Leveraging Unstructured Text Data for Federated Instruction Tuning of Large Language Models

Rui Ye[1]* **Rui Ge**[1]* **Fengting Yuchi**[1] **Jingyi Chai**[1] **Yanfeng Wang**[2,1] **Siheng Chen**[1,2]

[1] Shanghai Jiao Tong University [2] Shanghai AI Laboratory

Abstract

Federated instruction tuning enables multiple clients to collaboratively fine-tune a shared large language model (LLM) that can follow humans' instructions without directly sharing raw data. However, existing literature impractically requires that all the clients readily hold instruction-tuning data (i.e., structured instruction-response pairs), which necessitates massive human annotations since clients' data is usually unstructured text instead. Addressing this, we propose a novel and flexible framework FedIT-U2S, which can automatically transform unstructured corpus into structured data for federated instruction tuning. FedIT-U2S consists two key steps: (1) few-shot instruction-tuning data generation, where each unstructured data piece together with several examples is combined to prompt an LLM in generating an instruction-response pair. To further enhance the flexibility, a retrieval-based example selection technique is proposed, where the examples are automatically selected based on the relatedness between the client's data piece and example pool, bypassing the need of determining examples in advance. (2) A typical federated instruction tuning process based on the generated data. Overall, FedIT-U2S can be applied to diverse scenarios as long as the client holds valuable text corpus, broadening the application scope of federated instruction tuning. We conduct a series of experiments on three domains (medicine, knowledge, and math), showing that our proposed FedIT-U2S can consistently and significantly brings improvement over the base LLM.

1 Introduction

Instruction tuning has become one of the most imperative components in training contemporary instruction-followed large language models (LLMs) [1, 2, 3, 4], where typically, the training samples are collected from diverse sources by a central party [5, 6, 7]. However, these data could contain sensitive (e.g., private or proprietary) information that cannot be directly shared, making such centralized learning paradigm inapplicable especially for domains such as medicine [8] and finance [9].

Addressing this, federated learning [10, 11] has emerged as a well-suited technique to achieve instruction tuning of LLMs without direct data sharing. In federated instruction tuning (FedIT), each party (i.e., client) keeps its private data locally and shares the instruction-tuned LLM with the central server, while the server aggregates LLMs from multiple parties and distributes the aggregated LLM back to participating parties. Such paradigm has attracted massive attention and interests from both academia [12, 13, 14] and industry [15, 16, 17].

Despite extensive efforts dedicated to FedIT, existing methods impractically rely on the assumption that each party possesses structured instruction-tuning data (i.e., instruction-response pairs), which

[*] Equal contribution.

International Workshop on Federated Foundation Models In Conjunction with NeurIPS 2024 (FL@FM-NeurIPS'24).

significantly constrains the real-world applicability of FedIT. In practice, while clients may possess valuable data locally, this data often exists in an unstructured format (just strings of text) rather than naturally aligns with the structured format required for IT [18]. Consequently, current FedIT systems face challenges in scalability, as they necessitate manual annotation of data by each client.

To fill this gap, we propose a novel and flexible framework FedIT-U2S, which can automatically transform unstructured corpus into structured instruction-tuning data for FedIT, bypassing the massive human efforts required for data annotation. Specifically, FedIT-U2S consists of two key steps: few-shot instruction-tuning data generation and FedIT on the generated data. (1) The server first distributes an open-sourced general LLM and a few examples (could be as few as only one) to participating clients. During data generation, each client queries the LLM to generate multiple instruction pairs, where each pair is generated by feeding the LLM with a prompt that is composed of few examples as the context and a sampled piece of its unstructured data. To further enhance the generality and scalability of FedIT-U2S, we propose a retrieval-based example selection approach, where for each sampled piece of unstructured data, similarity scores are computed by comparing it with all the examples sent from the server, after which the top-k examples are selected as the few-shot examples in the context for data generation. (2) Subsequently, typical federated instruction tuning is launched based on the general LLM and the generated datasets in the previous step. Considering communication and computation efficiency, LoRA [19] is applied and therefore only a small set of parameters are learned and communicated. Overall, our FedIT-U2S framework makes FedIT system as practical as Google's GBoard application (next word prediction) [20], where the supervision data directly comes from user's data without any manual effort.

To verify the effectiveness of our proposed framework, we conduct a series of experiments covering three domains (i.e., medicine, knowledge, and math). We show that across these domains, our FedIT-U2S consistently improves the performance of the general LLM on the corresponding downstream task. Besides, we show the effectiveness of several designs, including retrieval-based example selection and filtering during data generation, providing potential directions for further improving the performance of FedIT-U2S.

Our contributions are as follows:

1. We propose the first end-to-end framework (FedIT-U2S) for directly leveraging unstructured data for federated instruction tuning of large language models.
2. We propose a retrieval-based example selection technique and a few-shot data generation mechanism, which automatically selects examples for higher relatedness and generates structured data in an expected manner.
3. We verify the effectiveness of FedIT-U2S through a series of experiments on multiple domains.

2 Related Work

Federated Learning of Large Language Models. Federated learning is a privacy-preserving machine learning paradigm that enables multiple clients to collaboratively train machine learning models without sharing their raw data [10, 11]. With the rise of large language models (LLMs), researchers have recently begun to consider federated training of LLMs to safeguard client data privacy or to address the scarcity of publicly available data [21, 12], which has attracted massive attention and interests from both academia [12, 13, 14] and industry [15, 16, 17].

Specifically, OpenFedLLM [12] offers an integrated framework and provides a comprehensive empirical study to show the potential of federated instruction tuning of LLMs (FedIT). Similarly, FederatedScope-LLM [17] and FedML-LLM [15] provide frameworks that implement FedIT; while FedLLM-Bench [13] offers real-world datasets and benchmarks. Besides frameworks and benchmarks [22], a series of methods are proposed to target various perspectives including safety alignment [23], privacy [24], heterogeneous computation [25].

However, existing literature assumes that client data is structured in the form of instruction-response pairs, overlooking the reality that client data often exists in an unstructured format. In such cases, clients are required to annotate data before participating in FedIT, which is labor-intensive and limits its broader adoption. In this paper, we address this issue for the first time by proposing FedIT-U2S, a method that automates the transformation of unstructured client data into structured data prior to FedIT. This reduces the need for manual annotation and broadens the applicability of FedIT.

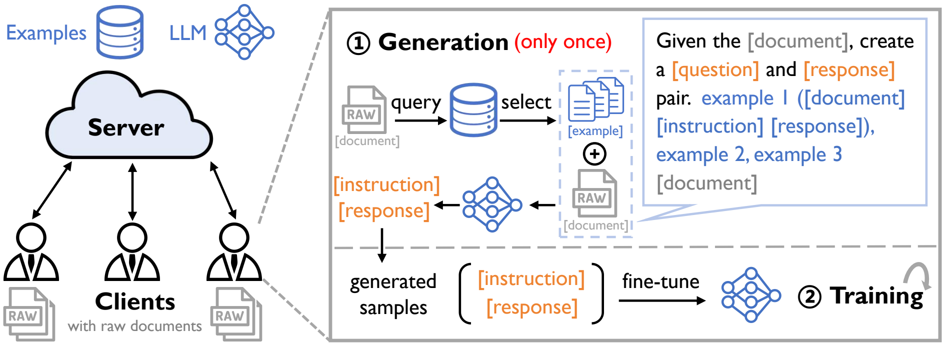


Figure 1: Overview of our proposed FedIT-U2S. It consists of two key steps: data generation and FedIT. Data generation is required only once before FedIT. (1) For each raw unstructured data piece, clients select a few examples by retrieving from an example database to construct a few-shot template, prompting the LLM to generate an instruction-response pair. (2) Typical federated instruction tuning starts based on the generated structured instruction-tuning data.

Data Generation in Large Language Models. The quality and quantity of data play a critical role in the training of large language models. However, manually generating and annotating data is labor-intensive and hard to scale up. Addressing this, the community turns to using LLMs to generate high-quality data [26, 27, 28, 29]. For example, Self-Instruct [30] leverages 8 in-context examples to prompt LLMs for generating new instruction samples. WizardLM [26] instructs ChatGPT to generate diverse instructions via evolving prompt. MATRIX [31] instructs the LLMs to generate data for value alignment via social simulation. Genie [18] employs few-shot methods [32] to transform unstructured data into three kinds of structured data. Instruction Pre-training [27] generates instruction-tuning data to augment pre-training.

In this paper, we for the first time consider utilizing clients' unstructured data for FedIT of LLMs by leveraging the LLMs for data generation. We apply few-shot generation technique for its simplicity and effectiveness; while we believe that there could be other techniques applied to our scenario.

3 Methodology

In this section, we first introduce the overall framework of our proposed FedIT-U2S (Figure 1), which consists of two key steps: few-shot instruction-tuning data generation (which transforms unstructured data into structured instruction-tuning data pairs) and federated instruction tuning on the generated data. Then, we detail our design of retrieval-based example selection for few-shot data generation.

3.1 Pipeline of FedIT-U2S

At the beginning of FedIT-U2S, the server first distributes an open-sourced general LLM (denoted by $\boldsymbol{\theta}^*$) and a set of examples (unstructured and structured text pairs, denoted by $\mathcal{O}$) to participating clients.

Step 1: few-shot instruction-tuning data generation. Suppose there are M clients in the system and each client m holds an unstructured dataset $\mathcal{D}_m^u = \{\boldsymbol{d}_i\}_{i=1}^{N_m}$, where $\boldsymbol{d}_i$ is a data piece and N_m denotes the number of data pieces. Since such unstructured data cannot be directly used for instruction tuning, it conventionally requires each client's efforts to manually create instruction-response pairs for tuning, which is costly and faces the challenges of scaling up. To address this, we design to automatically transform the unstructured data into a structured instruction-response format via a few-shot data generation process, which leverages LLM's in-context learning capability [32].

Specifically, upon receiving example set $\mathcal{O} = \{(\boldsymbol{d}_i, \boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{O}$, where O is the example number, $\boldsymbol{d}_i$ is an unstructured data document, $\boldsymbol{x}_i$ and $\boldsymbol{y}_i$ is the document-grounded instruction and response respectively, each client selects several (denoted by k) examples as few-shot examples prompt the LLM $\boldsymbol{\theta}^*$. Denote the instruction for generation as I and the selected examples as $\mathcal{S} = \{(\hat{\boldsymbol{d}}_i, \hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i)\}_{i=1}^{k}$, given a user's data piece $\boldsymbol{d}$, the prompt P is constructed as: $P = Concat(I, \mathcal{S}, \boldsymbol{d})$, where $Concat$

denotes the concatenation operation (see full prompt in A). Note that these examples can be either randomly selected for diversity or selected according to relatedness between user data and examples for better diversity-relatedness trade-off, which will be detailed in Section 3.2. Based on the prompt, the LLM $\boldsymbol{\theta}^*$ will generate an instruction-response pair: $(\boldsymbol{x}, \boldsymbol{y}) = f(P; \boldsymbol{\theta}^*)$. Therefore, by iterating on client's unstructured dataset $\mathcal{D}_m^u = \{\boldsymbol{d}_i\}_{i=1}^{N_m}$, we obtain a structured dataset for instruction tuning: $\mathcal{D}_m^s = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N_m}$.

Since the responses of LLMs are in an open-ended form and there are randomness during generation, some generated data might fall short in terms of data quality. Therefore, additional data filtering is necessary for enhancing the data quality. Here, we consider two filtering mechanisms: rule-based filtering to remove data with undesired format and reward-based filtering to ensure the quality of selected data. Specifically, we first filter out data that does not follow the format of instruction-response pair. Secondly, we use an publicly available reward model to score the generated data samples and select the top two-thirds samples. This enables us to select data that is more aligned with human preference since reward model is trained to model human preference.

Step 2: federated instruction tuning on the generated data. With the generated data, a typical process of federated instruction tuning is started. Considering computation and communication efficiency, we apply LoRA [19] as the parameter-efficient fine-tuning technique. Suppose there are T rounds of federated learning rounds in total. At each round t, the server sends the model parameters $\boldsymbol{\theta}^t$ to each available client. Then, each client m initializes its local trainable parameters with $\boldsymbol{\theta}^t$, keeps the base model parameters $\boldsymbol{\theta}^*$ fixed, and starts supervised fine-tuning on its generated dataset $\mathcal{D}_m^s = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^{N_m}$, where the model learns to predict the response $\boldsymbol{y}_i$ given the instruction $\boldsymbol{x}_i$. By fine-tuning for several steps, each client m obtains a fine-tuned model parameters $\boldsymbol{\theta}_m^t$ and sends it to the server. Finally, the server aggregates model parameters of clients to obtain the global model parameters for the next round: $\boldsymbol{\theta}^{t+1} = \sum_m p_m \boldsymbol{\theta}_m^t$, where $p_m = \frac{N_m}{\sum_i N_i}$ is the relative dataset size of client m.

3.2 Retrieval-based Example Selection for Few-Shot Generation

The chosen examples (i.e., the context) in the prompt could significantly affect the behaviour of LLMs [33, 34], resulting in different quality of the genrated data. Therefore, to generate high-quality structured data, selecting appropriate few-shot examples is essential. Generally, examples that closely match the target text in terms of content and structure tend to produce more effective results. However, in practical applications, manually identifying suitable examples can be a time-consuming process, making it inflexible in adapting to diverse scenarios. To mitigate this challenge, we propose a retrieval-based example selection method for few-shot generation which automatically selects few-shot examples from a mixed example pool according to similarity between user data and examples.

Given the set of examples sent from the server $\mathcal{O} = \{(\boldsymbol{d}_i, \boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{O}$, each client aims to select k examples for each of its sampled unstructured data piece. Specifically, for each data piece $\boldsymbol{d}$, we compute the similarity $Sim(\boldsymbol{d}, \boldsymbol{d}_i)$ for each $\boldsymbol{d}_i$ in the example pool $\mathcal{O}$ using BERT Score as the metric, which gives a similarity score that reflects the relatedness between the target data piece and the example's content. Subsequently, we rank the similarity scores and select top-k examples $\mathcal{S} = \{(\hat{\boldsymbol{d}}_i, \hat{\boldsymbol{x}}_i, \hat{\boldsymbol{y}}_i)\}_{i=1}^{k}$, which are mostly likely to guide the LLM to generate high-quality and highly-related data. The other procedures remain unchanged as in Section 3.1.

4 Experiments

4.1 Experimental Details

Training Dataset. We consider three datasets for our experiments [27], which cover domains including medicine, knowledge, and math. Specifically, PubMedQA [35] is a medical dataset for biomedical research question answering with corresponding abstracts as the context. HotpotQA [36] is a dataset of Wikipedia-based questions with supporting facts as the context. AQUA_RAT [37] is a math dataset for algebraic word problems answering. The problems, together with solutions, form the context. We select 10,000 samples from each dataset for the experiments [27], with each sample comprising a piece of original unstructured text, along with a human annotated instruction

	PubMedQA		HotpotQA		AQUA_RAT	
	BERT Score	ROUGE-L	BERT Score	ROUGE-L	BERT Score	ROUGE-L
Base Model	0.1483	0.1496	0.0566	0.2380	-0.0171	0.1529
FedIT-U2S	0.1876	0.1727	0.1774	0.2942	0.0885	0.2383
FedIT-U2S (Filtered)	0.2043	0.1859	0.2439	0.3226	0.1131	0.2452
FedAvg on Human Data	0.2306	0.2017	0.2701	0.3531	0.1381	0.2890

Table 1: Experiments on three datasets: PubMedQA (medical), HotpotQA (knowledge), and AQUA_RAT (math). Our proposed FedIT-U2S consistently brings performnace improvement compared to base model. FedIT-U2S (Filtered) hugely fills the gap between base model and FedAvg on human-annotated data, indicating the effectiveness of our proposed method in bypassing massive human efforts in annotation.

and response, both derived from the text. Only the unstructured text is used in our method FedIT-U2S, while the human annotated instruction-response pairs are used to implement FedAvg as a reference to verify the effectiveness of our method.

Implementation Details. Our implementation is based on the open-sourced codebase OpenFedLLM[*] [12]. We use Vicuna-7B [38] as the base model and set the learning rate as $2e^{-5}$ with a batch size of 16. The communication round is set to 200 and 2 clients are sampled out of 5 each round to participate federated instruction tuning. We use *reward-model-deberta-v3-large-v2* as the reward model following [18]. We select $k = 3$ examples for few-shot generation.

Evaluation Metrics. (1) BERT Score: BERT Score [39] is an evaluation metric for natural language generation that measures the similarity between a candidate sentence and reference sentences by leveraging contextual embeddings from pre-trained language models like BERT. **(2) ROUGE-L:** ROUGE-L [40] is an evaluation metric used for summarization and text generation tasks, focusing on the longest common subsequence (LCS) between a candidate sentence and a reference sentence. ROUGE-L evaluates the extent to which the candidate sentence preserves the order and content of the reference, providing a more holistic assessment of the generated text's quality. We select 50 samples from each dataset to serve as the test set. We compare the model-generated responses to the gold standard answers (i.e., human-annotated answers in the test set) by calculating BERT Score and ROUGE-L to assess performance.

Compared LLMs. (1) The base LLM, i.e., the Vicuna model without additional tuning; (2) the base LLM tuned via FedAvg on human-annotated data, which serves as a performance reference; (3) the base LLM tuned by our FedIT-U2S without filtering technique; and (4) the base LLM tuned by our FedIT-U2S with filtering technique.

4.2 Experimental Results

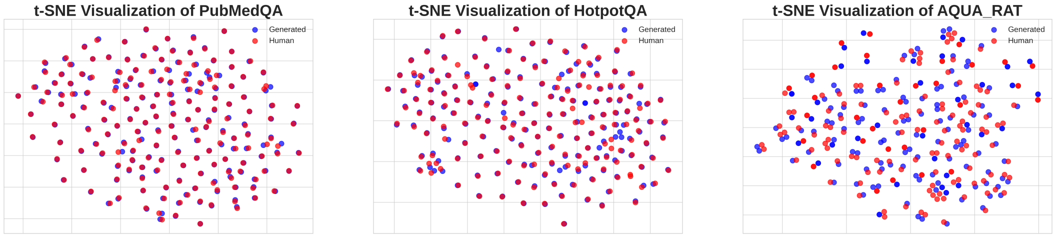
Comparisons with baselines. In Table 1, we compare models trained via our methods on generated data with base model and model trained via FedAvg [10] on human-annotated data (as a reference). Experiments are conducted on three datasets and evaluated by two metrics. From the table, we see that (1) our methods consistently and significantly improves the performance of the base model across datasets and evaluation metrics, indicating the effectiveness of our proposed methods. Specifically, in HotpotQA, our method can achieve 0.1873 higher BERT Score (0.2439 v.s. 0.0566). (2) Our methods hugely fill the gap between base model and that tuned via FedAvg on human data, further verifying FedIT-U2S's effectiveness. However, there is still a room for improvement, calling for more future works to further enhance the performance. With the increasing generation capability of LLMs [29, 28], we even believe that there is potential for surpassing this baseline (FedAvg on human data). (3) Although the data filtered using the reward model is smaller in quantity, it brings a more significant improvement to the model's performance, indicating the importance of data quality in this scenario.

Analysis of example selection for few-shot generation. The effectiveness of few-shot generation may heavily rely on the chosen examples in the context. Therefore, here, we deeply analyze the example selection by conducting a series of experiments on HotpotQA dataset since we observe a

[*] https://github.com/rui-ye/OpenFedLLM

Experimental Setup	Bert Score	ROUGE-L
Base Model	0.0566	0.2380
① Random 0 + 3 (3 out-domain examples)	0.0868	0.2211
② Random 1 + 2 (1 in-domain and 2 out-domain examples)	0.1143	0.2426
③ Fixed 3 + 0 (3 fixed in-domain examples)	0.1774	0.2942
④ Random 3 + 0 (3 randomly selected in-domain examples)	0.2128	0.3054
⑤ **Retrieval-based Selection from A Mixed Pool**	0.2035	0.2994

Table 2: Experiments on HotpotQA dataset for analysis of example selection during few-shot data generation. The results show that our proposed automated retrieval-based selection technique can achieves comparable performance compared to selecting in-domain examples (which requires prior knowledge).


t-SNE Visualization of PubMedQA

t-SNE Visualization of HotpotQA

t-SNE Visualization of AQUA_RAT

(a) PubMedQA

(b) HotpotQA

(c) AQUA_RAT

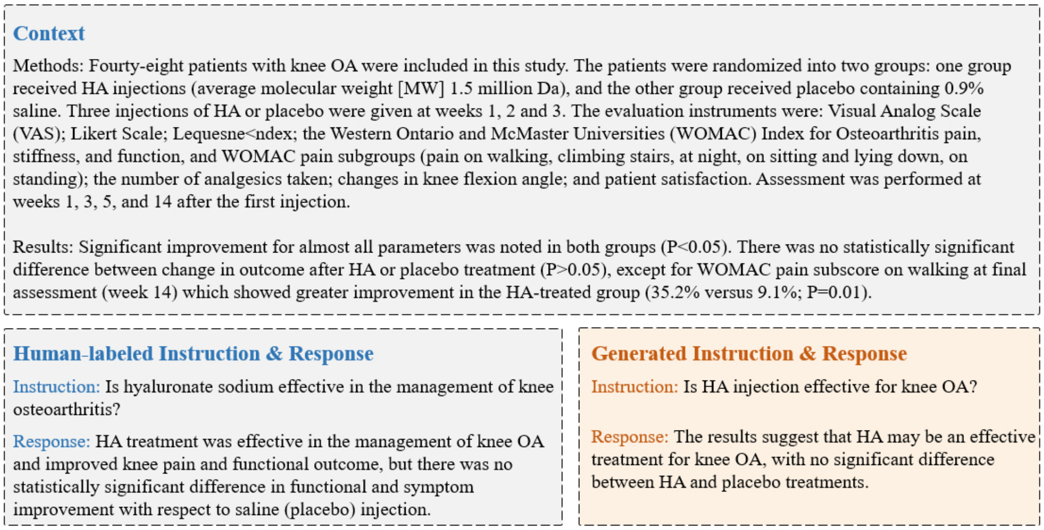
Figure 2: The t-SNE visualization of embeddings of instruction-response pairs in PubMedQA, HotpotQA and AQUA_RAT. Blue dots represent generated data, while red dots represent human-annotated data. The close proximity of each pair of red and blue dots indicates that the generated data closely aligns with the human-annotated data.

large improvement in previous experiments. In this experiment, the example pool has 50 samples in total, covering five domains: medicine, math, knowledge, common sense, and daily life. We consider the following setups of few-shot generation in our proposed FedIT-U2S: ① Random 0 + 3: 3 out-domain examples are randomly selected (e.g., for medical task, examples from other domains are randomly selected); ② Random 1 + 2: 1 in-domain and two out-domain examples are randomly selected; ③ Fixed 3 + 0: 3 fixed in-domain examples are selected for all generation; ④ Random 3 + 0: 3 in-domain examples are randomly selected; ⑤ Retrieval-based Selection: 3 examples are automatically selected from a mixed example pool by our retrieval-based example selection technique.

The experimental results are shown in Table 2. (1) Compared to the base model, ①, which introduces out-domain examples for few-shot generation, does not bring evident improvement while ②-⑤ all bring consistent improvement. This indicates the importance of selecting appropriate examples for few-shot data generation. (2) Comparing ①, ②, and ④, we can see that increasing the number of in-domain examples consistently brings more performance improvement, indicating the value of introducing in-domain examples to facilitate generation. (3) Comparing ③ and ④, we see that randomly selecting in-domain examples performs better than selecting fixed examples, indicating the value of example diversity in generation. (4) Comparing ④ and ⑤, we see that our proposed retrieval-based selection from a mixed pool performs comparably to selecting examples from a in-domain pool (which requires prior knowledge), indicating the effectiveness of our retrieval-based selection technique. This result suggests that equipped with this technique, our proposed FedIT-U2S framework can be automatically deployed in various domains without much prior knowledge.

Comparisons of generated and human-annotated data. To better understand our method, we further analyze the characteristics of our generated data by comparing it with human-annotated data from two perspectives: embedding visualization and case studies.

(1) Embedding visualization: Here, we use t-SNE [41] to visualize the data points of generated and human-annotated data. For each dataset, 200 generated and human-annotated sample pairs, sharing the same context, are selected. The embeddings of the concatenated instruction and response texts



Context

Methods: Fourty-eight patients with knee OA were included in this study. The patients were randomized into two groups: one group received HA injections (average molecular weight [MW] 1.5 million Da), and the other group received placebo containing 0.9% saline. Three injections of HA or placebo were given at weeks 1, 2 and 3. The evaluation instruments were: Visual Analog Scale (VAS); Likert Scale; Lequesne<ndex; the Western Ontario and McMaster Universities (WOMAC) Index for Osteoarthritis pain, stiffness, and function, and WOMAC pain subgroups (pain on walking, climbing stairs, at night, on sitting and lying down, on standing); the number of analgesics taken; changes in knee flexion angle; and patient satisfaction. Assessment was performed at weeks 1, 3, 5, and 14 after the first injection.

Results: Significant improvement for almost all parameters was noted in both groups (P<0.05). There was no statistically significant difference between change in outcome after HA or placebo treatment (P>0.05), except for WOMAC pain subscore on walking at final assessment (week 14) which showed greater improvement in the HA-treated group (35.2% versus 9.1%; P=0.01).

Human-labeled Instruction & Response

Instruction: Is hyaluronate sodium effective in the management of knee osteoarthritis?

Response: HA treatment was effective in the management of knee OA and improved knee pain and functional outcome, but there was no statistically significant difference in functional and symptom improvement with respect to saline (placebo) injection.

Generated Instruction & Response

Instruction: Is HA injection effective for knee OA?

Response: The results suggest that HA may be an effective treatment for knee OA, with no significant difference between HA and placebo treatments.

Figure 3: Example illustration.

are extracted via *sentence-transformers*[†] and mapped to a two-dimensional space via t-SNE. The final 2D embeddings are plotted as shown in Figure 2, where blue and red dots represent generated and human-annotated data respectively. From the figure, we observe close proximity between the generated and human data points, indicating a high degree of alignment between the generated and human data across the datasets.

(2) Case Study: In Figure 3, we show a specific example of generated data sample from PubMedQA. The human-annotated data sample with the same context is also given for comparison. Instructions of both samples ask about the effectiveness of HA injections in treating knee OA. The generated response conveys a meaning similar to human-annotated response based on the context.

These two aspects of comparison demonstrate that our generated data is highly similar to the manually annotated data in both content and structure, reflecting the high quality of the generated data.

5 Conclusions

This paper proposes FedIT-U2S, which directly leverages clients' unstructured text data to achieve federated instruction tuning of large language models. FedIT-U2S consists of two key steps: few-shot instruction-tuning data generation and federated instruction tuning on the generated data. During data generation, for each unstructured data piece, a client firstly selects related examples via a retrieval-based example selection mechanism and then uses these examples for guiding the LLM to generate instruction-response pair based on the data piece. A typical process of federated instruction tuning is then conducted based on the generated data. Experiments on three domains (medicine, knowledge, and math) verify the effectiveness of our proposed FedIT-U2S. Our method for the first time enables clients with unstructured data to be involved in the process of federated instruction tuning, which occupy a large proportion in practice and are underutilized previously. We believe that this work can contribute to broadening the application scope of federated instruction tuning.

[†] https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

References

[1] OpenAI. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

[3] Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. Mistral 7b. *arXiv preprint arXiv:2310.06825*, 2023.

[4] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

[5] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *NIPS*, 35:27730–27744, 2022.

[6] Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M Dai, and Quoc V Le. Finetuned language models are zero-shot learners. In *ICLR*, 2021.

[7] Chunting Zhou, Pengfei Liu, Puxin Xu, Srini Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. Lima: Less is more for alignment. *arXiv preprint arXiv:2305.11206*, 2023.

[8] Karan Singhal, Tao Tu, Juraj Gottweis, Rory Sayres, Ellery Wulczyn, Le Hou, Kevin Clark, Stephen Pfohl, Heather Cole-Lewis, Darlene Neal, et al. Towards expert-level medical question answering with large language models. *arXiv preprint arXiv:2305.09617*, 2023.

[9] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

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

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

[12] Rui Ye, Wenhao Wang, Jingyi Chai, Dihan Li, Zexi Li, Yinda Xu, Yaxin Du, Yanfeng Wang, and Siheng Chen. Openfedllm: Training large language models on decentralized private data via federated learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6137–6147, 2024.

[13] Rui Ye, Rui Ge, Xinyu Zhu, Jingyi Chai, Yaxin Du, Yang Liu, Yanfeng Wang, and Siheng Chen. Fedllm-bench: Realistic benchmarks for federated learning of large language models. *arXiv preprint arXiv:2406.04845*, 2024.

[14] Jianyi Zhang, Saeed Vahidian, Martin Kuo, Chunyuan Li, Ruiyi Zhang, Guoyin Wang, and Yiran Chen. Towards building the federated gpt: Federated instruction tuning. *arXiv preprint arXiv:2305.05644*, 2023.

[15] FedML Inc. Federated learning on large language models (llms). `https://doc.fedml.ai/federate/fedllm`, 2023. Accessed: 2024-03-31.

[16] Tao Fan, Yan Kang, Guoqiang Ma, Weijing Chen, Wenbin Wei, Lixin Fan, and Qiang Yang. Fate-llm: A industrial grade federated learning framework for large language models. *arXiv preprint arXiv:2310.10049*, 2023.

[17] Weirui Kuang, Bingchen Qian, Zitao Li, Daoyuan Chen, Dawei Gao, Xuchen Pan, Yuexiang Xie, Yaliang Li, Bolin Ding, and Jingren Zhou. Federatedscope-llm: A comprehensive package for fine-tuning large language models in federated learning. *arXiv preprint arXiv:2309.00363*, 2023.

[18] Asaf Yehudai, Boaz Carmeli, Yosi Mass, Ofir Arviv, Nathaniel Mills, Assaf Toledo, Eyal Shnarch, and Leshem Choshen. Genie: Achieving human parity in content-grounded datasets generation, 2024.

[19] Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. Lora: Low-rank adaptation of large language models. In *ICLR*, 2021.

[20] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. Federated learning for mobile keyboard prediction. *arXiv preprint arXiv:1811.03604*, 2018.

[21] Pablo Villalobos, Jaime Sevilla, Lennart Heim, Tamay Besiroglu, Marius Hobbhahn, and Anson Ho. Will we run out of data? an analysis of the limits of scaling datasets in machine learning. *arXiv preprint arXiv:2211.04325*, 2022.

[22] Liam Collins, Shanshan Wu, Sewoong Oh, and Khe Chai Sim. Profit: Benchmarking personalization and robustness trade-off in federated prompt tuning. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[23] Rui Ye, Jingyi Chai, Xiangrui Liu, Yaodong Yang, Yanfeng Wang, and Siheng Chen. Emerging safety attack and defense in federated instruction tuning of large language models. *arXiv preprint arXiv:2406.10630*, 2024.

[24] Youbang Sun, Zitao Li, Yaliang Li, and Bolin Ding. Improving loRA in privacy-preserving federated learning. In *The Twelfth International Conference on Learning Representations*, 2024.

[25] Yae Jee Cho, Luyang Liu, Zheng Xu, Aldi Fahrezi, Matt Barnes, and Gauri Joshi. Heterogeneous lora for federated fine-tuning of on-device foundation models. In *International Workshop on Federated Learning in the Age of Foundation Models in Conjunction with NeurIPS 2023*, 2023.

[26] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*, 2023.

[27] Daixuan Cheng, Yuxian Gu, Shaohan Huang, Junyu Bi, Minlie Huang, and Furu Wei. Instruction pre-training: Language models are supervised multitask learners, 2024.

[28] Bo Adler, Niket Agarwal, Ashwath Aithal, Dong H Anh, Pallab Bhattacharya, Annika Brundyn, Jared Casper, Bryan Catanzaro, Sharon Clay, Jonathan Cohen, et al. Nemotron-4 340b technical report. *arXiv preprint arXiv:2406.11704*, 2024.

[29] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

[30] Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language model with self generated instructions. *arXiv preprint arXiv:2212.10560*, 2022.

[31] Xianghe Pang, Shuo Tang, Rui Ye, Yuxin Xiong, Bolun Zhang, Yanfeng Wang, and Siheng Chen. Self-alignment of large language models via monopolylogue-based social scene simulation. In *Forty-first International Conference on Machine Learning*, 2024.

[32] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners, 2020.

[33] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, pages 1877–1901, 2020.

[34] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. A survey for in-context learning. *arXiv preprint arXiv:2301.00234*, 2022.

[35] Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W. Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering, 2019.

[36] Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering, 2018.

[37] Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation : Learning to solve and explain algebraic word problems, 2017.

[38] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[39] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger, and Yoav Artzi. Bertscore: Evaluating text generation with bert, 2020.

[40] Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July 2004. Association for Computational Linguistics.

[41] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

A Appendix

Listing 1: Few-shot prompt template

```
Given the next [document], create a [question] and [answer] pair that
are grounded in the main point of the document, don’t add any
additional information that is not in the document. The [question] is
by an information-seeking user and the [answer] is provided by a
helping AI Agent.

[document]: {The content of document 1}

### Response:
[question]: {The content of question 1}
[answer]: {The content of answer 1}

[document]: {The content of document 2}

### Response:
[question]: {The content of question 2}
[answer]: {The content of answer 2}

[document]: {The content document 3}

### Response:
[question]: {The content of question 3}
[answer]: {The content of answer 3}

[document]: {The content of the target text}

### Response:
```