Adapter-based Selective Knowledge Distillation for Federated Multi-domain Meeting Summarization

Xiachong Feng*, Xiaocheng Feng*, Xiyuan Du, Min-Yen Kan, Bing Qin†

***Abstract*—Meeting summarization has emerged as a promising technique for providing users with condensed summaries. However, existing work has focused on training models on centralized data, neglecting real-world scenarios where meeting data are infeasible to collect centrally, due to their sensitive nature. This gap motivates us to explore federated learning for meeting summarization. Two critical challenges impede progress. First, state-of-the-art summarizers are based on parameter-heavy pre-trained models. Exchanging such a model's parameters across clients imposes large bandwidth costs. Second, as real-world meeting data belong to various domains and are distributed across clients, they are instances of non-identically and independently distributed (non-IID). IID assumptions do not hold, which changes which forms of learning algorithms best apply. To address this, we propose *Adapter-based Federated Selective Knowledge Distillation* (ADAFEDSELECKD) for training performant client models. Specifically, we develop an adapter-based summarization model where two adapters cooperatively facilitate learning using fewer parameters to reduce communication costs. Then, we devise a selective knowledge distillation strategy, assisting clients in robustly handling domain-focused modelling on their own data, while leveraging global parameters based on non-IID data. Extensive experiments on the QMSum benchmark demonstrate ADAFEDSELECKD can achieve comparable performance with powerful centralized training methods, and shows its generalizability and robustness.**

***Index Terms*—Meeting Summarization, Federated Learning, Knowledge Distillation, Parameter-efficient Fine-tuning.**

I. INTRODUCTION

MEETING summarization aims to produce concise meeting summaries given lengthy meeting transcripts, efficiently facilitating readers to grasp essential meeting information [1]. With the advancement of meeting technologies, many meetings are now also recorded regularly and automatically transcribed with AI tools, facilitating offline meeting reviews. Meeting summarization can leverage these inputs, further building capabilities to mitigate meeting overload.

Meeting summarization has attracted extensive research attention as of late [2], [3]. Existing endeavours focus on developing performant summarization models, utilizing data resources located in a single location (also known as *centralized meeting summarization*). [4]–[10].

* Equal contributions. † Corresponding author.

Xiachong Feng is with the University of Hong Kong, 999077, Hong Kong (e-mail: fengxc@hku.hk)

Xiaocheng Feng and Bing Qin are with the Peng Cheng Laboratory, 518000, Shenzhen, Guangdong, China (e-mail: xcfeng,qinb@ir.hit.edu.cn)

Xiaocheng Feng, Xiyuan Du and Bing Qin are with the Harbin Institute of Technology, 150001, Harbin, Heilongjiang, China (e-mail: xcfeng,xydu,qinb@ir.hit.edu.cn)

Min-Yen Kan is with the School of Computing, National University of Singapore, 117417, Singapore (e-mail: knmnyn@nus.edu.sg)

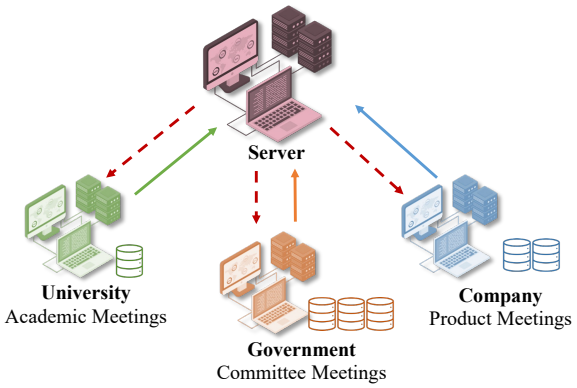


Fig. 1. The overall federated learning framework of multi-domain meeting summarization. In the concrete setting for this paper, there is one central server and three clients covering distinct domains: *Academic*, *Committee* and *Product*. Each client uniquely maintains its own domain-specific data.

While meaningful in theory, in practice real-world meeting summarization has additional privacy challenges that substantially change the problem framing. Concretely speaking, real-world meetings inextricably contain highly private and sensitive information; e.g., confidential company contents and personal information that are private [11]. When extended to multi-modal data, video and audio meeting recordings often also meet with facial representation and voiceprint issues since both are likewise highly sensitive [12]. For these reasons, meeting data is highly sensitive and unable to be shared for model training purposes and is typically siloed. This makes the collection of meeting data in a central location infeasible.

As such, despite the encouraging research achievements reported in the current literature, we find such solutions do not meet the requirements of real-world scenarios. They neglect the investigation towards developing solutions where meeting data are necessarily siloed and are distributed across different client sites.

To close the above gap, we take the first step to study the meeting summarization task by leveraging a *federated learning framework*, a widely-adopted approach for decentralized machine learning. It enables model training across multiple distributed clients [13], [14]. Figure 1 depicts the entire learning framework that aims to effectively train performant client-side summarization models by deriving global knowledge from other clients, without needing to access their private data. However, there are two critical challenges that need to be carefully addressed in order to learn high-performance summarization models under federated learning. First, current state-of-the-art meeting summarization models are based on

pre-trained language models that maintain a very large number of parameters. Updating all model parameters represents an infeasible communication cost. Instead, *limited scale client–server communication* is more realistic. This restricts the exchange of parameter updates between the server and its clients to a budget. Second, meetings distributed across multiple clients often belong to different domains. Figure 1 illustrate this scenario, in which there exists three meeting domains: academic, committee, and product, respectively. A single, central model would not serve to support the distinct needs of the different domains. This challenging *non-identically and independently distributed (non-IID)* data learning setting often causes the client model to deviate from its own domain as it learns global knowledge based on non-IID data.

To mitigate the above two challenges, we propose a unified method, dubbed Adapter-based Federated Selective Knowledge Distillation (ADAFEDSELECTKD). To address the first challenge, we draw support from parameter-efficient fine-tuning techniques and design an adapter-based summarization model to reduce communication costs. Specifically, we introduce a few lightweight trainable adapters [15] to pre-trained language models [16], [17] while keeping the pre-trained language models frozen. We meticulously design two types of adapters — global adapter and local adapter — tailored for the federated learning framework to facilitate information exchange between the server and clients. In particular, the global adapter is responsible for providing global knowledge while local adapters are optimized towards the local meeting summarization task. To address the second challenge, we devise a federated selective knowledge distillation strategy to not only effectively derive global knowledge for the client summarization model, but also train the model to favour its own local domain performance. Specifically, the client model adopts knowledge distillation [18] as the optimization algorithm to both learn from its local data and distill global knowledge from the global adapter. Moreover, we propose an entropy-based selective strategy based on the assumption that the higher the entropy of global knowledge, the more uncertain the knowledge. This adaptively distills knowledge from the global adapter.

We conduct experiments on the QMSum benchmark [6], which comprises meeting summarization data across three distinct domains: academic, committee and product. The automatic evaluation results based on three model variants across three clients consistently demonstrate the efficacy of our proposed method. Our results achieve comparable results to centralized training methods. Moreover, human evaluation results validate the substantial improvements attained by our method over baseline approaches. We further conduct downstream analyses of our model's various settings that allow us to conclude that our method is both generalizable and robust.

II. PRELIMINARIES

We first introduce the multi-domain meeting summarization dataset and the task definition, then provide an overview of federated learning. We define all of the mathematical notation employed in this work in Table I.

TABLE I
MATHEMATICAL NOTATIONS UTILIZED IN THIS PAPER.

Notation	Description
Dataset-related	
$\mathbb{D}$	meeting summarization dataset
$\mathbb{X}$	a set of input documents
$\mathbb{Y}$	a set of meeting summaries
$\mathcal{X}$	one input document $\mathcal{X} \in \mathbb{X}$
$\mathcal{Y}$	one meeting summary $\mathcal{Y} \in \mathbb{Y}$
V	vocabulary
Framework-related	
$\mathcal{S}$	server
$\mathcal{C}$	client
i	index of the client
$\mathbb{C}$	a set of clients, $\mathcal{C}_i \in \mathbb{C}$
$\mathcal{A}_c$	client-side optimization algorithm
$\mathcal{A}_s$	server-side aggregation algorithm
Model-related	
$\mathcal{M}$	client-side task model (meeting summarizer)
$\boldsymbol{W}$	learnable parameters of the model $\mathcal{M}$
$\boldsymbol{X}$	output representation of the encoder layer
$\boldsymbol{Y}$	output representation of the decoder layer
$\hat{\boldsymbol{Y}}$	output representation of the adapter
l	index of the transformer layer
L	the number of transformer layers
n	the dimension of the model $\mathcal{M}$
m	adapter bottleneck dimension
q	normalized output distribution (after `softmax`)
Learning-related	
$\mathcal{L}$	loss function
λ	weight for the knowledge distillation loss $\mathcal{L}_{KD}$
r	index of the federated learning round
$H(q)$	entropy of the distribution q
τ	threshold for the entropy

A. Multi-domain Meeting Summarization Dataset

In this paper, we leverage the QMSum dataset [6] to conduct experiments under the federated learning setting. QMSum consists of query–summary pairs over 232 meeting transcripts from three distinct domains: namely academic, committee and product meetings. This dataset is thus well-suited for the multi-domain meeting summarization task. Under our federated scenario, we posit that three clients hold meetings from each of the three distinct domains, respectively. Notably, QMSum is a query-based meeting summarization dataset, in which each instance is composed of a specific query, the relevant meeting transcripts and the corresponding summary. Following Lee and Sogaard [8], we concatenate the query and the meeting transcripts to construct the input document $\mathcal{X}$, resulting in a parallel corpus $\mathbb{D} : (\mathcal{X}, \mathcal{Y}) \in (\mathbb{X}, \mathbb{Y})$, where $\mathcal{Y}$ is the corresponding summary with respect to $\mathcal{X}$. We give detailed statistics for the QMSum dataset in Table II.

B. Task Definition

Given the document $\mathcal{X}$, the client-side meeting summarization model aims to produce a concise meeting summary $\mathcal{Y}$, where $\mathcal{X}$ is the concatenation of one query's words and relevant meeting transcripts $[\underbrace{x_1, x_2, ..., x_i}_{\texttt{query}} \texttt{\#SEP\#} \underbrace{x_{i+1}, x_{i+2}, ..., x_{|\mathcal{X}|}}_{\texttt{transcripts}}]$, `#SEP#` denotes

TABLE II
STATISTICS FOR THE QMSUM DATASET ENCOMPASS THREE DOMAINS: ACADEMIC, COMMITTEE AND PRODUCT. "#" INDICATES THE QUANTITY OF DOCUMENT–SUMMARY PAIRS. THE AVERAGE NUMBER OF TURNS DURING MEETINGS IS DENOTED BY "AVG. TURNS". "AVG. SPEAKERS" REPRESENTS THE AVERAGE NUMBER OF SPEAKERS PARTICIPATING IN THE MEETINGS. "AVG. TOKENS" REFERS TO THE AVERAGE NUMBER OF WORDS SPOKEN DURING THE MEETINGS, AND "AVG. SUM" INDICATES THE AVERAGE NUMBER OF WORDS IN THE SUMMARIES.

	Academic			**Committee**			**Product**		
	Train	**Valid**	**Test**	**Train**	**Valid**	**Test**	**Train**	**Valid**	**Test**
#	218	45	49	284	67	66	593	125	129
Avg. Turns	54.64	59.33	46.45	9.51	9.13	10.85	68.61	79.30	77.86
Avg. Speakers	4.35	4.09	4.22	3.26	3.10	4.14	3.76	3.90	3.93
Avg. Tokens	1049.8	1156.53	912.00	667.14	607.9	713.26	898.04	903.53	933.67
Avg. Sum	46.22	50.67	43.82	77.95	73.67	69.94	65.09	65.38	57.38

one specific token between the query and transcripts. Note that speaker roles, such as "marketing" and "project manager", are treated as ordinary tokens and included in transcripts. $\mathcal{Y}$ consists of $|\mathcal{Y}|$ words $[y_1, y_2, ..., y_{|\mathcal{Y}|}]$. A brief example is shown as follows:

- **Query** $\mathcal{X}_{[1:i]}$: Summarize the discussion about the trends of current remote controls.
- **Meeting Transcripts** $\mathcal{X}_{[i+1:|\mathcal{X}|]}$: Marketing: This is just a presentation on the trends that we're gonna use to make the product stand out from ..... Project Manager: What do you think of adding an LCD? .....
- **Corresponding Summary** $\mathcal{Y}$: The group discussed different trends based on different ages of people, ..., finally they decide to add an LCD screen.

C. Federated Learning Framework

We investigate the multi-domain meeting summarization task under the federated learning framework. Federated learning adopts a client–server paradigm and enables the collaborative training of models across multiple decentralized data sources without harvesting the sensitive raw data [13].

Roughly speaking, the federated learning methodology progresses in a synchronous, iterative fashion. During each learning round, every client locally optimizes its own ***client-side model*** based on its private data via the ***client-side optimization algorithm*** and then transmits the updated parameters to the central server. Subsequently, the server gathers the updates from clients and aggregates them into new server parameters by means of the ***server-side aggregation algorithm***. Finally, the new global parameters are broadcast from the server to all clients for the next round. It should be noted that three crucial components constitute this learning process:

- ***Client-side model*** $\mathcal{M}$ housing the client-specific model parameters $\boldsymbol{W}$. It is in charge of generating meeting summaries.
- ***Client-side optimization algorithm*** $\mathcal{A}_c$ endeavors to optimize the client-side model $\mathcal{M}$ based on the local private data $\mathbb{D}$.
- ***Server-side aggregation algorithm*** $\mathcal{A}_s$ is responsible for aggregating parameters furnished by clients.

Formally speaking, there exists a set $\mathbb{C}$ of $|\mathbb{C}|$ clients, where each individual client is denoted $\mathcal{C} \in \mathbb{C}$. As illustrated in our concrete scenario in Figure 1, there are three ($|\mathbb{C}| = 3$) clients in our set; namely, the academic client, the committee client, and the product client. Each client $\mathcal{C}_i$ possesses its own private domain-specific corpus of meeting summaries $\mathbb{D}_i$, as well as a client-specific meeting summarization model $\mathcal{M}_i$. The learnable parameters $\boldsymbol{W}_i$ of model $\mathcal{M}_i$ are optimized using the client-side optimization algorithm $\mathcal{A}_c$ based on the local dataset $\mathbb{D}_i$ in the r_{th} round of optimization.

$$\boldsymbol{W}_i^{r+1} \leftarrow \mathcal{A}_c(\mathcal{M}_i(\boldsymbol{W}_i^r), \mathbb{D}_i). \tag{1}$$

Subsequently, the central server $\mathcal{S}$ aggregates all updated parameters $\boldsymbol{W}_i^{r+1}$ from the clients and adopts a server-side aggregation algorithm ($\mathcal{A}_s$) to consolidate the information. In particular, we utilize the Federated Averaging (FedAvg) algorithm [19] as our aggregation algorithm $\mathcal{A}_s$.

$$\boldsymbol{W}^{r+1} \leftarrow \sum_{i=1}^{|\mathbb{C}|} \frac{|\mathbb{D}_i|}{|\mathbb{D}|} \boldsymbol{W}_i^{r+1}, \text{ where } |\mathbb{D}| = \sum_{i=1}^{|\mathbb{C}|} |\mathbb{D}_i|, \tag{2}$$

where $|\mathbb{C}|$ denotes the number of clients, $|\mathbb{D}_i|$ represents the number of instances in the local dataset $\mathbb{D}_i$ and $|\mathbb{D}|$ gives the total number of instances among all clients.

Afterwards, the newly-gathered server-side parameters $\boldsymbol{W}^{r+1}$ are distributed to all clients $\mathcal{C}_i$ to offer enriched global knowledge. In the forthcoming methodology section (§III), we demonstrate our contribution towards a more **communication-efficient client-side model** $\mathcal{M}_i$ and **robust client-side optimization algorithm** $\mathcal{A}_c$.

III. METHODOLOGY

We invent an integrated method, adapter-based federated selective knowledge distillation (ADAFEDSELECTKD), to achieve the efficient and robust federated multi-domain meeting summarization task. It comprises (1) A client-side model $\mathcal{M}$, which is an adapter-based meeting summarizer, and (2) A client-side optimization algorithm $\mathcal{A}_c$: which is a selective knowledge distillation algorithm. The overall ADAFEDSELECTKD learning procedure is illustrated in Figure 2.

A. Overview

Our proposed ADAFEDSELECTKD method significantly enhances two dimensions of the overall learning process.

- ***At the model level***, we introduce a client-side model called the adapter-based meeting summarizer ($\mathcal{M}$), which employs a frozen pre-trained language model as its backbone model (refer to §III-B2) and integrates few learnable

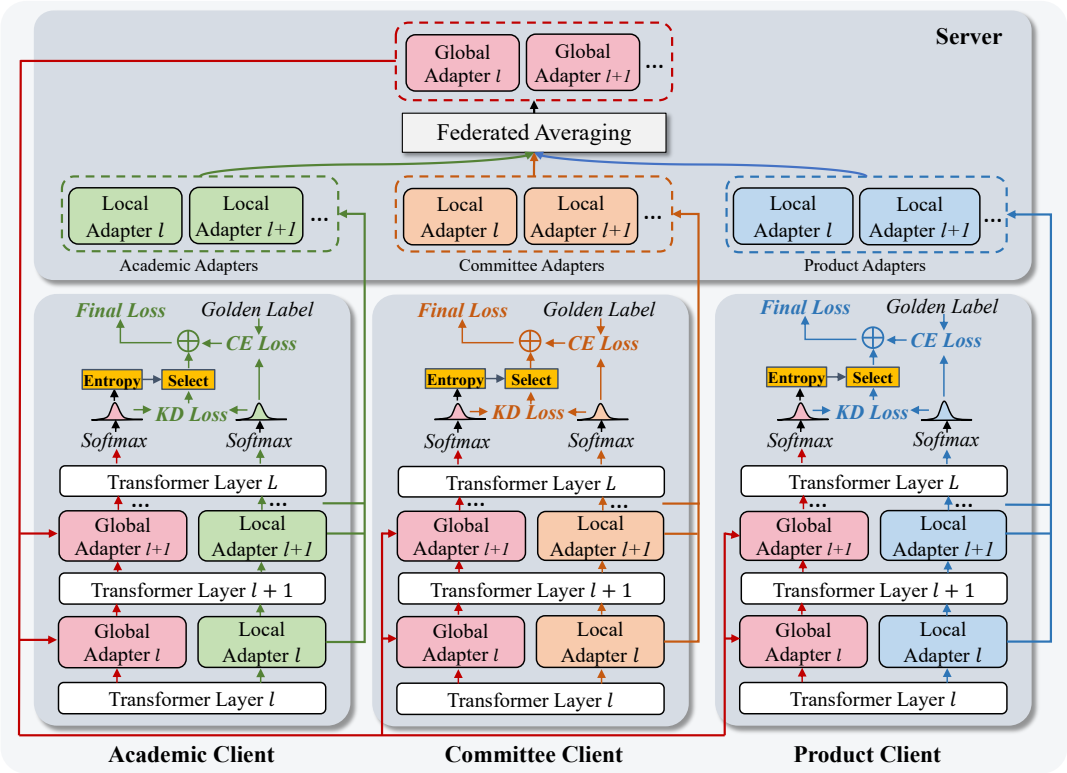


Fig. 2. Illustration of our proposed AdaFedSelectKD learning framework. The overall framework adheres to a client–server learning paradigm. At the bottom, three clients are depicted, where each client adopts the selective knowledge distillation algorithm to optimize its own adapter-based meeting summarizer using its domain-specific private data. Two types of adapters are tailored for the information exchange between the server and clients, including the global adapter and the local adapter. The optimized parameters from three clients are then conveyed to the central server. At the top, the central server employs the federated averaging algorithm to aggregate client information. The resulting new parameters are distributed to the clients for the subsequent learning round.

lightweight adapters to facilitate communication-efficient learning (refer to §III-B3). Specifically, we develop two kinds of adapters tailored for the federated learning setting, namely the global adapter and the local adapter. The global adapter functions as an intermediary for exchanging information between the server and clients, while the local adapter not only distils global knowledge from the global adapter but also is optimized towards the local domain.

- ***At the algorithm level***, we devise a client-side optimization algorithm termed the selective knowledge distillation strategy ($\mathcal{A}_c$), which adaptively and robustly optimizes local learnable adapters. Concretely, the knowledge distillation method permits deriving global knowledge from the server while ensuring the summarizer is prone to the local domain (refer to §III-C2). Furthermore, our meticulously designed selective strategy draws support from entropy as a measure of uncertainty and shows great promise in transferring credible global knowledge to clients (refer to §III-C3).

B. Adapter-based Meeting Summarizer

In this section, we elucidate our motivation for incorporating adapters and subsequently delineate the precise model architecture for both the backbone model as well as two varieties of adapters, namely the global adapter and the local adapter.

1) Motivation: In this section, we elaborate on the motivation underlying the design of our adapter-based meeting summarization system by addressing the following two questions:

- ***Why do we employ the adapter?*** In recent years, pre-trained language models have dominated the natural language processing field and have achieved remarkable success. Therefore, it is ideal to leverage such models as potent meeting summarization systems. However, there are two key challenges. Firstly, exchanging the parameters of these pre-trained language models incurs *high client–server communication costs* due to a large number of model parameters. Secondly, the *lack of sufficient hardware capabilities* in real-world scenarios means some clients may struggle to handle such compute-intensive tasks. On this account, we apply the parameter-efficient fine-tuning strategy to the pre-trained language model by fine-tuning only a few lightweight adapters, thereby addressing the above challenges.
- ***Why do we design two types of adapters?*** The most primitive and widely adopted federated learning algorithm is the Federated Averaging, which directly broadcasts newly aggregated global parameters to clients to initialize their models for the next round of training [19]. However, such a method notoriously performs poorly when clients hold non-IID data since aggregating divergent model parameters leads to model distraction [20], thereby leading

to the client model cannot focus on its own domain. To combat this issue, we design two types of adapters. One is the *global adapter* that receives server parameters and provides global knowledge via output distribution. The other is the *local adapter* that is optimized towards the local summarization task. By bifurcating the parameters in this fashion, we overcome the difficulties that arise from aggregating disparate model parameters across clients with non-IID data.

2) Backbone Model: We employ two types of pre-trained language models, one is BART [16] and the other is LED [17], as the backbone model. Both of them adopt the Transformer architecture [21] and have been pre-trained on a huge volume of data. They inherit a sequence-to-sequence framework, whereby the encoder first encodes the source sequence into distributed representations, which are then decoded by the decoder to generate the target summary.

Formally speaking, the input to the encoder is $\boldsymbol{X}^0$, which denotes the sum of the word embeddings $\boldsymbol{X}_{\text{emb}}$ and position embeddings $\boldsymbol{X}_{\text{pos}}$ of the input document $\mathcal{X}$. $\overset{L}{\underset{l=1}{:=}}$ symbolizes L identical layers and $\boldsymbol{X}^{l-1}$ signifies the output representation of the $l-1_{th}$ encoder layer. Besides, $\text{FFN}(\cdot)$ represents a position-wise feed-forward network, and $\text{SELF-ATT}(\cdot)$ denotes a multi-head self-attention.

$$\boldsymbol{X}^L = \text{ENCODER}(\boldsymbol{X}^0) \overset{L}{\underset{l=1}{:=}} \text{FFN}\left(\text{SELF-ATT}(\boldsymbol{X}^{l-1})\right) \quad (3)$$

The decoder takes the output $\boldsymbol{X}^L$ of the encoder and the shifted right representation $\boldsymbol{Y}^0$ of $\mathcal{Y}$ as the input to produce the final representation $\boldsymbol{Y}^L$, which will be projected into the vocabulary space in order to predict the summary.

$$\begin{aligned}\boldsymbol{Y}^L &= \text{DECODER}(\boldsymbol{Y}^0, \boldsymbol{X}^L) \\ &\overset{L}{\underset{l=1}{:=}} \text{FFN}\left(\text{CROSS-ATT}\left(\text{SELF-ATT}(\boldsymbol{Y}^{l-1}), \boldsymbol{X}^L\right)\right)\end{aligned} \quad (4)$$

where CROSS-ATT represents multi-head cross-attention. Additionally, each encoder and decoder layer is surrounded by residual connection [22] and layer normalization [23].

3) Global-Local Adapters: Adapters are additional modules interpolated between layers of a pre-trained model[1].

Note that the core attribute of adapters is the exceedingly small number of parameters compared with the entire pre-trained language model, which paves the way for efficient fine-tuning and communication cost reduction. Specifically, we craft two types of adapters tailored for the federated learning framework.

- ***Global adapter*** plays the role of parameter container, which receives aggregated parameters from the server and generates the output distribution that provides global knowledge to the local client. Note that the global adapter is only responsible for passing parameters and will not be optimized.
- ***Local adapter*** combined with the pre-trained language model servers as the final client meeting summarization model, which is core to mitigate the non-IID data learning challenge. Instead of directly adopting server parameters as local adapter parameters, the local adapter is optimized towards its local domain by training on the local private dataset while deriving global knowledge from the global adapter.

[1]We conduct preliminary experiments and find that it is more effective to only add adapters between decoder layers of the pre-trained language model for the meeting summarization task. Similar conclusions are corroborated by Dai et al. [24].

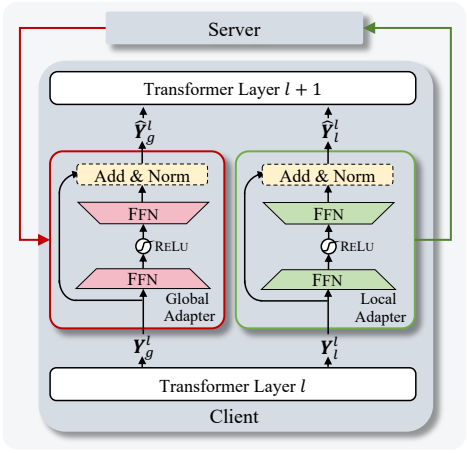


Fig. 3. Illustration of the adapter architecture. Two types of adapters are added between transformer layers, including the global adapter and the local adapter. Both adapters share the same architecture, comprising a down-projection feed-forward layer, a non-linear activation function, an up-projection feed-forward layer and a residual connection module equipped with layer normalization. The global adapter receives parameters from the server and provides global knowledge, whereas the local adapter is co-optimized through training on the local data and distilling knowledge from the global adapter. The updated parameters are then transmitted to the server for the next round of learning.

Despite their distinct functions, the two types of adapters share an identical architecture. Specifically, we have adopted the bottleneck adapter architecture exemplified in Houlsby et al. [15] for our adapters. Precisely, each adapter consists of two feed-forward layers, one non-linear activation function, and a residual connection module with layer normalization. The overall architecture is illustrated in Figure 3.

Formally speaking, given the adapter bottleneck dimension m, each adapter first utilizes a down-projection feed-forward module with learnable parameter $\boldsymbol{W}_{\text{down}} \in \mathbb{R}^{n \times m}$ to project the input $\boldsymbol{Y}^l \in \mathbb{R}^n$ into the m-dimensional representation[2], where $\boldsymbol{Y}^l$ is produced by the l_{th} transformer decoder layer of the pre-trained language model and m is smaller than n. Subsequently, a non-linear activation function `ReLu` and an up-projection feed-forward module with learnable parameter $\boldsymbol{W}_{\text{up}} \in \mathbb{R}^{m \times n}$ are employed to project the vector back into n-dimensional representation. Finally, a residual connection and layer normalization is applied to produce final $\hat{\boldsymbol{Y}}^l$.

$$\hat{\boldsymbol{Y}}^l \leftarrow \texttt{LayerNorm}(\boldsymbol{Y}^l + \texttt{ReLu}(\boldsymbol{Y}^l \boldsymbol{W}_{\text{down}}) \boldsymbol{W}_{\text{up}}) \quad (5)$$

Concretely, upon processing by the global adapter, $\boldsymbol{Y}^l$ and $\hat{\boldsymbol{Y}}^l$ are instantiated as $\boldsymbol{Y}_g^l$ and $\hat{\boldsymbol{Y}}_g^l$, respectively. Likewise, when processed by the local adapter, $\boldsymbol{Y}^l$ and $\hat{\boldsymbol{Y}}^l$ are instantiated as $\boldsymbol{Y}_l^l$ and $\hat{\boldsymbol{Y}}_l^l$, respectively.

[2]We use the notation $\boldsymbol{Y}$ since our adapters are added between transformer decoder layers.

C. Selective Knowledge Distillation Strategy

We first introduce our motivation to leverage knowledge distillation as the client-side optimization algorithm, and then present our selective strategy that further boosts the performance.

1) Motivation: Despite the apparent benefits of federated learning, the non-IID data learning setting leads to the domain drift problem of the client model. In other words, directly using the global parameters derived by aggregating updates from distinct different domains makes it impossible for local models to focus on their own domain. To remedy this issue, previous efforts have discovered knowledge distillation as one performant method [20]. Building on this foundation, we first put forward our optimization method by dexterously unifying both the adapter-based parameter-efficient fine-tuning strategy and the knowledge distillation method, which not only reduces communication costs but also facilitates the robust learning of non-IID data across clients. Additionally, global knowledge from the server is not always informative and beneficial [25]. To address this concern, we devise a selective strategy, culminating in our final client-side optimization algorithm $\mathcal{A}_c$, AdaFedSelectKD, which adaptively and robustly distils credible global knowledge to the client model. Algorithm 1 shows the entire AdaFedSelectKD algorithm.

2) Knowledge Distillation: Knowledge distillation is a solid method for transferring knowledge of the teacher model to the student model by minimizing the discrepancy between the outputs from two models with a proxy dataset [26].

Formally speaking, given the training set $\mathbb{D}$, for each training instance $(\mathcal{X}, \mathcal{Y}) \in (\mathbb{X}, \mathbb{Y})$, we obtain the final-layer representations, $\hat{\boldsymbol{Y}}_g^L$ and $\hat{\boldsymbol{Y}}_l^L$, produced by the global adapter and the local adapter, respectively. After being transformed by the language head, which projects the representation into $|V|$-dimensional probability distributions (after `softmax` operation), we obtain the outputs q_g and q_l, respectively. Within our framework, we regard q_g as the output of the teacher model and q_l as the output of the student model. Consequently, the local adapter can be trained utilizing a linear combination of two loss functions.

$$\mathcal{L} = (1-\lambda)\mathcal{L}_{CE}\left(q_l, y\right) + \lambda \mathcal{L}_{KL}\left(q_l, q_g\right) \tag{6}$$

where $\mathcal{L}_{CE}$ represents the cross-entropy loss between the predicted distribution q_l and the one-hot true label y. $\mathcal{L}_{KL}$ denotes the Kullback-Leibler divergence between q_g and q_l. The scalar λ serves to determine the weight between the two loss terms in the overall objective function.

3) Selective Strategy: The immaturity parameters provided by the server inevitably introduce useless information to local model learning [27]. To alleviate this problem, we draw inspiration from previous works in classification [28] and summarization [29], which employ the *entropy* as a measure of uncertainty, and devise a selective strategy to adaptively distill the knowledge provided by the server.

In detail, when training the t_{th} target word y_t of the instance $(\mathcal{X}, \mathcal{Y})$, we have a normalized $|V|$-dimensional probability distribution $q_g = [q_g^1, q_g^2, ..., q_g^{|V|}]$, where $|V|$ is the vocabulary size. Given this, the entropy of q_g is defined as:

$$H(q_g) = -\sum_{i=1}^{|V|} P\left(q_g^i\right) \log P\left(q_g^i\right) \tag{7}$$

We assume that ***the higher the entropy, the more uncertain the knowledge provided by the global adapter***, which means global knowledge with high entropy has no confidence in handling the current learning situation, thereby needing to be ignored. Based on this assumption, we finally propose our selective knowledge distillation strategy whereby the knowledge distillation loss is only accounted for when the entropy falls below a pre-defined entropy threshold τ.

$$\mathcal{L} = \begin{cases} (1-\lambda)\mathcal{L}_{CE}\left(q_l, y_t\right) + \lambda \mathcal{L}_{KL}\left(q_l, q_g\right) & \text{if } H(q_g) < \tau \\ \mathcal{L}_{CE}\left(q_l, y_t\right) & \text{otherwise.} \end{cases} \tag{8}$$

Algorithm 1 Adapter-based Federated Selective Knowledge Distillation Algorithm.
$\mathbb{C}$ is the client set, $\boldsymbol{W}$ is the learnable parameters, $\mathbb{D}$ is the meeting summarization dataset, E is the number of local epochs and η is the learning rate.

procedure SERVER EXECUTES:
  initialize $\boldsymbol{W}^1$
  for each round $r = 1, 2, \dots$ **do**
    for each client $\mathcal{C}_i \in \mathbb{C}$ **in parallel do**
      $\boldsymbol{W}_i^{r+1} \leftarrow$ AdaFedSelectKD$(i, \boldsymbol{W}^r)$
    end for
    $|\mathbb{D}| = \sum_{i=1}^{|\mathbb{C}|} |\mathbb{D}_i|$
    $\boldsymbol{W}^{r+1} \leftarrow \sum_{i=1}^{|\mathbb{C}|} \frac{|\mathbb{D}_i|}{|\mathbb{D}|} \boldsymbol{W}_i^{r+1}$
  end for
end procedure

procedure AdaFedSelectKD:$(i, \boldsymbol{W})$
  for each local epoch from 1 to E **do**
    for each training instance $(\mathcal{X}, \mathcal{Y}) \in \mathbb{D}_i$ **do**
      for each training target word $y_t \in \mathcal{Y}$ **do**
        $q_g, q_l = \mathcal{M}_i(\boldsymbol{W}, (\mathcal{X}, \mathcal{Y}_{[1:y_{t-1}]}))$
        if $H(q_g) < \tau$ **then**
          $\mathcal{L} = (1-\lambda)\mathcal{L}_{CE}\left(q_l, y_t\right) + \lambda \mathcal{L}_{KL}\left(q_l, q_g\right)$
        else
          $\mathcal{L} = \mathcal{L}_{CE}\left(q_l, y_t\right)$
        end if
        $\boldsymbol{W} \leftarrow \boldsymbol{W} - \eta \nabla \mathcal{L}(\boldsymbol{W}; (\mathcal{X}, \mathcal{Y}_{[1:y_{t-1}]}))$
      end for
    end for
  end for
  return $\boldsymbol{W}$ to server
end procedure

IV. Experiments

In this section, we first introduce our research questions and then present baseline methods including both non-federated and federated learning settings, and finally describe evaluation metrics and implementation details.

TABLE III
MAIN RESULTS OF ADOPTING BART-LARGE AS THE BACKBONE MODEL. THE ADACENTRALIZED IS ONE SUPER STRONG METHOD THAT EXPLICITLY TRAINS THE MODEL ON DATA FROM ALL THREE DOMAINS. CLIENT MODELS OBTAINED VIA DIFFERENT METHODS ARE TESTED ON THE DOMAIN-SPECIFIC TEST SET OF EACH CLIENT. † AND †† INDICATE THE FIRST-RANKED AND SECOND-RANKED RESULTS RESPECTIVELY. RESULTS ARE AVERAGED OVER THREE RANDOM RUNS.

Client	Setting	Method	ROUGE-1	ROUGE-2	ROUGE-L
Academic	Single	ADASINGLE	24.83	5.70	17.24
	Centralized	ADACENTRALIZED	**26.74**††	**7.95**†	**20.73**†
	Federated	ADAFEDAVG	25.76	6.18	18.70
	Federated	ADAFEDKD	26.66	7.02	19.26
	Federated	ADAFEDSELECTKD	**27.09**†	**7.62**††	**19.79**††
Committee	Single	ADASINGLE	32.38	13.08	23.24
	Centralized	ADACENTRALIZED	**34.59**††	**15.79**†	**25.32**††
	Federated	ADAFEDAVG	33.80	14.04	24.19
	Federated	ADAFEDKD	34.12	14.65	24.83
	Federated	ADAFEDSELECTKD	**34.70**†	**15.19**††	**25.37**†
Product	Single	ADASINGLE	31.83	11.11	21.27
	Centralized	ADACENTRALIZED	**34.53**†	**12.78**††	**23.71**†
	Federated	ADAFEDAVG	32.50	12.19	22.45
	Federated	ADAFEDKD	33.09	12.50	23.11
	Federated	ADAFEDSELECTKD	**33.32**††	**12.86**†	**23.53**††

A. Research Questions

Our experiments are intended to address the following research questions:

- **Research Question 1**: How does the proposed ADAFEDSELECKD perform, and is it comparable to powerful centralized training methods?
- **Research Question 2**: How well does the proposed ADAFEDSELECKD generalize? Can it achieve good performance under a variety of settings, particularly under more severe non-IID data situations?
- **Research Question 3**: How does the proposed selective knowledge distillation strategy work specifically and what are the underlying mechanisms?

B. Baseline Methods

Our baseline methods can be divided into two categories: non-federated learning and federated learning. All adopt the adapter-based pre-trained language model as the backbone model.

- ADASINGLE.
  Setting: Non-federated learning setting.
  Model: The model only has the local adapter that will be optimized.
  Method: Training and testing the model with data from a single domain.
- ADACENTRALIZED.
  Setting: Non-federated learning setting.
  Model: The model only has the local adapter that will be optimized.
  Method: Training the model using the whole QMSum that covers all three domains and testing the model using only one single-domain data. *Centralized training methods are always viewed as one super strong baseline for federated methods.*
- ADAFEDAVG.
  Setting: Federated learning setting.
  Model: The model has one type of adapter that will be optimized during training.
  Method: The fundamental federated learning algorithm, where clients hold client-specific parameters and perform local updates based on their private data via maximum likelihood estimation. Afterwards, the server gathers the weighted average of all client updates as the new global parameters, which will be distributed to all clients as new client-specific parameters.
- ADAFEDKD.
  Setting: Federated learning setting.
  Model: The model has two types of adapters including both the global adapter and the local adapter. The local adapter is optimized during training.
  Method: Each client performs local updates via knowledge distillation while the server employs federated averaging as the parameter-aggregation algorithm.
- ADAFEDSELECTKD.
  Setting: Federated learning setting.
  Model: The model has two types of adapters including both the global adapter and the local adapter. The local adapter is optimized during training.
  Method: Based on the ADAFEDKD, the selective strategy is introduced to filter out global knowledge.

C. Evaluation Metrics

We adopt the standard metrics ROUGE [30] for evaluation and obtain the F_1 scores for ROUGE-1, ROUGE-2, and ROUGE-L that measures the word-overlap, bigram-overlap and longest common sequence between the ground-truth and the generated summary respectively. We use the implementation provided by HuggingFace[3].

[3]https://github.com/huggingface/evaluate

TABLE IV
MAIN RESULTS OF ADOPTING BART-BASE AS THE BACKBONE MODEL. THE ADACENTRALIZED IS ONE SUPER STRONG METHOD THAT EXPLICITLY TRAINS THE MODEL ON DATA FROM ALL THREE DOMAINS. CLIENT MODELS OBTAINED VIA DIFFERENT METHODS ARE TESTED ON THE DOMAIN-SPECIFIC TEST SET OF EACH CLIENT. † AND †† INDICATE THE FIRST-RANKED AND SECOND-RANKED RESULTS RESPECTIVELY. RESULTS ARE AVERAGED OVER THREE RANDOM RUNS.

Client	Setting	Method	ROUGE-1	ROUGE-2	ROUGE-L
Academic	Single	ADASINGLE	23.97	5.70	17.81
	Centralized	ADACENTRALIZED	**25.66**†	**6.72**†	**18.88**††
	Federated	ADAFEDAVG	24.19	6.19	18.20
	Federated	ADAFEDKD	24.84	6.25	18.32
	Federated	ADAFEDSELECTKD	**25.25**††	**6.48**††	**18.98**†
Committee	Single	ADASINGLE	27.68	9.41	19.30
	Centralized	ADACENTRALIZED	**28.61**††	**11.23**†	**21.21**†
	Federated	ADAFEDAVG	28.09	9.87	20.31
	Federated	ADAFEDKD	28.21	10.40	20.49
	Federated	ADAFEDSELECTKD	**28.88**†	**11.14**††	**21.07**††
Product	Single	ADASINGLE	28.47	9.86	19.85
	Centralized	ADACENTRALIZED	**30.83**†	**11.15**††	**20.92**††
	Federated	ADAFEDAVG	29.34	10.47	20.42
	Federated	ADAFEDKD	29.89	10.56	20.74
	Federated	ADAFEDSELECTKD	**30.41**††	**11.25**†	**21.15**†

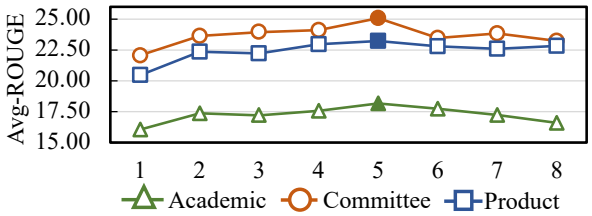


Fig. 4. Effect of the entropy threshold τ for the BART-large model.

TABLE V
AVERAGE ROUGE RESULTS OF DIFFERENT FEDERATE ALGORITHMS.

	Academic	Committee	Product
ADAFEDAVG	16.88	24.01	22.38
ADAFEDOPT	17.58	24.36	22.72
ADAFEDPROX	17.89	24.57	23.04
ADAFEDNOVA	18.09	24.81	23.11
ADAFEDSELECTKD	**18.17**	**25.08**	**23.23**

D. Implementation Details

We use the Flower framework to simulate the federated learning environment[4]. Specifically, we establish one central server and three distributed clients for the academic, committee, and product domains, respectively. During each round of federated learning, all three clients are engaged in the training process, indicating a client participation rate of 100%. The server employs the federated averaging algorithm to aggregate the gathered information. On the client side, we employ both BART [16] and LED [17] as the backbone model to conduct experiments. For the BART-large model and LED-large model, adapters are added to the top six transformer decoder layers with an adapter bottleneck dimension of 2048. For the BART-base model, adapters are added to the top three transformer decoder layers with an adapter bottleneck dimension of 1536. During the training phase, all parameters within the local adapter—spanning the down-projection feed-forward module, the up-projection feed-forward module, and layer normalization—are updated. Conversely, the parameters of the global adapter remain frozen. For each client, we used the AdamW optimizer with a learning rate of 2e-4 and a batch size of 16. The weight decay is set to 0.01. The loss weight λ is set to 0.2. We conducted exhaustive hyperparameter search experiments to determine the final entropy threshold τ. Figure 4 illustrates the search results for the BART-large model[5]. Accordingly, the entropy threshold τ for the selective strategy is set to 5 across all model variants[6].

V. RESULTS

A. Research Question 1

To answer the first research question "*How does the proposed* ADAFEDSELECKD *perform, and is it comparable to powerful centralized training methods?*", we conduct both automatic evaluations by comparing various methods and human evaluations to comprehensively access the performance.

1) Automatic Evaluation: The results illustrated in Tables III, IV and VI correspond to the BART-large, BART-based and LED-large backbone models, respectively. To sum up, the following conclusions can be drawn. Firstly, the outcomes confirm that our proposed ADAFEDSELECTKD outperforms the baseline method ADAFEDAVG, improving the ROUGE score by approximately 1.2 points. Secondly, compared with ADAFEDKD, our optimized ADAFEDSELECTKD demonstrates superior performance, which confirms that the selective strategy constitutes the vital component for robust and efficacious federated knowledge distillation. Thirdly, the results also validate that our ADAFEDSELECTKD can achieve comparable even superior performance relative to ADACENTRALIZED,

[4]https://github.com/adap/flower

[5]Search experiments for other model variants show the same results.

[6]Our codes and models will be made public.

TABLE VI
MAIN RESULTS OF ADOPTING LED-LARGE AS THE BACKBONE MODEL. THE ADACENTRALIZED IS ONE SUPER STRONG METHOD THAT EXPLICITLY TRAINS THE MODEL ON DATA FROM ALL THREE DOMAINS. CLIENT MODELS OBTAINED VIA DIFFERENT METHODS ARE TESTED ON THE DOMAIN-SPECIFIC TEST SET OF EACH CLIENT. † AND †† INDICATE THE FIRST-RANKED AND SECOND-RANKED RESULTS RESPECTIVELY. RESULTS ARE AVERAGED OVER THREE RANDOM RUNS.

Client	Setting	Method	ROUGE-1	ROUGE-2	ROUGE-L
Academic	Single	ADASINGLE	24.28	6.03	17.58
	Centralized	ADACENTRALIZED	**26.30**†	**6.94**††	**18.64**††
	Federated	ADAFEDAVG	25.13	6.34	18.19
	Federated	ADAFEDKD	25.49	6.58	18.30
	Federated	ADAFEDSELECTKD	**25.92**††	**7.09**†	**18.67**†
Committee	Single	ADASINGLE	32.69	13.03	22.58
	Centralized	ADACENTRALIZED	**33.71**†	**14.94**††	**23.87**†
	Federated	ADAFEDAVG	32.72	13.88	23.22
	Federated	ADAFEDKD	32.98	14.39	23.46
	Federated	ADAFEDSELECTKD	**33.53**††	**14.95**†	**23.85**††
Product	Single	ADASINGLE	30.12	9.85	19.70
	Centralized	ADACENTRALIZED	**32.72**†	**10.75**†	**22.17**††
	Federated	ADAFEDAVG	30.43	9.93	21.66
	Federated	ADAFEDKD	31.69	10.12	21.93
	Federated	ADAFEDSELECTKD	**32.07**††	**10.67**††	**22.64**†

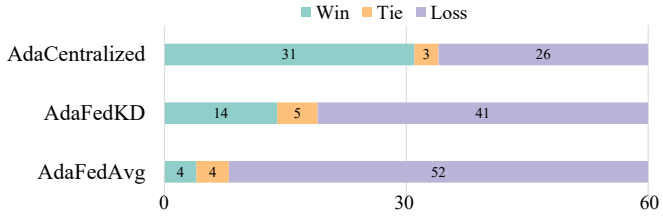


Fig. 5. Generated meeting summary comparison of ADAFEDSELECTKD with other methods on 60 randomly-chosen meetings. For example, compared with ADAFEDSELECTKD, ADAFEDAVG performs better on 4 of the 60 summaries and worse on 52.

which is a piece of solid evidence to verify the effectiveness of our method. Fourthly, the improvements achieved across the three model variants indicate the stability and generalizability of our method. **Due to the better results based on the BART-large backbone model, the following experiments are all based on the BART-large model**.

To further substantiate the effectiveness of ADAFEDSELECTKD in managing non-IID data, we conduct comparative analyses against three established federated learning algorithms: FedOPT [31], FedProx [32], and FedNova [33]. These algorithms are particularly tailored for non-IID data scenarios. For the sake of ensuring consistency across models, adapter-based models are utilized for all clients involved in the study. The outcomes of these comparisons are detailed in Table V. From these results, we infer two key insights: firstly, ADAFEDOPT, ADAFEDPROX, and ADAFEDNOVA, being designed for non-IID contexts, exhibit superior performance in comparison to the conventional ADAFEDAVG approach; secondly, despite the competitive landscape, our ADAFEDSELECTKD emerges as the most effective, thereby underscoring its robustness and superior capability in handling non-IID data challenges.

2) Human Evaluation: We employ three evaluators to undertake our human evaluation. All three evaluators are researchers in natural language processing who are well-versed in the task of meeting summarization. Each evaluator is remunerated $30 for this evaluation task.

First, 20 meetings are randomly selected per domain, amounting to 60 meetings in total. For each meeting, we generate its meeting summary based on four methods: ADACENTRALIZED, ADAFEDAVG, ADAFEDKD and ADAFEDSELECTKD. Each evaluator is provided with the meeting and a pair of summaries generated based on ADAFEDSELECTKD and another method respectively, in random order. Ehe evaluator determines which summary is better (wins) or decides a tie between the two summaries according to domain expertise, informativeness and factual correctness of summaries.

We count the number of wins, ties and losses for each method, with the average results across the three evaluators (Figure 5). These observations point to the conclusion that our method exhibits an impressive win rate of up to 87% vis-à-vis the baseline ADAFEDAVG method. It is noteworthy that ADAFEDSELECTKD achieves competitive results compared with the strong ADACENTRALIZED, with a 43% win rate. Additionally, the comparison with ADAFEDKD also proves the necessity of our designed selective strategy.

B. Research Question 2

To answer the second research question "*How well does the proposed* ADAFEDSELECKD *generalize? Can it achieve good performance under a variety of settings, particularly under more severe non-IID data situations?*", we set up various experimental settings to provide more comparisons.

1) IID and Balanced Data Setting: Under this setting, each client maintains meeting summarization data of the same distribution (IID), with equal quantities distributed across the three clients (balanced). First, we evenly divide the data of the three domains into three parts respectively. Then, for the data of each domain, we distribute the three divided parts into the three clients respectively, resulting in our IID and balanced data setting, wherein each client holds one-third of the data in each of the three domains. Subsequently, we conduct experiments leveraging ADAFEDAVG, ADAFEDKD

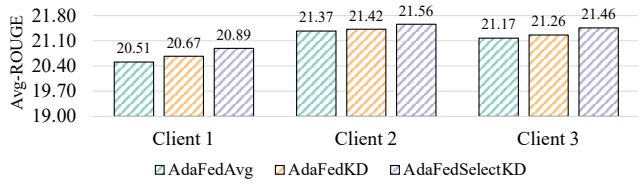


Fig. 6. Average ROUGE results based on the IID and balanced data setting, where each client maintains meeting summarization data of the same distribution (IID) and holds the same amount of data instances (balanced).

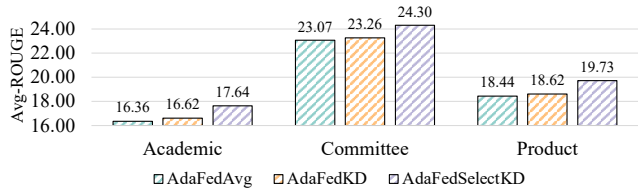


Fig. 7. Average ROUGE results based on the non-IID and balanced data setting, where each client maintains domain-specific meeting summarization data (non-IID) and holds the same amount of data instances (balanced).

and AdaFedSelectKD based on this newly curated data. The results (shown in Figure 6) are averaged over three random runs, with the data being randomly re-divided for each run. We find that given the IID and balanced data, all three clients demonstrate similar performance, with our AdaFedSelectKD being more effective compared with other federated baselines. Despite the effectiveness of both AdaFedKD and AdaFedSelectKD, we find they contribute marginally under this setting. Our evidence reinforces the findings of previous works that federated knowledge distillation methods excel at overcoming the challenge of non-IID data learning but contribute little under the IID data setting.

2) Non-IID and Balanced Data Setting: Under this setting, each client maintains domain-specific meeting summarization data (non-IID), with equal quantities distributed across the three clients (balanced). Specifically, for each domain, we randomly select 200 training instances, 40 validation instances and 40 testing instances from the corresponding QMSum portion, resulting in balanced data quantities across the three clients. Subsequently, we conduct experiments using AdaFedAvg, AdaFedKD and AdaFedSelectKD based on this newly curated balanced data. The results (depicted in Figure 7) are averaged over three random runs, with data randomly re-selected for each run. Firstly, it is evident that the ROUGE results show varying degrees of decline due to the reduction of data quantity relative to the full amount of data. Secondly, under this well-formed setting, the utility of our AdaFedSelectKD is more fully exploited, with over 1 point ROUGE improvement directly compared with AdaFedKD. We attribute this to the fact that the balanced data setting, which facilitates a consistent parameter optimization process on the client side, thus generating stable global parameters that allow our entropy-based selective strategy to make reliable distillation decisions. Thirdly, the results reveal that despite having the same amount of data across all three clients, the committee client attains superior performance. As indicated in Table II, this can be attributed to the committee meeting's

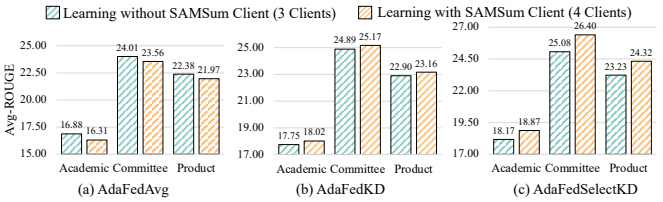


Fig. 8. Average ROUGE results based on the extreme non-IID and unbalanced data setting. "Learning without SAMSum client" means the original three clients participate in the federated learning process while "Learning with SAMSum client" means there are four clients in total with the newly-added SAMSum client joining the learning process.

TABLE VII
AVERAGE ROUGE RESULTS FOR NEW DOMAIN ADDITION USING ADAFEDSELECTKD. "BASELINE" REPRESENTS THE INITIAL THREE CLIENTS IN THE FEDERATED LEARNING PROCESS. "POST-ADDITION" ADDS THE FOURTH SAMSUM CLIENT AFTER THE FIRST THREE CONVERGE. "PRE-ADDITION" INCLUDES THE FOURTH CLIENT FROM THE BEGINNING, LEARNING ALONGSIDE THE INITIAL THREE CLIENTS.

	Academic	**Committee**	**Product**	**SAMSum**
Baseline	18.17	25.08	23.23	-
Post-addition	18.62	26.07	23.79	34.25
Pre-addition	**18.87**	**26.40**	**24.32**	**34.86**

fewer turns and reduced input tokens, making it easier to train an effective summarizer.

3) Extreme Non-IID and Unbalanced Data Setting: To further verify the effectiveness and robustness of our method, we set up an extreme non-IID and unbalanced data distribution setting to assess the performance of different methods. To this end, we employ the SAMSum dialogue summarization dataset [34] and establish the fourth client, which will participate in the federated learning process along with the previous three clients. Specifically, SAMSum is a widely-used dataset for the dialogue summarization task, which is vastly different from QMSum. The number of instances (more than 16000 instances), topics (in various scenes of real life), the length of the dialogue (120 tokens on average), the length of the summary (23 tokens on average) and the number of turns (11 turns on average) all differ greatly from QMSum. We conduct experiments leveraging AdaFedAvg, AdaFedKD and AdaFedSelectKD. The results are shown in Figure 8. Firstly, according to Figure 8(a), we find the previous three clients do not benefit from the newly introduced SAMSum client and actually perform worse. This is in line with the previous conclusion that the federated averaging algorithm has severe limitations in the presence of non-IID data. Secondly, on the contrary, knowledge distillation-based federated learning algorithms exhibit their advantages under this challenging setting, with improvements regarding the ROUGE score, as shown in Figure 8(a) and (b). Thirdly, it is worth noting that AdaFedSelectKD achieves the best results, demonstrating its efficacy when dealing with extreme non-IID data.

Building on the results discussed earlier, we broaden our experimental framework to explore the possibility of continuing federated training with the inclusion of new domain clients after the initial federated learning phase has concluded. To investigate this, we commence with training the Academic,

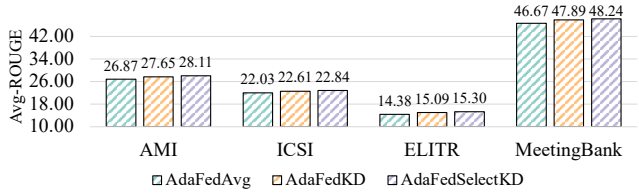


Fig. 9. Average ROUGE score for AMI, ICSI, ELITR and MeetingBank meeting summarization datasets.

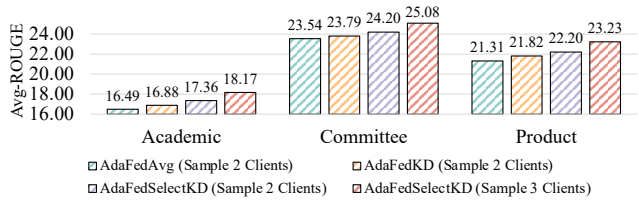
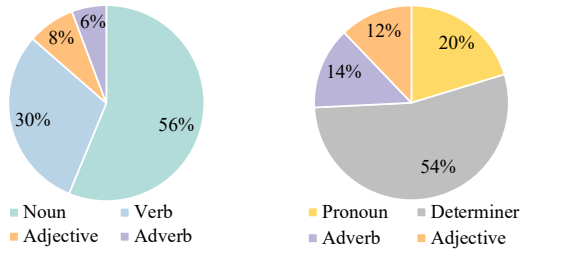


Fig. 10. Average ROUGE score based on the client sampling setting. "Sample 2 clients" means that during each round of learning, 2 clients are randomly selected to participate in the learning process.



(a) POS tag distribution of words learned using KD loss.

(b) POS tag distribution of words learned without using KD loss.

Fig. 11. Part-of-speech tag distribution for academic target summary words learned (a) using knowledge distillation loss and (b) without using knowledge distillation loss.

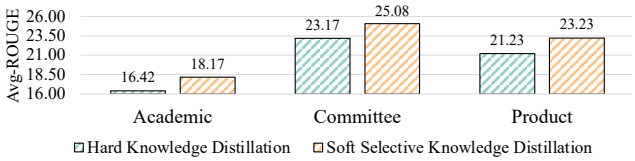


Fig. 12. Averaged ROUGE results based on the noun–verb-based hard knowledge distillation and our proposed selective knowledge distillation strategy.

Committee, and Product clients using the AdaFedSelectKD methodology. Upon reaching convergence with these initial clients, we introduce an additional, fourth client from the SAMSum domain to extend the training process. We term the scenario involving this subsequent integration of the SAMSum client as the "post-addition" scenario, in contrast to the "pre-addition" scenario where the SAMSum client is involved from the start. The comparative outcomes are detailed in Table VII. It is evident from these results that our AdaFedSelectKD strategy surpasses baseline performances in both scenarios of new domain client integration. Notably, the "pre-addition" scenario outperforms the "post-addition" scenario. We believe this is because the "pre-addition" approach can better ensure the consistency of parameter updates among clients, thereby achieving globally optimal performance.

4) Long Meeting Setting: In this scenario, considering that meetings often involve longer input texts, we select four datasets: AMI [35], ICSI [36], ELITR [37], and MeetingBank [9], and set them up as four clients to conduct experiments using our proposed AdaFedSelectKD. It is noteworthy that, due to the longer input length of these datasets, we chose to use the LongT5 [38] for our experiments. The LongT5 model allows for a maximum input length of 16,384, enabling effective modelling of the aforementioned lengthy meeting texts. The experimental results are shown in Figure 9. The results clearly demonstrate that AdaFedSelectKD not only outperforms competing methods but also exhibits exceptional generalizability across different model architectures and a variety of meeting summarization datasets, thereby underscoring its robustness and adaptability.

5) Client Sampling Setting: In this scenario, we simulate a more pragmatic federated learning environment, in which only a subset of clients participate in each round of the learning process. Specifically, we set the participation rate to 70%, meaning that in our setup, two clients are randomly chosen to participate in the learning procedure during each round. The results are illustrated in Figure 10. Firstly, the experiment shows that utilizing all three clients — academic, committee and product — leads to better performance than learning from only two clients. Secondly, our proposed method AdaFedSelectKD consistently and stably outperforms the other baseline methods.

C. Research Question 3

To answer the third research question "*How does the proposed selective knowledge distillation strategy work specifically and what are the underlying mechanisms?*", we assess the training process by examining whether appropriate target summary words are learned using knowledge distillation and their part-of-speech (POS) tag distribution.

For each client, we extract all target summary words trained from the first round until the final, optimally-performing round to determine whether each summary word adopts knowledge distillation loss and its POS tag information. For example, at the first client, suppose there is only one training instance with the corresponding summary "project manager decides to use lcd", and after two rounds of training, the model achieves its best performance with three target summary words: manager (once), decides (twice). These are optimized using the knowledge distillation loss. Therefore, the proportion of words optimized via knowledge distillation is $3/(2 \times 6) = 25\%$, where 2 is the number of learning rounds and 6 is the number of words in the sentence.

Accordingly, our statistics show that knowledge distillation loss is 86%, 88%, and 89% utilized in generating the target summary words during the training process for the academic, committee, and product clients, respectively. Furthermore, we calculate the part-of-speech tag distribution of target summary words learned with and without knowledge distillation. Figure 11 illustrates the outcomes for the academic client, while the other two clients exhibit similar distributions. The

TABLE VIII
MEETING SUMMARIES FROM OUR THREE DOMAINS GENERATED BY DIFFERENT METHODS.

Method	Meeting Summaries
	Academic
ADAFEDAVG	PhD C explained that there were various delays with different components along the processing chain.
ADAFEDSELECTKD	PhD C said that there were delays of 100ms for silence, 40ms at the input, and 10ms from LDA filters.
GOLD STANDARD	PhD C explained that the silence probabilities had a 100ms delay, the delta at the input had a 40ms delay, and a 10ms delay was created by LDA filters.
	Committee
ADAFEDAVG	The governing organization developed the framework and simplified the options for resolving issues outside of legal proceedings.
ADAFEDSELECTKD	The National Police Chiefs' Council streamlined out-of-court disposals by developing the police approach.
GOLD STANDARD	The National Police Chiefs' Council was responsible for developing the police approach to out-of-court disposals and simplifying the range of out-of-court disposals.
	Product
ADAFEDAVG	Industrial Designer proposed an eco-friendly option but Project Manager agreed more with the commercially-appealing proposal.
ADAFEDSELECTKD	The Industrial Designer suggested solar panel and rechargeable batteries but the Project Manager preferred Project Manager's cradle idea.
GOLD STANDARD	Industrial Designer proposed to incorporate a solar panel and rechargeable batteries, but Project Manager agreed more with Marketing's proposal to include a cradle.

observations suggest that nouns and verbs constitute nearly all of the words optimized through knowledge distillation loss, aligning with the intuition that both nouns and verbs are essential for articulating the core and domain-specific ideas of the meetings. In contrast, determiners and pronouns make up 74% of all words learned without using knowledge distillation loss.

This distribution insight on the target summary words elicits a natural follow-up: "*How effective would it be to use knowledge distillation by absolute means only when learning target nouns and verbs?"*. To address this question, we conduct experiments by exclusively adopting knowledge distillation rigorously upon learning nouns and verbs. Figure 12 shows the results. We can clearly find that our ADAFEDSELECTKD, which adaptively makes the distillation decision exhibits superior performance compared to the hard distillation method, demonstrating the necessity of our designed selective strategy.

D. Case Study

Table VIII illustrates summaries of meetings from three domains generated using various methodologies. We observe that the baseline method, ADAFEDAVG, consistently produces generic meeting summaries lacking in detailed information. In contrast, our proposed method, ADAFEDSELECTKD, yields summaries that are more informative and tailored to the domain. Moreover, the gold standard meeting summaries continue to demonstrate advantages in conciseness and informativeness, highlighting the challenges intrinsic to meeting summarization.

VI. RELATED WORK

A. Meeting Summarization

Meeting summarization [2], [3], [39] aims to pack crucial information of a given meeting into a concise yet comprehensive summary highlighting the most salient points. In addition to the challenges inherent in traditional summarization tasks, meeting summarization must address unique difficulties arising from its multi-participant nature. To facilitate progress in this domain, various datasets have been curated [6], [9], [10], [35], [36], enabling the development of state-of-the-art models that incorporate versatile knowledge [4], [5], [7], [40]–[43] and achieve the best results. However, privacy concerns, which are inextricably intertwined with meeting content, have received little attention in the literature, hampering real-world application. Lee and Sogaard [8] take the initiative to address this issue by exploring differential privacy (DP) [44] for meeting summarization, focusing primarily on a single domain. In this work, we conduct the first systematic study of meeting summarization under the federated learning framework, accounting for the heterogeneity and unbalance of data across multiple domains.

B. Federated Learning

Federated learning enables collaborative machine learning without the centralized collection of potentially sensitive raw data, thereby paving the way for stronger privacy guarantees when building predictive models [19]. With mounting concerns regarding privacy issues, this paradigm has garnered significant research interest including diverse research directions [13]. In particular, owing to the inevitable inclusion of private information in texts, a variety of studies have explored diverse natural language processing tasks within the federated learning framework [14]. The predominant efforts in this realm have focused on natural language understanding tasks, such as spoken language understanding [45] and text classification [46]. Recent years have witnessed a trend toward applying the federated learning framework to natural language

generation tasks [47]. Additionally, amid the rapid ascent of Large Language Models (LLMs), several studies offer their perspectives on federated LLMs. [48]. Our work follows this line of work and is the first to explore the federated multi-domain meeting summarization task.

C. Parameter-efficient Fine-tuning

The field of natural language processing is currently dominated by large language models [49]. Despite their superiority, fine-tuning all the parameters of these immense models on various downstream tasks becomes prohibitively complicated as both model size and number of tasks increase [50]. To alleviate this problem, parameter-efficient fine-tuning is coming to the rescue by updating only a small number of extra parameters while keeping most pre-trained parameters frozen [15], [51]. Opportunely, such lightweight alternatives are well-suited for reducing communication costs in the federated learning framework. Based on this foundation, the amalgamation of federated learning and parameter-efficient fine-tuning unveils vast potential for diverse applications [52]. In this paper, we craft two types of adapter modules, a global adapter and a local adapter, which collaboratively and efficiently facilitate federated client–server communication.

D. Knowledge Distillation

Knowledge distillation refers to the process of transferring knowledge from a teacher model to a student model without significant performance degradation. It has proven to be an effective method for improving model performance [26]. In recent years, knowledge distillation has been applied in the federated learning framework and has demonstrated its ability to mitigate the effects of data heterogeneity [20]. Our proposed framework builds upon this and takes one step further to explore the combination of both knowledge distillation and parameter-efficient fine-tuning while introducing one carefully designed selective strategy to enable an adaptive learning process.

VII. Limitations and Potential Advancements

The aforementioned experiments are conducted using the dataset provided by the research community and based on a simulated federated environment. This may not fully reflect the complexities of real-world scenarios and could potentially lead to two limitations:

1. Well-curated dataset exhibits less variability. In reality, meetings, even those within the same domain, vary with respect to their participants, discussion topics, and duration. Moreover, meeting transcripts are typically generated via automatic speech recognition (ASR) systems, resulting in noisy, imperfect textual data. Useful forms of meeting summaries also depend on the target user needs, spanning full-text summaries, highlighted extracts, identification of action items, and more. While our research attempts to address real-world limitations, the dataset employed in our experiment likely does not adequately capture the complexities of real-world meeting settings. We envision that future advancements can develop more appropriate benchmarks, design more comprehensive experimental paradigms, and possibly even collaborate with corporations to narrow the divide between research explorations and practical real-world applications.

2. Simulated federated environment lacks uncertainty. In practical federated learning deployments, addressing the challenges of non-IID and unbalanced data is still insufficient. It is also imperative to overcome various issues stemming from communication uncertainties, such as client–server latency, asynchronous client learning updates, and client dropout. These factors are unaddressed in the current research, and present challenges to any federated learning techniques. As such, future advancements could conduct larger-scale decentralized experiments in which the federated learning procedure is intentionally perturbed to stress-test such a framework's robustness to uncertainty, thereby approximating real-world conditions more closely.

VIII. Conclusion and Future Work

We examine the multi-domain meeting summarization task under a federated learning paradigm. We show that this represents a more pragmatic and realistic configuration than prior work on learning meeting summary models over centralized meeting data. Moreover, to mitigate two challenges, namely limited server–client communication and the non-IID data learning situation, we propose a unified method, AdaFedSelectKD, which succeeds in reducing communication costs and addressing the domain drift problem of the client model. Through comprehensive empirical studies, our method demonstrates its effectiveness and robustness that can achieve comparable results with centralized training methods while exhibiting its superiority in handling the intricacies of non-IID data.

We believe that future work will strive to apply the proposed method to real scenarios. In our own work, we aim to craft data resources and design experimental settings to adequately simulate real-world federated learning circumstances. We plan to collaborate with organizations to implement the federated learning framework and evaluate our proposed method in addressing a variety of exigencies. Further investigation will build from this foundation, to incorporate differential privacy techniques to further augment our model's privacy preservation characteristics. In addition, we also plan to apply the AdaFedSelectKD framework to a broader range of summarization and text generation tasks, such as dialogue response generation and financial data-to-text generation tasks, thereby further extending the applicability of our proposed method. Beyond this, we are also considering extending our algorithm to more complex scenarios, such as cases where there is collaboration between clients. This involves designing a server-side hierarchical aggregation method and a client-side transfer learning approach from both data and algorithmic perspectives to further enhance performance. All these advances promise to make AdaFedSelectKD a practical solution for meeting summarization and more diverse tasks.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their insightful comments. This work was supported by the Hong Kong Innovation and Technology Support Programme Platform Research Project fund (ITS/269/22FP), the National Key R&D Program of China via grant No. 2021ZD0112905, National Natural Science Foundation of China (NSFC) via grant U22B2059 and 62276078, the Key R&D Program of Heilongjiang via grant 2022ZX01A32, the International Cooperation Project of PCL, PCL2022D01 and the Fundamental Research Funds for the Central Universities (Grant No.HIT.OCEF.2023018).

REFERENCES

[1] S. Banerjee, P. Mitra, and K. Sugiyama, "Generating abstractive summaries from meeting transcripts," *Proceedings of the 2015 ACM Symposium on Document Engineering*, 2015.

[2] L. P. Kumar and A. Kabiri, "Meeting summarization: A survey of the state of the art," *ArXiv preprint*, 2022.

[3] V. Rennard, G. Shang, J. Hunter, and M. Vazirgiannis, "Abstractive meeting summarization: A survey," *ArXiv preprint*, 2022.

[4] X. Feng, X. Feng, B. Qin, and X. Geng, "Dialogue discourse-aware graph model and data augmentation for meeting summarization," in *Proc. of IJCAI*, 2020.

[5] C. Zhu, R. Xu, M. Zeng, and X. Huang, "A hierarchical network for abstractive meeting summarization with cross-domain pretraining," in *Proc. of EMNLP Findings*, 2020.

[6] M. Zhong, D. Yin, T. Yu, A. Zaidi, M. Mutuma, R. Jha, A. H. Awadallah, A. Celikyilmaz, Y. Liu, X. Qiu, and D. Radev, "QMSum: A new benchmark for query-based multi-domain meeting summarization," in *Proc. of NAACL*, 2021.

[7] Z. Liu and N. F. Chen, "Dynamic sliding window modeling for abstractive meeting summarization," in *Proc. of Interspeech*, 2022.

[8] S. Lee and A. Sogaard, "Private meeting summarization without performance loss," 2023.

[9] Y. Hu, T. J. Ganter, H. Deilamsalehy, F. Dernoncourt, H. Foroosh, and F. Liu, "Meetingbank: A benchmark dataset for meeting summarization," *ArXiv preprint*, 2023.

[10] H. Wu, M. Zhan, H. Tan, Z. Hou, D. Liang, and L. Song, "Vcsum: A versatile chinese meeting summarization dataset," *ArXiv preprint*, 2023.

[11] H. McCosker, A. Barnard, R. Gerber *et al.*, "Undertaking sensitive research: Issues and strategies for meeting the safety needs of all participants," in *Forum Qualitative Sozialforschung/Forum: Qualitative Social Research*, 2001.

[12] A. Jain, L. Hong, and S. Pankanti, "Biometric identification," *Communications of the ACM*, 2000.

[13] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *arXiv: Artificial Intelligence*, 2019.

[14] M. Liu, S. Ho, M. Wang, L. Gao, Y. Jin, and H. Zhang, "Federated learning meets natural language processing: A survey," *ArXiv preprint*, 2021.

[15] N. Houlsby, A. Giurgiu, S. Jastrzebski, B. Morrone, Q. de Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, "Parameter-efficient transfer learning for NLP," in *Proc. of ICML*, 2019.

[16] M. Lewis, Y. Liu, N. Goyal, M. Ghazvininejad, A. Mohamed, O. Levy, V. Stoyanov, and L. Zettlemoyer, "BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension," in *Proc. of ACL*, 2020.

[17] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *ArXiv preprint*, 2020.

[18] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge distillation: A survey," *International Journal of Computer Vision*, 2020.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. of AISTATS*, 2017.

[20] A. Mora, I. Tenison, P. Bellavista, and I. Rish, "Knowledge distillation for federated learning: a practical guide," *ArXiv preprint*, 2022.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. of NeurIPS*, 2017.

[22] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. of CVPR*, 2016.

[23] J. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *ArXiv preprint*, 2016.

[24] D. Dai, L. Dong, S. Ma, B. Zheng, Z. Sui, B. Chang, and F. Wei, "StableMoE: Stable routing strategy for mixture of experts," in *Proc. of ACL*, 2022.

[25] Y. He, Y. Chen, X. Yang, H. Yu, Y.-H. Huang, and Y. Gu, "Learning critically: Selective self-distillation in federated learning on non-iid data," *IEEE Transactions on Big Data*, 2022.

[26] G. E. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," *ArXiv preprint*, 2015.

[27] D. Qi, H. Zhao, and S. Li, "Better generative replay for continual federated learning," *ArXiv preprint*, 2023.

[28] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *Proc. of ICML*, 2016.

[29] L. van der Poel, R. Cotterell, and C. Meister, "Mutual information alleviates hallucinations in abstractive summarization," in *Proc. of EMNLP*, 2022.

[30] C.-Y. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Text Summarization Branches Out*, 2004.

[31] S. J. Reddi, Z. B. Charles, M. Zaheer, Z. Garrett, K. Rush, J. Konecný, S. Kumar, and H. B. McMahan, "Adaptive federated optimization," *ArXiv*, vol. abs/2003.00295, 2020. [Online]. Available: https://api.semanticscholar.org/CorpusID:211678094

[32] A. K. Sahu, T. Li, M. Sanjabi, M. Zaheer, A. Talwalkar, and V. Smith, "Federated optimization in heterogeneous networks," *arXiv: Learning*, 2018. [Online]. Available: https://api.semanticscholar.org/CorpusID:59316566

[33] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, "Tackling the objective inconsistency problem in heterogeneous federated optimization," *Advances in neural information processing systems*, 2020.

[34] B. Gliwa, I. Mochol, M. Biesek, and A. Wawer, "SAMSum corpus: A human-annotated dialogue dataset for abstractive summarization," in *Proceedings of the 2nd Workshop on New Frontiers in Summarization*, 2019.

[35] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The ami meeting corpus: A pre-announcement," in *International workshop on machine learning for multimodal interaction*, 2005.

[36] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *Proc. of ICASSP*, 2003.

[37] A. Nedoluzhko, M. Singh, M. Hledíková, T. Ghosal, and O. Bojar, "Elitr minuting corpus: A novel dataset for automatic minuting from multi-party meetings in english and czech," in *International Conference on Language Resources and Evaluation*, 2022. [Online]. Available: https://api.semanticscholar.org/CorpusID:250157963

[38] M. Guo, J. Ainslie, D. C. Uthus, S. Ontañón, J. Ni, Y.-H. Sung, and Y. Yang, "Longt5: Efficient text-to-text transformer for long sequences," in *NAACL-HLT*, 2021. [Online]. Available: https://api.semanticscholar.org/CorpusID:245144820

[39] X. Feng, X. Feng, and B. Qin, "A survey on dialogue summarization: Recent advances and new frontiers," *ArXiv preprint*, 2021.

[40] K. Riedhammer, B. Favre, and D. Z. Hakkani-Tür, "A keyphrase based approach to interactive meeting summarization," *2008 IEEE Spoken Language Technology Workshop*, 2008.

[41] M. Zhong, Y. Liu, Y. Xu, C. Zhu, and M. Zeng, "Dialoglm: Pre-trained model for long dialogue understanding and summarization," in *Proc. of AAAI*, 2022.

[42] Y. Zhang, A. Ni, T. Yu, R. Zhang, C. Zhu, B. Deb, A. Celikyilmaz, A. H. Awadallah, and D. Radev, "An exploratory study on long dialogue summarization: What works and what's next," in *Proc. of EMNLP Findings*, 2021.

[43] Y. Zhang, A. Ni, Z. Mao, C. H. Wu, C. Zhu, B. Deb, A. Awadallah, D. Radev, and R. Zhang, "Summ^n: A multi-stage summarization framework for long input dialogues and documents," in *Proc. of ACL*, 2022.

[44] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, 2014.

[45] Z. Huang, F. Liu, and Y. Zou, "Federated learning for spoken language understanding," in *Proc. of COLING*, 2020.

[46] Z. Li, S. Si, J. Wang, and J. Xiao, "Federated split bert for heterogeneous text classification," *2022 International Joint Conference on Neural Networks (IJCNN)*, 2022.

[47] B. Y. Lin, C. He, Z. Ze, H. Wang, Y. Hua, C. Dupuy, R. Gupta, M. Soltanolkotabi, X. Ren, and S. Avestimehr, "FedNLP: Benchmarking federated learning methods for natural language processing tasks," in *Proc. of ACL Findings*, 2022.

[48] C. Chen, X. Feng, J. Zhou, J. Yin, and X. Zheng, "Federated large language model: A position paper," *ArXiv*, vol. abs/2307.08925, 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:259950775

[49] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proc. of NeurIPS*, 2020.

[50] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, "Towards a unified view of parameter-efficient transfer learning," in *Proc. of ICLR*, 2022.

[51] X. L. Li and P. Liang, "Prefix-tuning: Optimizing continuous prompts for generation," in *Proc. of ACL*, 2021.

[52] H. Zhao, W. Du, F. Li, P. Li, and G. Liu, "Fedprompt: Communication-efficient and privacy-preserving prompt tuning in federated learning," *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022.