping the rest frozen. Unlike conventional full-parameter fine-tuning, which can have high computation and communication costs due to the sheer size of LLMs, PEFT reduces overhead and enables effective model adaptation to new tasks, even within the constraints of edge devices. One of the most well-known PEFT techniques is Low-Rank Adaptation (LoRA), which freezes the original model weights and injects trainable low-rank matrices into each layer of the Transformer architecture. Other notable methods include prompt tuning, which prepends tunable virtual tokens to the input sequence, and adapter tuning, which inserts lightweight trainable modules between existing layers of the model [12, 26].

In addition to PEFT, another promising strategy is Split Learning (SL), which partitions the model into two segments—one deployed on edge devices and the other on the server. Clients perform partial computations locally to generate intermediate representations (smashed data), which are then transmitted to the server for further processing. This approach reduces training costs and enhances privacy by keeping raw data on-device [27]. Although this emerging collaborative learning method can be regarded as an alternative to the FL paradigm, a promising research direction aims to combine the two strategies to unleash the respective advantages while mitigating their weaknesses [28].

Alternatively, model quantization reduces the size of LLMs by decreasing the number of bits used to represent model weights. Although there are different quantization methods, their common objective is to replace full-precision computations with low-precision alternatives. In contrast, model pruning reshapes the weights to compress LLMs. This approach can be categorized into structured model pruning, which decreases the number of layers in the model or attention heads in the case of Transformers,

and semi-structured model pruning, which zeros out specific weights. Model compression can also be achieved through knowledge distillation, which transfers the knowledge of a high-capacity teacher model to a smaller student model by training the student to match the teacher’s output distributions [12].

Model compression techniques can also be used to derive high-performing SLMs from larger counterparts. While the definitions of “small” and “large” significantly depend on both context and time, SLMs have attracted growing interest in the research community due to their ability to perform a wide range of language tasks effectively with limited computational resources. This makes them well-suited for deployment in resource-constrained environments such as mobile, on-device, and edge computing platforms [29]. SLMs typically have lightweight encoder-only or decoder-only architectures. Several encoder-only models are derived from BERT [30], with notable examples including MobileBERT [31], DistilBERT [32], and TinyBERT [33]. In contrast, lightweight decoder-only architectures include TinyLLaMA [34], Gemma [35], Phi-3-mini [36], MobileLLM [37], and SmoLLM¹.

The integration of LLMs into edge and federated environments has driven the development of specialized frameworks and aggregation techniques aimed at addressing the specific challenges of distributed learning, while leveraging the unique characteristics of these models. For example, FedDat addresses the challenge of data heterogeneity across clients by introducing a Dual-Adapter Teacher to regularize local updates and employing Mutual Knowledge Distillation for knowledge transfer [38]. To reduce the high computational and storage demands of deploying LLMs at the edge, the authors of [39] propose M²FEDSA, a framework that employs split learning to partition large-scale models and assign only privacy-sensitive components to client devices. Additionally, it transfers multimodal knowledge from the server to unimodal clients, aiming at further enhancing model performance. In contrast, FlexLoRA addresses resource and data heterogeneity by introducing an aggregation scheme that dynamically adjusts local LoRA ranks. These ranks are then aggregated on the server using Singular Value Decomposition, which also redistributes weights to ensure that all clients contribute effectively, regardless of resource capacity [40].

3. Token-based Federated Averaging

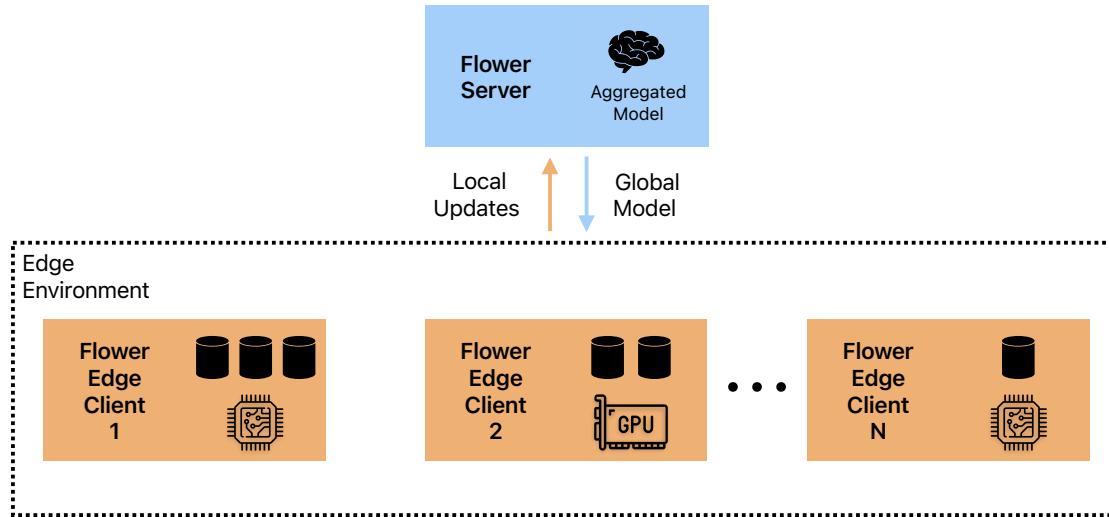


Figure 1: Typical Federated Learning architecture, where each client has different computation resources and datasets.

¹<https://huggingface.co/blog/smollm>

In a typical FL setting, as the one illustrated in Fig. 1, traditional aggregation methods like FedAvg, which weights client updates by the number of processed examples, may be insufficient for fine-tuning LLMs. This limitation arises because it fails to account for the variability in the information content of different training examples. In the context of LLMs, longer input sequences often contain more intricate linguistic patterns and richer semantic structures, which contribute more significantly to model learning. As a result, treating all examples equally, regardless of their token length, can lead to suboptimal updates and hinder overall model performance.

To address this, we propose a token-based weighting scheme that adjusts each client's contribution based on the total number of tokens processed during local training. This refinement ensures that updates from clients processing more linguistically dense and informative sequences are weighted more appropriately during aggregation. Such an approach is particularly valuable in federated settings where data is highly heterogeneous, and the lengths of input sequences vary widely. By aligning the aggregation process more closely with the actual learning signal, token-based weighting has the potential to improve fine-tuning accuracy and ensure more effective model convergence across diverse client datasets.

Therefore, when this FedAvg variation is used, the global model weights are computed as a weighted average of the local model updates, based on the processed number of tokens, from participating clients:

$$\mathbf{w}_{\tau+1} = \mathbf{w}_{\tau} - \eta \sum_{k=1}^K ((1 - \alpha) \frac{n_k}{N} + \alpha \frac{t_k}{T}) \mathbf{w}_t^{(k)} \quad (2)$$

where:

\mathbf{w}_{τ} is the updated global model weights after round τ ,

$\mathbf{w}_t^{(k)}$ is the model weights from client k after local training in round τ ,

t_k is the number of local data tokens used by client k ,

$T = \sum_{k=1}^K t_k$ is the total number of tokens across all K ,

n_k is the number of local data samples used by client k ,

$N = \sum_{k=1}^K n_k$ is the total number of samples across all K participating clients,

K is the total number of clients involved in the current communication round,

η is the learning rate,

α is a customizable parameter, which varies in a range between 0 and 1 and measures how the number of tokens and the number of samples influence the overall weights. Noteworthy, if $\alpha = 0$, the aggregation strategy is the classical FedAvg.

4. Experimental Evaluation

To evaluate the PEFT performance of the token-based FedAvg variation in our FL setting, we designed a set of experiments that simulate geographically distributed edge clients, each with access to a unique dataset to represent the local data distribution. Subsequently, we compared the results obtained with the token-based variation against both the classical FedAvg approach and a centralized PEFT, both used as baselines.

Each training step performs PEFT leveraging LoRA as a fine-tuning technique, which enables efficient adaptation of LLMs by injecting low-rank trainable matrices into frozen pre-trained models, significantly reducing the memory and computational overhead, enabling resource-constrained clients to join the federation.

Furthermore, to address the scalability and device heterogeneity that characterize the real world, where, for instance, some devices may be unavailable, we experimented using a semi-synchronous communication protocol between clients and the central server in our experiments, as also introduced in [41]. In detail, during the PEFT process, we define a minimum number of clients, smaller than the total client population, required to complete a training round and trigger the global weight aggregation. Clients unable to finish local training within the current round’s time window are not discarded. Instead, they transmit their updates in the subsequent round, avoiding wasted computation and preserving energy efficiency. This design choice eliminates the need for strict synchronization barriers, allowing edge devices with varying availability, computational capacity, and connectivity to participate effectively.

4.1. Datasets

The datasets used to fine-tune the models revolve around the topic of healthcare, but they differ in structure, content, and style, reflecting the heterogeneous nature of real-world medical data where silos often emerge due to institutional, regional, or organizational boundaries. Such differences serve not only to enhance the realism of our evaluation but also to improve experimental flexibility. The diverse nature of the datasets allows for systematic evaluation of model performance under varying conditions, such as differences in input length, linguistic complexity, and contextual depth, enabling a deeper understanding of model strengths and limitations.

PubMedQA [42] is the largest dataset included in this study, comprised of approximately 211,000 biomedical question-answer pairs derived from PubMed abstracts, providing a rich resource of information in the form of long answers. Complementing this, collections of medical flashcards from Medical Meadow [43] encapsulate concise, high-yield clinical facts and definitions, offering shorter sequences ideal for assessing models on shorter examples. Patient information from WikiDoc includes detailed narratives, differential diagnoses, and treatment pathways, introducing complex, longer-context clinical scenarios that test models’ abilities in a longer, more elaborate context. Finally, conversational data from iCliniq² reflects real-world doctor–patient dialogues, capturing colloquial language, patient concerns, and pragmatic clinical advice, thereby allowing evaluation of models’ performance interactive communication. By leveraging these datasets and varying both the length of the sequences and their number, our experiments enable robust assessment of model capabilities across multiple conditions, with different input lengths and context, showing how different data types and structures may influence the model’s understanding and generalization in a distributed federated environment.

4.2. Environment

As LLMs we used two versions of the SmolLM2 model, specifically the 135M and 360M ones, since their great tradeoff between model size and performance. We then compared multiple aggregation strategies, including traditional FedAvg and the token-based approach, to evaluate convergence behavior and final model performance. Our experiments also included a comparison with local fine-tuning, where we assume a single node has all the datasets available in a central location.

The experiments were conducted in a federated setting consisting of one central server and six client configurations (from A to F) distributed on 3 machines:

- Machine 1 (Configurations A and D): NVIDIA GeForce RTX 3060 GPU (1320 MHz base, up to 1777 MHz boost), Intel Core i7-9750H CPU (2.6 GHz base, up to 4.5 GHz turbo), 32GB RAM.
- Machine 2 and 3 (Configurations B, C, E, and F): Intel Core Ultra 7 155U CPU (1.7 GHz base, up to 4.8 GHz turbo), 32GB RAM.

Each client was assigned a distinct dataset for training. The number of examples processed per round was adjusted based on the computational capacity of each machine, ensuring a balanced workload distribution. Table 1 summarizes the specific configurations, including dataset names, example counts, and batch sizes, Table 2 describes how configurations are distributed on the machines.

²<https://huggingface.co/datasets/lavita/ChatDoctor-iCliniq>

The first experiment, involving client configurations A, B, and C, as presented in Table 1, was conducted by assigning a portion of the training set from each dataset to a separate machine, simulating a situation where each edge node has access to its private information.

In the second experiment, all training datasets were merged and then split into 'long' and 'short' examples: records with a length greater than 90 words (corresponding to the 90th percentile) were assigned to the 'long' dataset and used with configuration D. The remaining examples formed the 'short' dataset, used with configurations E and F. This is to better measure the performance variations when model weights are aggregated using standard FedAvg and token-based FedAvg, which also take care of the number of tokens used to fine-tune the model.

Finally, each experiment was benchmarked against classical centralized fine-tuning, in which LoRA was performed on the same data used in the federated setup, serving as the baseline.

Evaluation examples were extracted via seeded random sampling from data previously separated from the training sets, ensuring the test examples were not seen by the model during training. Specifically, the evaluation has been carried out by the central server using the global model on 512 examples, drawn from all the datasets. RougeL [44] and Bert Score [45] were used as evaluation metrics. RougeL measures the overlap between a generated text and the reference by computing the longest common subsequence (LCS) of tokens, which captures the highest degree of in-order matching without requiring contiguous matches; this makes RougeL sensitive to the preservation of key semantics and phrase structures. In contrast, BERTScore leverages contextualized embeddings from a pre-trained BERT model to compute token-level cosine similarities between candidate and reference sentences, thereby capturing not only exact matches but also paraphrases and synonyms [45]. Both BERTScore and RougeL metrics range from 0 to 1, making them interpretable as percentage values. When fine-tuning LLMs on data from a specific domain, such as the medical one, ROUGE-L provides a straightforward indication of how well the model reproduces specialized nomenclature and phraseology, while BERTScore offers a deeper semantic assessment, ensuring that conceptually equivalent but lexically varied outputs are properly credited.

4.3. Results

The results of the first experiment, conducted using the 135M parameter version of the SmolLM2 model, are presented in Fig. 2 and Table 3. This figure depicts the BERTScore and Rouge-L metrics of the global model across 10 rounds of federated PEFT training with varying values of α . Both BERTScore and Rouge-L metrics are tracked to provide a comprehensive view of model quality over time. Notably, the most significant performance gains occur during the initial round of training, after which the metrics stabilize. In particular, Table 3 (SmolLM2-135M, first experiment with heterogeneous clients) shows that the largest gains occur immediately after the first round (0→1), with RougeL rising from 0.172 to 0.210 (+0.038) and BERTScore from 0.519 to 0.575 (+0.056) across all α . After round 2–3, both metrics plateau with only minor fluctuations. This behavior is consistent with the findings reported in [46], where

Table 1

Client configurations used in the experimental setup, including datasets, number of training examples per round, batch sizes, and assigned machines.

Config	Dataset	#Examples	Batch size
A	pubmed_qa_211k	32,768	4
B	medical_meadow_medical_flashcards_34k	4,096	1
C	medical_meadow_wikidoc_patient_information_6k + medical_meadow_wikidoc_10k + chatdoctor_icliniq_7k	4,096	1
D	long examples	16,384	4
E	short examples (1)	8,192	1
F	short examples (2)	8,192	1

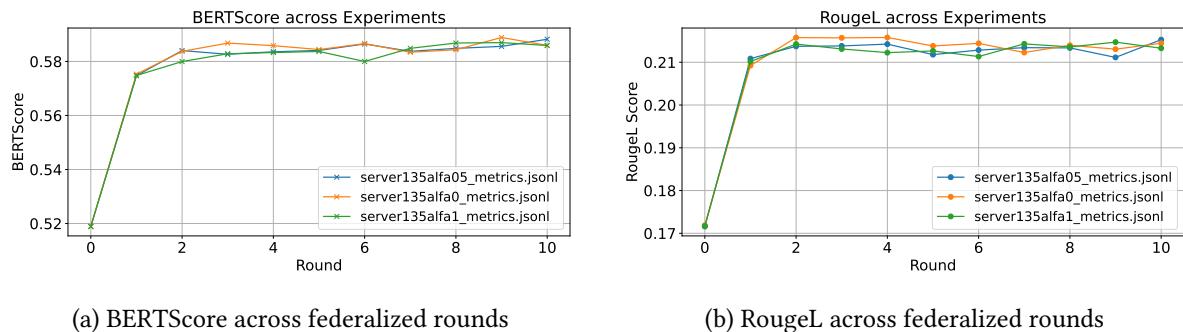
Experimental setups			
Experiment	Machine 1	Machine 2	Machine 3
1	Server, A	B	C
2	Server, D	E	F

Table 2

Machine-Configuration association for the experimental setups. Machine 1, being the most performing, is assigned the Server task to leverage the RTX 3060 GPU during inference.

early-stage adaptation captures the bulk of the performance improvement. Following this initial gain, in subsequent training rounds, performance remains stable and relatively consistent across different α settings.

Table 4 presents the corresponding results for the same experimental setup using the larger 360M version of the model. The overall trend mirrors that of the smaller model, reinforcing the consistency of the observed dynamics across model scales. Importantly, the table includes the results from a centralized PEFT configuration, which are consistently lower than those obtained in the federated setting. This contrast highlights the advantages of federated fine-tuning, particularly in leveraging diverse local data distributions while maintaining privacy and decentralization. These findings underscore the efficacy and scalability of federated PEFT, as they hold true across both small and larger model architectures. Finally, of particular note is that by round 10, the 135M model trained with $\alpha = 0.5$ exhibits slightly superior performance, suggesting a mild advantage for this setting over extended training durations. Although the performance gains are not substantial, they may point to a favorable balance between local update strength and global model coherence at this α value, making it a reasonable default in practical applications.



(a) BERTScore across federalized rounds

(b) RougeL across federalized rounds

Figure 2: Evaluation metrics across federated rounds for SmolLM2-135M in experimental setup 1, where each client is trained on examples drawn from a distinct dataset.

Table 3

First experiment: each client is trained on a different dataset with SmolLM2-135M fine-tuned. We report per-round RougeL and BERTScore for three aggregation settings $\alpha \in \{0, 0.5, 1\}$.

Round	SmolLM2-135M					
	Alpha = 0		Alpha = 0.5		Alpha = 1	
	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore
0	0.172	0.519	0.172	0.519	0.172	0.519
1	0.210	0.575	0.210	0.575	0.210	0.575
2	0.216	0.584	0.214	0.584	0.214	0.580
3	0.216	0.587	0.214	0.583	0.213	0.583
4	0.216	0.586	0.214	0.584	0.212	0.583
5	0.214	0.584	0.212	0.584	0.213	0.584
6	0.214	0.587	0.213	0.586	0.211	0.580
7	0.212	0.583	0.213	0.584	0.214	0.585
8	0.214	0.584	0.213	0.585	0.214	0.587
9	0.213	0.589	0.211	0.586	0.215	0.587
10	0.214	0.586	0.215	0.588	0.213	0.586

Table 4

First experiment: each client is trained on a different dataset with SmoLLM2-360M fine-tuned. We report RougeL and BERTScore per round for three aggregation settings $\alpha \in \{0, 0.5, 1\}$, plus a centralized PEFT configuration.

SmoLLM2-360M								
Round	Alpha = 0		Alpha = 0.5		Alpha = 1		Centralized	
	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore
0	0.176	0.533	0.176	0.533	0.176	0.533	0.176	0.533
1	0.214	0.590	0.214	0.589	0.214	0.588	0.204	0.579
2	0.216	0.595	0.214	0.594	0.215	0.592	0.206	0.586
3	0.218	0.598	0.216	0.595	0.217	0.593	0.205	0.587
4	0.218	0.597	0.150	0.596	0.217	0.595	0.205	0.588
5	0.218	0.599	0.218	0.598	0.220	0.597	0.207	0.590

From the results of the second experiment, which are presented in Tables 5 and 6, we can draw the same conclusion, as the federated settings proved to be valuable when compared with the centralized one, with minimal differences varying the α value. This is an encouraging outcome, as it implies that the system is relatively insensitive to the exact choice of α , allowing practitioners to adjust this parameter based on data availability, client heterogeneity, or system constraints without risking significant degradation in model quality. This flexibility enhances the practical applicability of the approach, especially in real-world federated environments where data distributions and client participation levels may vary considerably.

Table 5

Second experiment: each client is assigned a portion of the global dataset formed by aggregating all individual datasets. Examples are distributed across nodes based on the number of tokens. SmoLLM2-135M fine-tuned. We report per-round RougeL and BERTScore for aggregation settings $\alpha \in \{0, 0.25\}$.

SmoLLM2-135M					
Round	Alpha = 0		Alpha = 0.25		
	RougeL	BERTScore	RougeL	BERTScore	
0	0.173	0.521	0.173	0.521	
1	0.210	0.575	0.211	0.574	
2	0.214	0.584	0.213	0.593	
3	0.219	0.591	0.217	0.585	
4	0.215	0.591	0.216	0.591	
5	0.217	0.591	0.220	0.590	

Table 6

Second experiment: each client is assigned a portion of the global dataset formed by aggregating all individual datasets. Examples are distributed across nodes based on the number of tokens. SmoLLM2-360M fine-tuned. We report per-round RougeL and BERTScore for aggregation settings $\alpha \in \{0, 0.1, 0.25\}$, plus a centralized PEFT configuration.

SmoLLM2-360M								
Round	Alpha = 0		Alpha = 0.1		Alpha = 0.25		Centralized	
	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore	RougeL	BERTScore
0	0.180	0.533	0.180	0.533	0.180	0.533	0.180	0.533
1	0.221	0.594	0.218	0.596	0.216	0.593	0.207	0.578
2	0.221	0.599	0.217	0.596	0.217	0.595	0.211	0.587
3	0.223	0.601	0.216	0.596	0.217	0.597	0.210	0.588
4	0.224	0.603	0.219	0.601	0.219	0.599	0.210	0.590
5	0.225	0.604	0.219	0.601	0.221	0.600	0.213	0.590

5. Conclusions and Future Work

In this paper, we endeavoured to address the challenges inherent in the deployment and fine-tuning of LLMs on edge devices, with an aggregation strategy specifically designed for LLMs and a distributed

infrastructure replicating a real-world federated setting.

Our preliminary experiments, conducted using open-source healthcare datasets and two lightweight versions of the SmolLM2 model, demonstrate that a token-based approach achieves competitive—and in some cases superior—performance compared to both centralized training and conventional FL with FedAvg. These results validate the potential of our approach for enabling effective, privacy-preserving, and resource-efficient training of LLMs at the edge.

However, the observed performance improvement was marginal, suggesting that further experimentation is needed to validate these findings, for example, using datasets with a greater variety of example lengths. Future work will extend the experimentation to other datasets, from different use cases, to better highlight the differences between the token-based variation and the classical FedAvg.

Given that our evaluation relies on standard metrics, which are widely adopted in the literature, but are often criticized for having limitations in capturing the semantic quality of outputs, focusing more on surface-level differences than on true meaning. As a promising direction, we also plan to explore the use of LLMs as evaluators to provide a more nuanced, semantically-informed assessment of model outputs [47].

Finally, we plan to take into consideration the scalability aspects in terms of training times and communication overhead.

Acknowledgments

This work has been partially supported by the Spoke 1 “Future HPC & Big Data” of the Italian Research Center on High-Performance Computing, Big Data and Quantum Computing (ICSC) funded by MUR Missione 4 - Next Generation EU (NGEU).

Declaration on Generative AI

During the preparation of this work, the authors used ChatGPT and Grammarly for grammar and spelling checks, as well as for paraphrasing and rewording. All content generated with these tools was subsequently reviewed and edited by the authors, who take full responsibility for the final version of the publication.

References

- [1] S. Dahdal, L. Colombi, M. Brina, A. Gilli, M. Tortonesi, M. Vignoli, C. Stefanelli, An mlops framework for gan-based fault detection in bonfiglioli’s evo plant, *Infocommunications Journal* 16 (2024) 2–10. doi:10.36244/ICJ.2024.2.1.
- [2] L. Colombi, M. Vespa, N. Belletti, M. Brina, S. Dahdal, F. Tabanelli, F. Resca, E. Bellodi, M. Tortonesi, C. Stefanelli, et al., Embedding models for multivariate time series anomaly detection in industry 5.0: L. colombi et al., *Data Science and Engineering* (2025) 1–17.
- [3] L. Colombi, M. Brina, M. Vespa, F. Tabanelli, S. Dahdal, E. Bellodi, R. Venanzi, M. Tortonesi, M. Vignoli, C. Stefanelli, Optimizing industry 5.0 machine learning-based applications via synthetic data generation, in: 2024 IEEE 29th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD), IEEE, 2024, pp. 1–6.
- [4] Z. Alom, B. Carminati, E. Ferrari, A deep learning model for twitter spam detection, *Online Social Networks and Media* 18 (2020) 100079. URL: <https://www.sciencedirect.com/science/article/pii/S2468696420300203>. doi:<https://doi.org/10.1016/j.osnem.2020.100079>.
- [5] M. Bevilacqua, F. E. Ciarapica, C. Diamantini, D. Potena, Big data analytics methodologies applied at energy management in industrial sector: A case study, *International Journal of RF Technologies* 8 (2017) 105–122.
- [6] L. Colombi, A. Gilli, S. Dahdal, I. Boleac, M. Tortonesi, C. Stefanelli, M. Vignoli, A machine learning operations platform for streamlined model serving in industry 5.0, in: NOMS 2024-2024 IEEE

Network Operations and Management Symposium, 2024, pp. 1–6. doi:10.1109/NOMS59830.2024.10575103.

- [7] L. Colombi, I. Boleac, M. Brina, S. Dahdal, M. Tortonesi, M. Vignoli, C. Stefanelli, Multi-cluster mlops platform for industry 5.0, in: 2025 IEEE Symposium on Computers and Communications (ISCC), 2025.
- [8] M. Anisetti, C. A. Ardagna, N. Bena, E. Damiani, P. G. Panero, Continuous management of machine learning-based application behavior, *IEEE Transactions on Services Computing* 18 (2025) 112–125. doi:10.1109/TSC.2024.3486226.
- [9] N. Bena, M. Anisetti, G. Gianini, C. A. Ardagna, Certifying accuracy, privacy, and robustness of ml-based malware detection, *SN Computer Science* 5 (2024) 710.
- [10] A. D. Kotkar, R. S. Mahadik, P. G. More, S. A. Thorat, Comparative analysis of transformer-based large language models (llms) for text summarization, in: 2024 1st International Conference on Advanced Computing and Emerging Technologies (ACET), 2024, pp. 1–7. doi:10.1109/ACET61898.2024.10730348.
- [11] G. Hollósi, D. Ficzere, P. Varga, Generative ai for low-level netconf configuration in network management based on yang models, in: 2024 20th International Conference on Network and Service Management (CNSM), IEEE, 2024, pp. 1–7.
- [12] F. Piccialli, D. Chiaro, P. Qi, V. Bellandi, E. Damiani, Federated and edge learning for large language models, *Information Fusion* 117 (2025) 102840. URL: <https://www.sciencedirect.com/science/article/pii/S1566253524006183>. doi:<https://doi.org/10.1016/j.inffus.2024.102840>.
- [13] L. Colombi, E. D. Caro, S. Dahdal, F. Poltronieri, F. Tabanelli, M. Tortonesi, C. Stefanelli, M. Vignoli, Fededge-learn: a semi-supervised federated learning framework for industry 5.0, in: 2025 IEEE Symposium on Computers and Communications (ISCC), 2025.
- [14] B. McMahan, E. Moore, D. Ramage, S. Hampson, B. A. y. Arcas, Communication-Efficient Learning of Deep Networks from Decentralized Data, in: A. Singh, J. Zhu (Eds.), *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, volume 54 of *Proceedings of Machine Learning Research*, PMLR, 2017, pp. 1273–1282. URL: <https://proceedings.mlr.press/v54/mcmahan17a.html>.
- [15] I. Siragusa, S. Contino, M. La Ciura, R. Alicata, R. Pirrone, Medpix 2.0: a comprehensive multimodal biomedical dataset for advanced ai applications, *arXiv preprint arXiv:2407.02994* (2024).
- [16] L. Peng, G. Luo, sicheng zhou, jiandong chen, R. Zhang, Z. Xu, J. Sun, An in-depth evaluation of federated learning on biomedical natural language processing, 2023. URL: <https://arxiv.org/abs/2307.11254>.
- [17] X. Li, L. Peng, Y.-P. Wang, W. Zhang, Open challenges and opportunities in federated foundation models towards biomedical healthcare, *BioData Mining* 18 (2025) 2. URL: <https://doi.org/10.1186/s13040-024-00414-9>. doi:10.1186/s13040-024-00414-9.
- [18] S. R. Abbas, Z. Abbas, A. Zahir, S. W. Lee, Federated learning in smart healthcare: A comprehensive review on privacy, security, and predictive analytics with iot integration, *Healthcare* 12 (2024). URL: <https://www.mdpi.com/2227-9032/12/24/2587>. doi:10.3390/healthcare12242587.
- [19] L. Yuan, Z. Wang, L. Sun, P. S. Yu, C. G. Brinton, Decentralized federated learning: A survey and perspective, *IEEE Internet of Things Journal* 11 (2024) 34617–34638. doi:10.1109/JIOT.2024.3407584.
- [20] S. Sharma, S. Bhadula, Secure federated learning for intelligent industry 4.0 iot enabled self skin care application system, in: 2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC), 2023, pp. 1164–1170. doi:10.1109/ICAAIC56838.2023.10141028.
- [21] L. Li, Y. Fan, M. Tse, K.-Y. Lin, A review of applications in federated learning, *Computers & Industrial Engineering* 149 (2020) 106854. URL: <https://www.sciencedirect.com/science/article/pii/S0360835220305532>. doi:<https://doi.org/10.1016/j.cie.2020.106854>.
- [22] P. Qi, D. Chiaro, A. Guzzo, M. Ianni, G. Fortino, F. Piccialli, Model aggregation techniques in federated learning: A comprehensive survey, *Future Generation Computer Systems* 150 (2024) 272–293. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X23003333>. doi:<https://doi.org/10.1016/j.future.2023.09.008>.

[23] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, *ACM transactions on intelligent systems and technology* 15 (2024) 1–45.

[24] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, *Advances in neural information processing systems* 30 (2017).

[25] Y. Yao, J. Zhang, J. Wu, C. Huang, Y. Xia, T. Yu, R. Zhang, S. Kim, R. Rossi, A. Li, L. Yao, J. McAuley, Y. Chen, C. Joe-Wong, Federated large language models: Current progress and future directions, 2024. URL: <https://arxiv.org/abs/2409.15723>.

[26] Z. Lin, G. Qu, Q. Chen, X. Chen, Z. Chen, K. Huang, Pushing large language models to the 6g edge: Vision, challenges, and opportunities, 2024. URL: <https://arxiv.org/abs/2309.16739>.

[27] N. Zhong, Y. Wang, R. Xiong, Y. Zheng, Y. Li, M. Ouyang, D. Shen, X. Zhu, Casit: Collective intelligent agent system for internet of things, *IEEE Internet of Things Journal* 11 (2024) 19646–19656. doi:10.1109/JIOT.2024.3366906.

[28] Q. Duan, S. Hu, R. Deng, Z. Lu, Combined federated and split learning in edge computing for ubiquitous intelligence in internet of things: State-of-the-art and future directions, *Sensors* 22 (2022). URL: <https://www.mdpi.com/1424-8220/22/16/5983>. doi:10.3390/s22165983.

[29] C. V. Nguyen, X. Shen, R. Aponte, Y. Xia, S. Basu, Z. Hu, J. Chen, M. Parmar, S. Kunapuli, J. Barrow, J. Wu, A. Singh, Y. Wang, J. Gu, F. Dernoncourt, N. K. Ahmed, N. Lipka, R. Zhang, X. Chen, T. Yu, S. Kim, H. Deilamsalehy, N. Park, M. Rimer, Z. Zhang, H. Yang, R. A. Rossi, T. H. Nguyen, A survey of small language models, 2024. URL: <https://arxiv.org/abs/2410.20011>.

[30] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: <https://aclanthology.org/N19-1423/>. doi:10.18653/v1/N19-1423.

[31] Z. Sun, H. Yu, X. Song, R. Liu, Y. Yang, D. Zhou, MobileBERT: a compact task-agnostic BERT for resource-limited devices, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online*, 2020, pp. 2158–2170. URL: <https://aclanthology.org/2020.acl-main.195/>. doi:10.18653/v1/2020.acl-main.195.

[32] V. Sanh, L. Debut, J. Chaumond, T. Wolf, Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, 2020. URL: <https://arxiv.org/abs/1910.01108>.

[33] X. Jiao, Y. Yin, L. Shang, X. Jiang, X. Chen, L. Li, F. Wang, Q. Liu, TinyBERT: Distilling BERT for natural language understanding, in: T. Cohn, Y. He, Y. Liu (Eds.), *Findings of the Association for Computational Linguistics: EMNLP 2020*, Association for Computational Linguistics, Online, 2020, pp. 4163–4174. URL: <https://aclanthology.org/2020.findings-emnlp.372/>. doi:10.18653/v1/2020.findings-emnlp.372.

[34] P. Zhang, G. Zeng, T. Wang, W. Lu, Tinyllama: An open-source small language model, 2024. URL: <https://arxiv.org/abs/2401.02385>.

[35] G. Team, T. Mesnard, C. Hardin, R. Dadashi, S. Bhupatiraju, S. Pathak, L. Sifre, M. Rivière, M. S. Kale, J. Love, P. Tafti, L. Hussenot, P. G. Sessa, A. Chowdhery, A. Roberts, A. Barua, A. Botev, A. Castro-Ros, A. Slone, A. Héliou, A. Tacchetti, A. Bulanova, A. Paterson, B. Tsai, B. Shahriari, C. L. Lan, C. A. Choquette-Choo, C. Crepy, D. Cer, D. Ippolito, D. Reid, E. Buchatskaya, E. Ni, E. Noland, G. Yan, G. Tucker, G.-C. Muraru, G. Rozhdestvenskiy, H. Michalewski, I. Tenney, I. Grishchenko, J. Austin, J. Keeling, J. Labanowski, J.-B. Lespiau, J. Stanway, J. Brennan, J. Chen, J. Ferret, J. Chiu, J. Mao-Jones, K. Lee, K. Yu, K. Millican, L. L. Sjoesund, L. Lee, L. Dixon, M. Reid, M. Mikuła, M. Wirth, M. Sharman, N. Chinaev, N. Thain, O. Bachem, O. Chang, O. Wahltinez, P. Bailey, P. Michel, P. Yotov, R. Chaabouni, R. Comanescu, R. Jana, R. Anil, R. McIlroy, R. Liu, R. Mullins, S. L. Smith, S. Borgeaud, S. Girgin, S. Douglas, S. Pandya, S. Shakeri, S. De, T. Klimenko, T. Hennigan, V. Feinberg, W. Stokowiec, Y. hui Chen, Z. Ahmed, Z. Gong, T. Warkentin, L. Peran, M. Giang, C. Farabet, O. Vinyals, J. Dean, K. Kavukcuoglu, D. Hassabis, Z. Ghahramani, D. Eck, J. Barral,

F. Pereira, E. Collins, A. Joulin, N. Fiedel, E. Senter, A. Andreev, K. Kenealy, Gemma: Open models based on gemini research and technology, 2024. URL: <https://arxiv.org/abs/2403.08295>.

[36] M. Abdin, J. Aneja, H. Awadalla, A. Awadallah, A. A. Awan, N. Bach, A. Bahree, A. Bakhtiari, J. Bao, H. Behl, A. Benhaim, M. Bilenko, J. Bjorck, S. Bubeck, M. Cai, Q. Cai, V. Chaudhary, D. Chen, D. Chen, W. Chen, Y.-C. Chen, Y.-L. Chen, H. Cheng, P. Chopra, X. Dai, M. Dixon, R. Eldan, V. Fragoso, J. Gao, M. Gao, M. Gao, A. Garg, A. D. Giorno, A. Goswami, S. Gunasekar, E. Haider, J. Hao, R. J. Hewett, W. Hu, J. Huynh, D. Iter, S. A. Jacobs, M. Javaheripi, X. Jin, N. Karampatziakis, P. Kauffmann, M. Khademi, D. Kim, Y. J. Kim, L. Kurilenko, J. R. Lee, Y. T. Lee, Y. Li, Y. Li, C. Liang, L. Liden, X. Lin, Z. Lin, C. Liu, L. Liu, M. Liu, W. Liu, X. Liu, C. Luo, P. Madan, A. Mahmoudzadeh, D. Majercak, M. Mazzola, C. C. T. Mendes, A. Mitra, H. Modi, A. Nguyen, B. Norick, B. Patra, D. Perez-Becker, T. Portet, R. Pryzant, H. Qin, M. Radmilac, L. Ren, G. de Rosa, C. Rosset, S. Roy, O. Ruwase, O. Saarikivi, A. Saied, A. Salim, M. Santacroce, S. Shah, N. Shang, H. Sharma, Y. Shen, S. Shukla, X. Song, M. Tanaka, A. Tupini, P. Vaddamanu, C. Wang, G. Wang, L. Wang, S. Wang, X. Wang, Y. Wang, R. Ward, W. Wen, P. Witte, H. Wu, X. Wu, M. Wyatt, B. Xiao, C. Xu, J. Xu, W. Xu, J. Xue, S. Yadav, F. Yang, J. Yang, Y. Yang, Z. Yang, D. Yu, L. Yuan, C. Zhang, C. Zhang, J. Zhang, L. L. Zhang, Y. Zhang, Y. Zhang, Y. Zhang, X. Zhou, Phi-3 technical report: A highly capable language model locally on your phone, 2024. URL: <https://arxiv.org/abs/2404.14219>.

[37] Z. Liu, C. Zhao, F. Iandola, C. Lai, Y. Tian, I. Fedorov, Y. Xiong, E. Chang, Y. Shi, R. Krishnamoorthi, L. Lai, V. Chandra, Mobilellm: Optimizing sub-billion parameter language models for on-device use cases, 2024. URL: <https://arxiv.org/abs/2402.14905>.

[38] H. Chen, Y. Zhang, D. Krompass, J. Gu, V. Tresp, Feddat: An approach for foundation model finetuning in multi-modal heterogeneous federated learning, Proceedings of the AAAI Conference on Artificial Intelligence 38 (2024) 11285–11293. URL: <https://ojs.aaai.org/index.php/AAAI/article/view/29007>. doi:10.1609/aaai.v38i10.29007.

[39] Z. Zhang, F. Qi, C. Xu, Enhancing storage and computational efficiency in federated multimodal learning for large-scale models, in: R. Salakhutdinov, Z. Kolter, K. Heller, A. Weller, N. Oliver, J. Scarlett, F. Berkenkamp (Eds.), Proceedings of the 41st International Conference on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, PMLR, 2024, pp. 59685–59699. URL: <https://proceedings.mlr.press/v235/zhang24az.html>.

[40] J. Bai, D. Chen, B. Qian, L. Yao, Y. Li, Federated fine-tuning of large language models under heterogeneous tasks and client resources, in: A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, C. Zhang (Eds.), Advances in Neural Information Processing Systems, volume 37, Curran Associates, Inc., 2024, pp. 14457–14483. URL: https://proceedings.neurips.cc/paper_files/paper/2024/file/1a134b50202088aa8c595cc99b310e5a-Paper-Conference.pdf.

[41] J. Nguyen, K. Malik, H. Zhan, A. Yousefpour, M. Rabbat, M. Malek, D. Huba, Federated learning with buffered asynchronous aggregation, in: International conference on artificial intelligence and statistics, PMLR, 2022, pp. 3581–3607.

[42] Q. Jin, B. Dhingra, Z. Liu, W. Cohen, X. Lu, Pubmedqa: A dataset for biomedical research question answering, in: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), 2019, pp. 2567–2577.

[43] T. Han, L. C. Adams, J.-M. Papaioannou, P. Grundmann, T. Oberhauser, A. Figueira, A. Löser, D. Truhn, K. K. Bressem, Medalpaca – an open-source collection of medical conversational ai models and training data, 2025. URL: <https://arxiv.org/abs/2304.08247>.

[44] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: <https://aclanthology.org/W04-1013/>.

[45] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, Bertscore: Evaluating text generation with bert, in: International Conference on Learning Representations, 2020. URL: <https://openreview.net/forum?id=SkeHuCVFDr>.

[46] Z. Wang, B. Tian, Y. He, Z. Shen, L. Liu, A. Li, One communication round is all it needs for federated fine-tuning foundation models, arXiv preprint arXiv:2412.04650 (2024).

[47] Z. Li, X. Xu, T. Shen, C. Xu, J.-C. Gu, Y. Lai, C. Tao, S. Ma, Leveraging large language models for NLG evaluation: Advances and challenges, in: Y. Al-Onaizan, M. Bansal, Y.-N. Chen (Eds.), Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Miami, Florida, USA, 2024, pp. 16028–16045. URL: <https://aclanthology.org/2024.emnlp-main.896/>. doi:10.18653/v1/2024.emnlp-main.896.