CancersQA: Federated Learning with Pretrained Models for Intelligent Medical Diagnosis

Kunyu Yang[1, 2], Luyao Peng[1, 2], Zheng Liu[1, 3, *], Chaomurilige[1, 3, *], Yihang Dai[1, 4]

[1] Key Laboratory of Ethnic Language Intelligent Analysis and Security Governance, Ministry of Education,
Minzu University of China, Beijing 100081, China; wengyu@muc.edu.cn
[2] School of Information Engineering, Minzu University of China, Beijing 100081, China
[3] School of Chinese Ethnic Minority Languages and Literatures, Minzu University of China, Beijing 100081, China
[4] School of Software Engineering, JiangXi Agricultural University, Nanchang 330000, China
[*] Correspondence: liuzheng@muc.edu.cn (Z. L.); chaomurilige@muc.edu.cn (C.)

Abstract. Pre-trained Models (PTM) have demonstrated extraordinary efficacy in the domain of natural language processing (NLP), particularly in addressing critical challenges within question-answering (QA) systems. However, the medical QA field grapples with issues such as the low robustness of generation models and a shortage of available medical data, which is further complicated by stringent privacy requirements. Leveraging the synergy between Federated Learning (FL) and PTM, we propose an innovative CancersQA method for intelligent cancer diagnosis and effective aggregating algorithm WeiPro. This method employs the GPT-2 model in an FL framework and fine-tunes it using sparse and unevenly distributed medical consultation data. Our novel method is designed to perform optimally on small sample datasets, markedly reducing training time. Our WeiPro aggregation strategy consolidates the parameters of the FL clients by taking into account various factors. Finally, we conducted rigorous experiments to evaluate CancersQA's performance. Our experimental results indicate that our approach significantly surpasses other models in medical QA tasks underscoring its superior potential for adapting large language models to healthcare's critical domain.

Keywords: Federated learning; Pretrained model; Medical Question Answering; deep learning; natural language processing.

1 Introduction

In the current medical domain, cancer stands as one of the most challenging diseases, frequently resulting in incurable outcomes. The advent of the medical question-answering model has significantly revolutionized the healthcare sector. For example, AI techniques such as deep learning have demonstrate its great skill in medical textual processing.

Pretraining combined with LLMs are preliminary steps to a wide range of applications in medical domains such as translation of biomedical data [1-3], medical ma-

EAI MLICOM 2023, December 17, Beijing, People's Republic of China

DOI 10.4108/eai.17-12-2023.2342252

chine reading comprehension [4], etc. Large models with many learnable parameters have low error rates and are well suited for use in medical diagnostic analysis.

In particular, Abacha A.B. et al [5] proposed MEANS which is a medical QA system combined with rulebased NLP techniques, but it appears that such a system primarily processes the model's input and employs a conventional query method to derive the answer. While helpful in some cases, this approach may fall short in specific contexts, such as cancer diagnosis, where a more nuanced understanding and complex analytical processes might be required. Singhal K. et al [6-8] use large language model to learn medical knowledge and Wang S. [8] introduced image to analyze cases. Although these studies have great model performance, these LLMs are too large to train in public hospital. The method proposed by [4] focuses on enhancing the model's reading and comprehension capabilities for medical textual information. While this approach effectively improves the model's ability to understand medical content, its applicability to a medical QA system is limited, given that patients generally do not carry medical content with them during consultations. In previous researches conducted by Patel D. et al [9-11], a model was introduced that was trained using dialogue data. It is worth noting that the answers generated by this model were found to be informal and lacked professionalism. Consequently, this particular model may not be suitable for cancer question answering purposes.

Although there has been a lot of research on medical QA, either the datasets have been aggregated and trained together without considering data security, or the models have been trained on only a small portion of the dataset, resulting in poor performance. Our approach, on the other hand, not only takes privacy into account, but also maximizes the use of decentralized data and uses fine-tuning techniques that allow different hospitals to first fine-tune better results on a small local dataset, which is really adaptive in cancer question answering.

Cancer question answering is a subset of medical QA. In the hospital scenario, cancer question learning is always open domain and has a wide range of audiences. The question can be asked by professor, doctor, patients and their relatives, and the model cannot only answer "yes" or "no", but the answer must be correct and authoritative guidance. Such a powerful model with strong application capabilities is very demanding on the training data. There are even more challenges in medical data. Firstly, for quality data, there is a high cost of manual annotation and strict requirements from medical experts, especially annotation of medical text data, which has higher requirements of professionalism and understanding of the annotator. For privacy reasons, medical data is often not disclosed, and patients and their family members are sometimes reluctant to disclose relevant information, which can easily lead to data not being shared and disseminated, and the formation of data silos are highly sensitive and require strict protection that cannot be exploited. As a consequence, the lack of data flow between hospitals can lead to a limited availability of data for individual institutions to work with. Additionally, drawing inferences about specific diseases often requires comprehensive analysis of multiple factors, including disease status, descriptions, and diagnostic results. However, obtaining a substantial and comprehensive dataset that incorporates all these aspects simultaneously poses significant challenges. While data sharing could potentially enhance the volume of available data, it

also introduces risks of data breaches. Therefore, it is crucial to maintain a careful balance to ensure the integrity and confidentiality of the shared data. To address these issues, we propose a solution called CancersQA, which combines federated learning with GPT-2 to exploit the data silos and fine-tune a mutual question answering (QA) model using non-independent and identically distributed (non-IID) data. Furthermore, we introduce the WeiPro algorithm to aggregate parameters of local models, effectively integrating features from different datasets. This approach enables the model to consider comprehensive information, similar to utilizing a comprehensive dataset.

Specifically, the proposed approach initializes the original model in the central server and sends it to the client at the beginning. Then, each client updates its local model and sends the local model back to the central server in parallel instead of sharing the local dataset. Finally, they are aggregated by the WeiPro algorithm. In this way, they can train a model together and protect patient privacy. In other words, the combination of federated learning and PTMs is task-specific.

To evaluate the performance of the proposed deep learning approach under both controlled and wild conditions, this paper conducts experiments on Chinese medical dialogue dataset, cMedQA2 dataset and HuaTuo dataset. The experimental result shows that training in federated mode has a similar loss compared to centralised data sharing and has better performance than fragmented training.

The main contributions of this study can be summarized as follows:

- This paper proposes CancersQA, a medical question answering method that combines federated learning and PTMs to significantly improve the use of data silos and protect patient privacy.
- Addresses the shortcomings of fewer available datasets and less information covered by a single dataset due to the scarcity of medical datasets and improves the efficiency of training.
- Proposed WeiPro algorithm which considers more factors that may affect model performance such as data quality, loss value, participation rounds, etc. Then use the projection to measure the gaps of reference vector and practice vector which is more effective and stable than FedAvg and WeiAvg aggregating algorithm.

The rest of the paper is organized as follows. In Section 2, we describe the background of this work, including federated learning, question answering, pretrained models and fine tuning. In Section 3, we present the details of our proposed methodology. Section 4 presents the experimental results of this study. Finally, Section 5 and Section 6 present our discussion and conclusions.

2 Related Work

This section presents development and application of federated learning (from Sec. 2.1). Then illustrate question answering task in natural language processing (NLP) field and relevant datasets for QA task (from Sec. 2.2). Pretrained models (PTMs)and some fine-tuning technologies mentioned in the last paragraph which have already become a hot research topic in recent years (from Sec 2.3).

2.1 Medical question answering

One of the most important branches in QA task is medical question answering. Medical QA system, which means computer should understand the medical questions and answer quickly and accurately, most of these are have strict correctness requirements. Considering a lot of literature and articles online are not good use, many doctors still need to spend a long time to browse the retrieved information, Lee M. [12] proposed PubMed, which can generate abstract for papers by asking kinds of questions, i.e., "What is X?" by designing five blocks for text summarization. PubMed has a great performance by answering questions from six doctors.

More and more open domain question answering systems are based on deep learning techniques due to the scarcity of medical datasets. In order to fill the gap in the medical domain, Mutabazi E. et al [13] proposed the medical textual question answering systems based on deep learning approaches which had been evaluated on five public QA datasets. In recent years, due to the growing numbers of literature, medical experts and researchers about COVID-19. Raza S. [14] proposed CoQUAD, which can answer the question about any COVID-19-related from natural language effectively with higher level of accuracy and outperform the previous models. Despite many of these, the medical area is still in dire need of the introduction of large language models (LLMs) and have not yet performed optimally in biomedical domain tasks due to the need for medical expertise in the responses [15]. Huatuo created by [16] which consists of four different data from various ways. Wang H. et al [16] created HuaTuo model which has been supervised fine-tuned with generated QA (Question Answering) instances and integrated structured and unstructured medical knowledge from CMeKG and suggested SUS measures. Compared with other medical models, Huatuo reached the highest score and generated more sensible answer while facing amount of profession questions.

2.2 Federated learning and Pretrained models

Federated learning. The rise of federated learning truly changed the training pattern of deep learning, especially for natural language processing (NLP) and computer vision (CV) in Medical, financial and other fields, with an increasing emphasis on user privacy. A massive of data can be very sensitive, which make them untouchable. To deal with these problems, Google, H. Brendan et al and Konevcny et al [17, 18], proposed federated learning and point that data could be trained together while dispersed across different participants instead of sharing and centralizing. Considering five different models and experimenting on four datasets, which demonstrate that this approach is robust to the unbalanced and non-IID data distributions. However, federated learning has its own drawbacks [19] that computation and communication costs cannot be neglect when training largescale architecture. With regard to this, [20], whose lightweight framework (FedPCL) allows clients to fuse features generated by pre-trained models efficiently through prototypes sharing. PTMs such as Bert, always has a large structure. To address training difficulties, FedBERT proposed by [21] that combines the benefit of federated learning and split learning approaches improve the

computation efficiency. As a result, it can maintain its effectiveness without communicating to the sensitive local data of clients on seven GLUE tasks.

Pre-trained models. Pretraining, this word come from image field. Either computer vision (CV) or natural language processing (NLP) model is becoming deeper which is difficult to train a whole model without any learning before. Resnet [22] is a classic network that simply perform identity mapping by shortcut connections and is widely used in computer vision field. Similarly, Bert, GPT series, T5, LLAMA and other PTMs excel in natural language processing area [23]. Pretrained models can be more effective when fine-tuning these large language models which was trained in advance on others relative dataset. According to model's knowledge learned in pretraining period, PTMs always posse super compatibility with downstream tasks [24, 25]. Due to automatic metrics ROUGE may reach a bottleneck, Liu Y. et al [26] uses Bert to fine-tune a summary extraction model and the model become SOTA on CNN/Daily mail dataset. When meeting large-scale models like GPT-3 which has 175B parameters, it seems like even use fine-tuning technology model still difficult to train. Hu E.J. et al [27] discovered that to make the model perform well, they needn't to do full fine-tuning which retrains all model parameters to make the model unfeasible. They proposed LoRA, a novel training method, which freezes the pretrained model weights and injects trainable rank decomposition matrices into each layer of the transformer architecture, greatly reducing the number of trainable parameters for downstream tasks. As a result, LoRA is as same as or better than full fine-tuning on some large language models (LLMs). Fine-tuning is not always for pretrained models. Xiang Lisa Li and Percy Liang, Stanford University, in order to reduce costs of modifying all language model parameters they proposed prefix-tuning inspired by promoting, a lightweight alternative to fine-tuning for generation tasks and achieved commendable results.

3 Materials and Methods

The urgent demand for a medical QA model has become increasingly evident. However, existing researches often encounter challenges such as difficulties in training, concerns regarding data privacy, and insufficient performance. In order to address these issues, we propose the development of CancersQA, a Chinese medical QA model built using GPT-2 technology.

CancersQA adopts a federated learning approach, which presents unique challenges compared to open domain question answering matching in English. This is due to the combination of its domain restricted nature and the language-specific features of the Chinese language. By leveraging the power of federated learning and PTMs, institutions can securely handle their own data and solely transmit the updated model parameters back to the centralized center for aggregation.

This method offers enhanced data privacy protection compared to traditional data centralizing methods, while still ensuring performance that is comparable to centralized models. Through this implementation, CancersQA effectively addresses the dif-

ficulties faced by traditional medical QA model and offers a more robust and secure solution.

In this section we introduce our materials for research and proposed methods. We illustrate problem definition for CansersQA task in Sec 3.1 which is different from traditional comprehend-based QA problem and necessary dataset in Sec 3.3.

3.1 Problem definition

We know that QA tasks are someone who asks a model to answer some questions and expects to get an accurate answer for responding. These questions can be open-domain, because of their medical attribute. We make sense that open-domain question answering system can be trained as seq2seq pattern, so we maximum the following:

$$\boldsymbol{\Gamma}(\boldsymbol{T}) = \prod_{\boldsymbol{i=1}}^{\boldsymbol{m}} \boldsymbol{\Phi}(\boldsymbol{t_i} \mid \boldsymbol{t_1}, \boldsymbol{t_2}, \boldsymbol{t_3}, \dots, \boldsymbol{t_{i-1}}; \boldsymbol{\Theta}) \tag{1}$$

where $\boldsymbol{t_i}$ is one of the tokens of the sequence which tokenized by BPE on byte level and all of the tokens are bounded in a same size that was integrated by the question and the ground truth answer and $\boldsymbol{\Theta}$ is the learnable parameters from the model. We set m, the maximum token length, to 128.

Moreover, some questions just like “What do fibroids look like and can they be cured?” are directional or “I would like to ask what is atypical carcinoid and what is atypical carcinoid?” are divergent, so our model must be generative to deal with variety of questions. We roughly define model’s input and output like this:

$$\boldsymbol{A} = \boldsymbol{F}(\boldsymbol{Q} \mid \boldsymbol{P}) \tag{2}$$

where $\boldsymbol{F}$ represent our model and $\boldsymbol{A}$ is an output of the model which contain an answer for question and relative diagnosis and treatment plan. There is no doubt that asked question must without any article or literature because no one will bring a paper to see doctor and ask them to comprehend it then get treatment guidance. Of course, for medical QA, open-domain is the best destination which let the model to answer spurious questions without a given context. We suggest P, a priori experience of conditional probability, to be general medical knowledge absorbed from training dataset instead of being input requirement of joint probability like $\boldsymbol{A} = \boldsymbol{F}(\boldsymbol{Q}, \boldsymbol{P})$. Overall, simply put a question to the model and get an answer for responding, whose process of inference attribute to the model's experience originating in the training period.

3.2 Datasets

Our model is designed to solve Chinese cancers question answering so we used three different datasets Chinese medical dialogue data, cMedQA2 [28], HuaTuo [16], all of them are established by Chinese question and answers.

Chinese medical dialogue data. Chinese medical dialogue data which contains 1,145,231 consultancies from doctor and patient conversations. The total corpus is 3,959,333, 2,179,008 from doctors and 1,780,325 from patients, all of which came from the website and were reviewed by people.

cMedQA2. cMedQA2 [28], 108,00 questions and 203,569 answers in total, is a medium to large Chinese QA dataset. Zhang [28] et al create a new text corpus by harvesting questions and answers from an online Chinese health and wellness community in order to validate their network.

HuaTuo. HuaTuo [16] is a Chinese QA dataset which was integrated by four information sources, Distilled Instructions from ChatGPT, Real-world Instructions from Doctors, Distilled Conversations from ChatGPT and Real-world Conversations with Doctors.

3.3 CancersQA

QA system research plays a crucial role in the field of intelligent cancer diagnosis. In real-world scenarios, individual hospitals, possess low-resource and private medical data that is non-IId and exhibits diverse domain knowledge and data styles. This poses challenges due to limited data availability and computational complexity. To address these issues, we proposed CancersQA aiming to train a PTM-based question answering model under the multi-client setting. The framework enhances the utilization of privacy data through federated learning (FL) and PTM fine-tuning. Additionally, the WeiPro aggregation algorithm mitigates the adverse impact of different domains and styles, thereby improving model performance. CancersQA presents a promising solution for these challenges in real-world environments, offering enhanced privacy preservation and alleviating issues associated with diverse medical data.

We present a systematic process for federated learning, where $\boldsymbol{C} = \{\boldsymbol{C_1}, \boldsymbol{C_2}, \boldsymbol{C_3}, \dots, \boldsymbol{C_N}\}$ represents the set of clients, $\boldsymbol{D} = \{\boldsymbol{D_1}, \boldsymbol{D_2}, \boldsymbol{D_3}, \dots, \boldsymbol{D_N}\}$ represents the local dataset of each client from 1 to $\boldsymbol{N}$, and $\boldsymbol{T} = \{\boldsymbol{T_1}, \boldsymbol{T_2}, \boldsymbol{T_3}, \dots, \boldsymbol{T_N}\}$ represents the time required for each client to train for one local epoch.

In our approach, we select GPT-2 as the base model, which will be initialized on the central server $\boldsymbol{S}$. The central server then broadcasts the global model to each client $\boldsymbol{C_t} \in \boldsymbol{C}$. Each client proceeds to fine-tune the model synchronously using its local dataset and enters a waiting state for $\boldsymbol{T}^*$ until the last node has completed training, where $\boldsymbol{T}^*$ represents the maximum time required for training among all clients.

Once all clients have finished training, they collectively send back their updated parameters of the local model to the central server $\boldsymbol{S}$. The central server then employs the WeiPro aggregation algorithm to aggregate the local models and generate a new global model. This new global model is subsequently distributed back to each client for the next round of training.

By following this iterative process, our federated learning framework ensures efficient collaboration among clients while maintaining privacy and data security. The utilization of GPT-2 as the base model and the WeiPro aggregation algorithm enhance the overall performance and accuracy of the global model.

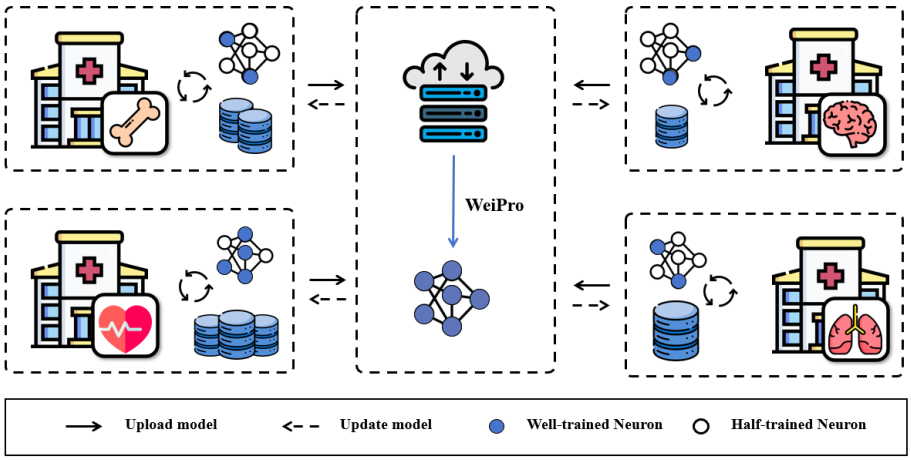

Fig 1. The figure shows the architecture of CancersQA which contains two parts, the central server and clients. The solid arrow represents sending the global model to individual clients, and dash arrow represent returning the update model to central server.

Fig 1. illustrates the architectural of our model, CancersQA. The diagram includes four clients, each representing a different medical specialty: Orthopaedics, Neurology, Cardiology, and Pulmonology. Each client holds a distinct set of data, varying in volume, as depicted by the pie charts associated with each hospital.

The functionality of the model in one global epoch operates as follows: firstly, a global model is initialized at the central server. This model is then downloaded by each client, resulting in each hospital having an identical local model. Subsequently, these hospitals individually fine-tune their respective local models using their unique datasets.

Given the variability in data, the resulting fine-tuned models from each hospital differ accordingly. In the diagram, blue neurons signify fully trained components, while white neurons represent partially trained ones. After the fine-tuning process, the central server consolidates the various models from each hospital. By employing the WeiPro algorithm, a new global model is aggregated and derived from these distinct models. The cycle then repeats, creating a dynamic, continually evolving model that constantly learns from a diverse set of medical data.

In this way, we show that the model can perform well in cancer question-answering tasks after fine-tuning on a small dataset, rather than training the entire model independently. CancersQA takes advantage of both large PTMs and federated learning, has a stable output, and can use fragmented non-IID data, either to ensure that different data features are learned or to protect patient privacy.

3.4 WeiPro Aggregating algorithm

The design of aggregation algorithms is one of the most important processes in federated learning frameworks that really infect the performance of the model. Inspired by Google, H. Brendan et al [17] which first bring the notion of federated learning, at the

same time, they proposed FedAvg, a useful aggregation algorithm, which is defined as follows:

$$\boldsymbol{w_{i+1}} = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\boldsymbol{j}}^{\boldsymbol{N}} \boldsymbol{w_{i+1}^{j}} \tag{3}$$

where $\boldsymbol{w_{i+1}}$ is the updated parameters of the whole model in the $\boldsymbol{i+1^{th}}$ iteration which is aggregate from local model parameters $\boldsymbol{w_{i+1}^{j}}$. This model considered all local participants and gave them equal weight, allowing the model to acquire knowledge from multiple sources. However, for non-IID data, such a simple algorithm may not work well. FedAvg only thinks about merging parameters from all nodes, but ignores other factors that really play an important role in model building such as data quantity, loss value, participation rounds, etc.

Reference point. In this paper, we proposed WeiPro inspired by [29], compared with federated averaging algorithm, WeiPro injects more sensible weights to balance model's ability to absorb knowledge. First, we remove $\boldsymbol{1/N}$, then add $\boldsymbol{\rho}$ to control the strength of the contribution of each model which called "balance factor", the formula is defined as:

$$\boldsymbol{\theta_i} = \sum_{\boldsymbol{k}}^{\boldsymbol{N}} \boldsymbol{\rho_i^k \theta_i^k}, \boldsymbol{st}. \sum_{\boldsymbol{k}}^{\boldsymbol{N}} \boldsymbol{\rho_i^k} = \mathbf{1} \ \boldsymbol{and}\ \boldsymbol{\rho_i^k} > \mathbf{0}; \boldsymbol{k\epsilon}\{\mathbf{0}, \mathbf{1}, \mathbf{2}, \dots, \boldsymbol{N}\} \tag{4}$$

where $\boldsymbol{\theta_i}$ is the updated parameters of the global model in the $\boldsymbol{i^{th}}$ iteration, $\boldsymbol{\theta_i^k}$ is the parameters updated by the local models, $\boldsymbol{\rho_i^k}$ is used to control $\boldsymbol{\theta_i^k}$. It's worth noting that the sum of all $\boldsymbol{\rho_i^k}$ are equal to 1 and all of them are positive. $\boldsymbol{\rho_i^k}$ takes into account several aspects that make it more convincing, such as the loss of the local model in the $\boldsymbol{i^{th}}$ iteration, the epoch numbers that the institution joins, and the quality of each local dataset. These factors will change our model's update tendency. This is the quantitative interpretation of $\boldsymbol{\rho_i^k}$:

$$\boldsymbol{\rho_i^k} = \frac{\boldsymbol{n_k \varphi(\xi_k, i)(\sigma_i^k(\theta))^{-\mu}}}{\sum_{\boldsymbol{j=1}}^{\boldsymbol{N}} \boldsymbol{n_j \varphi(\xi_j, i)(\sigma_i^j(\theta))^{-\mu}}}; \boldsymbol{j} = \mathbf{1}, \mathbf{2}, \mathbf{3}, \dots, \boldsymbol{N} \tag{5}$$

$$\boldsymbol{\sigma_i^t(\theta)} = \frac{\boldsymbol{C_t}}{\boldsymbol{X_t}}; \boldsymbol{t} = \mathbf{1}, \mathbf{2}, \mathbf{3}, \dots, \boldsymbol{N} \tag{6}$$

where $\boldsymbol{n_k}$ represents the number of GPUs in the $\boldsymbol{k^{th}}$ node. $\boldsymbol{\sigma_i^t(\theta)}$ is a transfer function defined as (6), $\boldsymbol{X_t}$ is the set of local validation dataset, while $\boldsymbol{C_t}$ denotes, in the $\boldsymbol{i^{th}}$ iteration, the set of outputs of the local model whose similarity exceeds a certain threshold after a local epoch. $\boldsymbol{\mu}$ is a hyper parameter which set before training step. Obviously, $\boldsymbol{\sigma_i^t(\theta)}$ is like the accuracy of the local model, the higher $\boldsymbol{\sigma_i^t(\theta)}$ the lower $(\boldsymbol{\sigma_i^k(\theta)})^{-\boldsymbol{\mu}}$, so the global model tends to lean toward bad performance parameters and gives more chance and intensity to fit its data set, which leads to the effect of mutual control and joint decision-making. There is a simple way to alternatively remove the exponent $-\boldsymbol{\mu}$ and use the local loss to compute $\boldsymbol{\sigma_i^t(\theta)}$, the purpose of this is to let the global model take more care of the underfitting model.

In addition, except data's own properties, the influence of external factors is also crucial. In the training period of the Federation, the departure of members does not affect the training process, whether they leave of their own accord or are disconnected

from the Internet due to unknown mistakes, so we must use a function to measure it. At each training epoch, all institutions send another value $\boldsymbol{\delta_i^k}$ to the central server, which is a BOOL value representing whether they have joined the $\boldsymbol{i^{th}}$ iteration, with only two possible values 0 and 1 representing online and offline respectively. Then, the central server needs to reassemble the received parameters into K sets in $\boldsymbol{i^{th}}$ iteration, $\boldsymbol{\xi_1, \xi_2, \xi_3, \dots, \xi_N}$, which show the details as follow:

$$\boldsymbol{\xi_k = \left[\delta_1^k, \delta_2^k, \delta_3^k, \dots, \delta_i^k\right], st. \delta_i^k \in \{0, 1\}; k = 1, 2, 3, \dots, N} \tag{7}$$

where $\boldsymbol{\xi_k}$ has $\boldsymbol{i}$ values. Another process is to calculate the number of epochs in which each node has participated in training up to $\boldsymbol{i^{th}}$ iteration. $\boldsymbol{\varphi(\xi_k, i)}$ is a function to compute the sum of previous $\boldsymbol{i}$ values:

$$\boldsymbol{\varphi(\xi_k, i) = \delta_1^k + \delta_2^k + \delta_3^k + \cdots + \delta_i^k; k = 1, 2, 3, \dots, N} \tag{8}$$

where $\boldsymbol{i}$ is current iteration round, so $\boldsymbol{\varphi(\xi_k, i)}$ is the epoch number that $\boldsymbol{k^{th}}$ node have joined. So, the node has high number of participations will get more attention.

Projection mapping. Now, we got the $\boldsymbol{\theta_i}$, however, it is clearly not profound enough to take this parameter as the global updated model. We need to consider model distributions in lower dimension. By flattening $\boldsymbol{\theta_i}$, $\boldsymbol{\theta_i^k}$ and original $\boldsymbol{\theta_0}$ (the global parameters before sending to the local nodes), we will get three vectors $\boldsymbol{v^i \in \mathbb{R}^d}$, $\boldsymbol{v^0 \in \mathbb{R}^d}$ and $\boldsymbol{v_i^k \in \mathbb{R}^d}$ where $\boldsymbol{d}$ is the number of total parameters. $\boldsymbol{v^i - v^0}$ gives a basic updated orientation which called the reference vector and $\boldsymbol{v_i^k - v^0}$ which represents the local updated orientation of the $\boldsymbol{k^{th}}$ node in the $\boldsymbol{i^{th}}$ iteration which called practice vector, then we compute the projection of the practice vector onto the reference vector. This projection value expresses the degree of similarity between the updated direction of each node and the reference direction:

$$\boldsymbol{p_k = \frac{(v_i^k - v^0) \cdot (v^i - v^0)}{\|v^i - v^0\|}; k = 1, 2, 3, \dots, N} \tag{9}$$

where $\boldsymbol{v^i - v^0}$ is the refence vector and $\boldsymbol{v_i^k - v^0}$ is the practice vector. For each node, in $\boldsymbol{i^{th}}$ iteration, central server will get a projection set like $\boldsymbol{\{p_1, p_2, p_3, \dots, p_N\}}$. These projections will be sent to a linear transformer block and get the output $\boldsymbol{\{z_1, z_2, z_3, \dots, z_N\}}$. This block can be learnt or be frozen. Another choice is to calculate the projection's length $\boldsymbol{\|p_k\|}$ to instead $\boldsymbol{z_k}$ and define $\boldsymbol{\alpha_k}$ as follows:

$$\boldsymbol{\alpha_k = \frac{(z_k)^\lambda}{\sum_{j=1}^N (z_j)^\lambda}; k = 1, 2, 3, \dots, N} \tag{10}$$

where $\boldsymbol{\lambda}$ is the hyper parameter and $\boldsymbol{\alpha_k}$ called incremental parameter. The last global parameter is $\boldsymbol{\theta_0 + \sum_k^N \alpha_k \cdot \theta_i^k}$. All of these processes are in Algorithm 1.

Algorithm 1 WeiPro

Require: The number of total clients *T*; The number of selected clients *N*; The number of total rounds *I*; The number of each client local epoch E; Dataloader length B; The current status of training participation ξ; Learning rate η.

Server:
Initialize θ_0, ξ.
for round i in $[1, 2, 3, \dots, I-1]$ **do**
    $C' \leftarrow$ Select N clients $[c_1, c_2, c_3, \dots, c_N]$ from clients set C randomly.
    Add 0 to each $\xi_k; k = 1, 2, 3, \dots, N$.
    Send the initialized model θ_0 to N clients.
    for all clients $k \in C'$ **do in parallel**
        $\theta_i^k, loss_i^k \leftarrow$ LocalUpdate(θ_0): Each client updates their own local model.
        $\xi_k[-1] \leftarrow 1$: k^{th} node has participant the i^{th} iteration.
    end for
    $\omega_k \leftarrow loss_k / \sum_j^N loss_j$: Compute the ω_k for k^{th} client.
    $\theta_i \leftarrow \sum_k^N \omega_k \cdot \theta_i^k$: Basic updated orientation.
    $\alpha^i \leftarrow$ ProjectionSimilarity$(\theta_0, \theta_i, \theta_i^k)$: incremental parameter.
    $\theta_0 \leftarrow \theta_0 + \sum_k^N \alpha_k \cdot \theta_i^k$: Update θ_0 for $i + 1^{th}$ iteration.
end for

ProjectionSimilarity$(\boldsymbol{\theta_0}, \boldsymbol{\theta_i}, \boldsymbol{\theta_i^k})$:
$v^0, v^i, v_i^k \leftarrow$ Flatten $\theta_0, \theta_i, \theta_i^k$.
$p_k \leftarrow$ Use formula (9) to get the projection p_k.
Return α^i by executing formula (10).

Client k:
LocalUpdate$(\boldsymbol{\theta_0})$:
$\theta_i^k \leftarrow \theta_0; loss_k \leftarrow 0$.
for local epoch e in E **do**
    for each batch $b \in B$ **do**
        Forward propagation ($loss_b$) and back propagation ($\nabla\theta_{i,b}^k$)
        $\theta_i^k \leftarrow \theta_i^k + \eta \cdot \nabla\theta_{i,b}^k$: Update the local model.
        $loss_k \leftarrow loss_k + loss_b$.
    end for
end for
Return $\boldsymbol{\theta_i^k}, \boldsymbol{loss_k}$

4 Results

We evaluate our method on three public datasets, Chinese medical dialogue data, cMedQA2 and HuaTuo. All of these datasets have samples consisting of a question-answer pair. After selecting cancer samples, we got three small cancer data and what we do as follows.

Our method is evaluated on three public datasets, including Chinese medical dialogue data, cMedQA2, and HuaTuo. Each dataset consists of multiple question-answer pairs. After selecting cancer-related samples, we managed to extract three cancer datasets. The subsequent procedures undertaken are outlined below.

4.1 Metrics

METEOR. CancersQA is designed for open-domain question answering, which can be treated as text generation task. METEOR [30] is employed to evaluate the performance of the QA model, which is defined as follows:

$$\boldsymbol{\alpha_k = \frac{(z_k)^{\lambda}}{\sum_{j=1}^{N}(z_j)^{\lambda}}; k = 1, 2, 3, \dots, N} \tag{11}$$

$$\boldsymbol{Pt = \gamma \cdot frag^{\beta}} \tag{12}$$

$$\boldsymbol{frag = ch/m} \tag{13}$$

$$\boldsymbol{score = (1 - Pt) \cdot S_{base}} \tag{14}$$

where $\boldsymbol{P}$ is the precision score and $\boldsymbol{R}$ is the recall score. $\boldsymbol{Pt}$ is the penalty for n-grams matches. $\boldsymbol{ch}$ is the number of successfully matches, while $\boldsymbol{m}$ is the length of the candidate sentence. $\boldsymbol{\alpha}$, $\boldsymbol{\gamma}$ and $\boldsymbol{\beta}$ is the hyper parameter which always set 3, 3 and 0.5 respectively. METEOR score can be computed by formula (14). METEOR calculates the degree of similarity in the WN synonymy way (Match strictness from highest to lowest).

ROUGE. Another popular metric for text generation tasks is ROUGE [31], we leverage ROUGE with METEOR to evaluate our method systematically. The difference between ROUGE-N and ROUGE-L is the length in grams when matching the reference answers and the candidate answer. ROUGE-N analyzes n-gram segments, while ROUGE-L retrieves the longest common subsequence. These are all defined as follows:

$$\boldsymbol{ROUGE - N = \frac{\sum_{A \in ReferenceAnswers} \sum_{gram_n \in A} Count_{match}(gram_n)}{\sum_{A \in ReferenceAnswers} \sum_{gram_n \in A} Count(gram_n)}} \tag{15}$$

$$\boldsymbol{p_{lcs} = \frac{LCS(X,Y)}{n}} \tag{16}$$

where $\boldsymbol{A}$ is one of the answers in the reference answers, $\boldsymbol{Count(gram_n)}$ is the number of $\boldsymbol{gram_n}$ in $\boldsymbol{S}$, and $\boldsymbol{Count_{match}(gram_n)}$ is the number of grams is matched. $\boldsymbol{LCS(X,Y)}$ is the longest common subsequence and $\boldsymbol{n}$ is the length of the generated answer.

4.2 Experiment

In this section, we present the details of the experiments including the experimental device, the parameter setting, and the deployment methods. Then, this section provides an in-depth analysis and discussion of the experimental results. For federated learning, the number of total epochs is set to 100 epochs and the number of local epochs is set to 1. We use the AdamW optimizer for mini-batch size 4 and update the parameters every 4 batches. The model is trained for up to $\boldsymbol{2.5 \times 10^4}$ iterations and accumulates approximately $\boldsymbol{3.2 \times 10^3}$ roughly. The learning rate starts at $\boldsymbol{2e^{-5}}$ and decreases by $\boldsymbol{2e^{-7}}$ with each iteration. For WeiPro, $\boldsymbol{\lambda}$ is set to 1. We use 80% of the

dataset as our training dataset for training, 10% for valid and 10% for test. Both of them are trained on NVIDIA 4090 with 3 GPUs, 24GB.

CancersQA and Other Methods Comparison Experiment. We compare the impact of the three training methods, centralized, fragmented and federated, on model performance. In centralized data training, the data is collected and randomly shuffled. One of the clients is designated as the central server, on which the model is trained with the centralized datasets. In fragmented training, data are stored secretly and the model will be trained on their local dataset separately. In federated learning way, three clients are trained on the separate datasets by exchanging the updated parameters without sharing the data, simulating multiple clients such as hospitals or healthcare institutions. The details are shown in Table 1.

Table 1. METEOR and ROUGE-L score from different methods of two different experiments.

Method	Size	Convergence epoch	METEOR	ROUGE-L
VS Centralized data sharing				
Centralized data sharing	148.5M	85	20.56	21.43
CancersQA	**148.5M**	**50**	**20.33**	**21.21**
VS Fragment data training				
Client1	45.6M	26	5.24	11.80
Client2	5.5M	5	4.90	11.39
Client3	138.3M	75	15.29	17.88
CancersQA*	**189.4M**	**66**	**21.40**	**21.36**

* Use all three clients' data for training, but do not mix them. CancersQA* has more data than CancersQA.

Cancer samples are extracted from three datasets and assigned to three different clients representing different hospitals in the real world. The initial total number of epochs is set to 100, and it ends when they converge by calculating the last loss value minus the current loss value is not higher than the threshold $\mathbf{1}e^{-5}$. There are 45.6M data in the first client, consisting of short question pairs, 5.5M data in the second client, also consisting of short question pairs, and 138.3M data in the third client, consisting of long question pairs. Different dataset represents three levels (low, medium and high) of the hospitals.

The Part I in Table 2 shows more details for the comparison between CancersQA and Centralized Data Sharing method. When equal amount of data is trained with the centralized data sharing method and our method, respectively, CancersQA converges faster by only 50 epochs 35 epochs less than centralized data sharing and reaches the similar score with centralized data sharing method as Table 2 illustrating which is 20.33 in METEOR and 21.21 in ROUGE-L. It's worth noting that all the data are on cancers selected from the original dataset.

Part II in Table 2 shows four identical models with different data and training mode, their number of epochs to converge and their METEOR score. We decentralize and deploy Chinese-medical-dialogue-data, cMedQA2 and HuaTuo to Client 1-3, respectively. It is obvious that CancersQA* outperforms the other clients, the METEOR is 16.16 points higher than Client1, 16.50 points higher than Client2 and 6.11 points higher than Client3. For ROUGE-L score, our method has the highest score. It can be seen that the model has better performance in the combination of federated learning and pre-training because we extend the data volume by integrated three datasets. In CancersQA*, the model converges in 66 epochs which is also lower than centralized data sharing but the dataset is larger than the former.

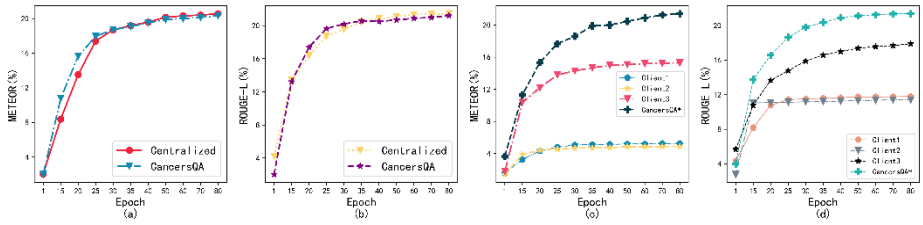

Fig 2. The figure shows the comparison of METEOR scores and ROUGE scores of different methods as the number of iterations increases during the training process, and it is clear that CancersQA not only converges quickly but also scores high and relatively stable.

We conducted experiments where we saved the models at different epochs and analyzed the changes in their METEOR score and ROUGE score. Our approach, CancersQA, exhibits remarkable similarity to the Centralized method, with a score difference of less than 0.5% after 80 epochs. However, CancersQA achieves convergence at a faster rate, requiring fewer epochs to reach the same score (See Fig. 2. (a) (b)).

Upon analyzing the impact of dataset size, we observed that the CancersQA score improves consistently while maintaining stable convergence. On the other hand, Clients 1-3 converge quickly but exhibit inconsistencies in curve smoothness. Additionally, the overfitting phenomenon is observed, leading to poor generalization and lower final scores for Clients1-3 (See Fig 2. (c) (d)).

In conclusion, our proposed method effectively utilizes data silos and ensures the stability of the model. The results demonstrate that CancersQA outperforms the Centralized method in terms of convergence speed, score improvement with dataset growth, and avoidance of overfitting.

Figure 3 shows the training time per unit of data up to the current epochs within 20 to 45 epochs, from which it is intuitively clear that the Centralized data training time to reach the same epochs is the longest, and CancersQA is almost 1/3 of it. The total time for Client1 and Client2, Client3 are not as long, but when compared to the amount of data processed per unit of time, CancersQA is only about half of them compared to the amount of data processed. From Figure 2 and Table 2, the results demonstrate that CancersQA outperforms other systems by saving half of the training

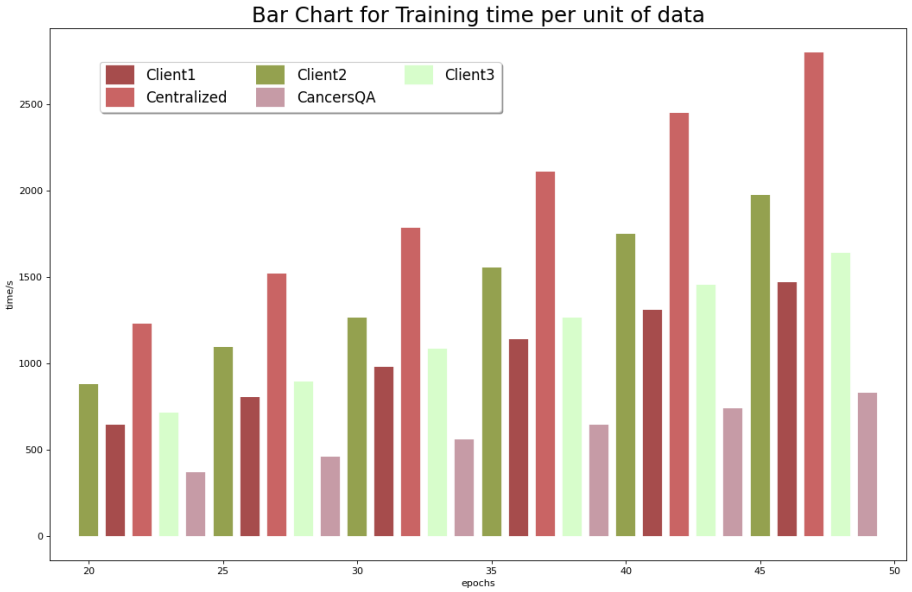

Fig. 3. The provided figure illustrates the training time for different individuals, spanning from 20 epochs to 45 epochs. In the case of client1, client2, and client3, a local model is efficiently fine-tuned on respective local datasets. Additionally, both centralized data sharing and the CancersQA approach have been performed using equal amounts of data.

time with a performance drop of only 0.53%, which validates the high stability and efficiency of our method CancersQA.

Aggregating Algorithm Comparison Experiment. In the last section, we show that our method is more suitable than other training approaches in the cancer question answering task due to our specific federated learning framework. In the aggregation algorithm comparison experiment, we compare CancersQA with other federated learning methods and carefully discuss the performance of the model.

In our comparison experiment, three baseline systems, FedAvg [17], FedProx [32] and WeiAvg (Entropy) [29], are selected as follows:

- **FedAvg:** Add the updated models of the three clients together and divide by the number of clients to get a new global model to replace the original global model.
- **FedProx:** An additional constraint on the distance between the client's local model and the global model is added after the traditional federated learning experience loss optimization function, so that the final updated local model cannot deviate too much from the original global model.
- **WeiAvg (Entropy):** Generate coefficients belonging to each client model by calculating the fine-tuning loss of each client's local model.
- **WeiPro:** The client's fine-tuning loss, data distribution, hardware state and number of global epochs involved jointly determine the model coefficients and derive the reference vector for the global model update direction, and

then calculate the projection of each client's practice update vector onto the reference vector to obtain each client's final model coefficients.

In common situations, especially in hospital, QA system suffers from the limitation of the data issue that they are the non-independent and identically distributed (non-IID). To simulate this issue, we select part of our dataset and conduct our experiment as follows:

- $\boldsymbol{D_1}$ **:** 0.006 times Chinese-medical-dialogue-data for Client1, 0.1 times cMedQA2 for Client2 and 0.001 times HuaTuo for Client3.
- $\boldsymbol{D_2}$**:** Mixing the above datasets and randomly shuffled then divided equally among three clients.
- $\boldsymbol{R_x}$: Keeping the $\boldsymbol{D_2}$ dataset unchanged, randomly select one or two nodes and deactivate them from participating in the current epoch.

Table 2. METEOR and ROUGE-L score from different methods of two different experiments.

Method	$\boldsymbol{D_1}$	$\boldsymbol{D_2}$	$\boldsymbol{R_x}$
	Metrics: METEOR/ROUGE-L		
FedAvg	16.72/17.54	19.88/20.90	16.46/17.11
FedProx	14.60/17.89	18.67/20.13	13.61/15.77
WeiAvg (Entropy)	18.23/19.56	19.11/20.47	16.91/18.04
WeiPro	**19.61/20.74**	**19.99/21.26**	**19.40/20.04**

As Table 4 shows, FedAvg works well when the data are independent and identically distributed, achieving a relatively high METEOR score of 19.88 compared to other approaches but the ROUGE-L score is a bit lower than ours. However, in $\boldsymbol{D_1}$, data are non-IID, the FedAvg aggregation approach exposes significant shortcomings, in the meantime, other methods' scores all decrease. Despite FedProx is designed for non-IID data, when facing the gap of cancer data in each hospital, FedProx perform worse only 14.60 and 17.89 points which is 4.07 and 2.24 points lower than $\boldsymbol{D_2}$. In contrast, our method remains stable, reducing only 0.38 and 0.52 points. Furthermore, in order to prove the superiority of our method, we randomly select one or two clients to deactivate in each epoch, the datasets we defined as $\boldsymbol{R_x}$, and the results are recorded in the last column. As a result, our method is also stable, the METEOR score and the ROUGE-L score is 19.40 and 20.04 respectively while other methods show fluctuations when meeting random uncertainties.

5 Discussion

When comparing CancersQA's model performance with the centralized data sharing training approach, there is only a slight degradation of 0.23 points. This is particularly relevant for cancer data, which is often more private and sensitive. Additionally, our proposed WeiPro method outperforms other aggregation algorithms, namely FedAvg, FedProx, and WeiAvg (Entropy). Specifically, when the data is not independently and

identically distributed, WeiPro demonstrates superiority by 2.89/3.20, 5.01/2.85, and 1.38/1.18 percentage points compared to FedAvg, FedProx, and WeiAvg (Entropy), respectively.

Heterogeneity is commonly observed in datasets within hospitals. FedAvg's simplistic aggregation approach through averaging fails to adequately address this issue. By assigning equal weights to all hospitals, unique attributes and local conditions held by each dataset are disregarded, which is deemed unreasonable. Consequently, the FedAvg algorithm exhibits significant fluctuations when substantial changes in the data occur. Additionally, the performance of the model trained using the FedAvg algorithm is greatly impacted by non-IID data, as demonstrated by the similarity between the FedAvg scores on $\boldsymbol{R_x}$ and $\boldsymbol{D_1}$.

FedProx addresses this deviation issue by incorporating a constraint on the distance between the updated local model and the global model. However, this alteration in the model update route and the change in convergence direction adversely affect the model's performance. WeiAvg (Entropy) calculates weights based on the loss values of each epoch at each node, but solely considering data loss lacks comprehensiveness. Both FedProx and WeiAvg (Entropy) neglect the importance of federated learning client dominance and fail to incorporate the consideration of the hospitals' involvement in the aggregation algorithms over epochs, resulting in lower performance in $\boldsymbol{R_x}$ compared to $\boldsymbol{D_1}$ and $\boldsymbol{D_2}$.

In contrast, our WeiPro method comprehensively considers all relevant factors. It calculates a common reference vector for all hospital models, accounting for data distribution, losses, and round participation. Without altering the update direction of the original local models, the weight coefficients are derived by projecting the local models onto the reference direction. Higher weight values indicate more significant deviations and a higher probability of optimization. This allows the central node to dynamically weigh the update intensity of each local model at any given time. Notably, our method demonstrates relative stability in $\boldsymbol{D_1}$, $\boldsymbol{D_2}$, and $\boldsymbol{R_x}$, while other methods exhibit significant fluctuations.

In summary, our proposed WeiPro method achieves comparable performance to the centralized data sharing approach with only a slight degradation in model performance. It outperforms other aggregation algorithms by considering the heterogeneity of hospital datasets and incorporating various factors.

6 Conclusions

We propose a CancersQA approach based on the realistic data feature for intelligent cancer diagnosis, which incorporates federated learning and pre-trained models. In the CancersQA, the models are fine-tuned on the individual hospital data locally to protect the data privacy. The weights are shared and aggregated through WeiPro algorithm through a central node. CancersQA enables multiple hospitals to jointly fine-tune LLMs, and enhances the utilization of cancer data, which is highly promising for applications. In addition, the weight aggregation method WeiPro outperforms existing

baselines in the comparison experiment. The results show that WeiPro has strong adaptability with comprehensive consideration.

For most cases, the number of participating organizations is very large and the data distribution is much more heterogeneous. Due to arithmetic reasons this paper only performs feature extraction to approximate real entities by building a small equivalence framework. It is also worth exploring that the communication overhead of federated learning is often much larger than that of model training, so how to reduce the communication overhead is the key to improve the efficiency of federated learning. In addition, for malicious attacks, the central node needs a unique analysis scheme to deal with them, which we will continue to follow up in the future.

Funding: This research was funded in part by the National Key Research and Development Program of China under Grant 2020YFB1406702-3, and in part by the National Natural Science Foundation of China under Grant 62006257 and 61772575.

References

1. Skianis K.; Briand Y.; Desgrippes F. Evaluation of machine translation methods applied to medical terminologies. Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis 2020, 59-69.
2. Corral A.; Saralegi X. Elhuyar submission to the Biomedical Translation Task 2020 on terminology and abstracts translation. Proceedings of the Fifth Conference on Machine Translation 2020, 813-819.
3. Jauregi Unanue I.; Piccardi M. Pretrained language models and backtranslation for English-basque biomedical neural machine translation. Fifth Conference on Machine Translation (WMT20). The Association for Computational Linguistics 2020.
4. Li D.; Hu B.; Chen Q.; et al. Towards medical machine reading comprehension with structural knowledge and plain text. Proceedings of the 2020 conference on empirical methods in natural language processing (EMNLP) 2020, 1427-1438.
5. Abacha A.B.; Zweigenbaum P. MEANS: A medical question-answering system combining NLP techniques and semantic Web technologies. Information processing & management 2015, 51(5), 570-594.
6. Singhal K.; Tu T.; Gottweis J.; et al. Towards expert-level medical question answering with large language models. arXiv preprint arXiv:2305.09617 2023.
7. Robinson J.; Rytting C.M.; Wingate D. Leveraging large language models for multiple choice question answering. arXiv preprint arXiv:2210.12353 2022.
8. Wang S.; Zhou W.; Yang Y.; et al. Adapting Pre-Trained Visual and Language Models for Medical Image Question Answering. CLEF2023 Working Notes, CEUR Workshop Proceedings, CEUR-WS. org, Thessaloniki, Greece 2023.
9. Patel D.; Konam S.; Selvaraj S.P. Weakly supervised medication regimen extraction from medical conversations. arXiv preprint arXiv:2010.05317 2020.
10. Joshi A.; Katariya N.; Amatriain X.; et al. Dr. summarize: Global summarization of medical dialogue by exploiting local structures. arXiv preprint arXiv:2009.08666 2020.
11. Wang X D.; Weber L.; Leser U. Biomedical event extraction as multi-turn question answering. Proceedings of the 11th International Workshop on Health Text Mining and Information Analysis 2020, 88-96.
12. Lee M.; Cimino J.; Zhu H.R.; et al. Beyond information retrieval—medical question answering. AMIA annual symposium proceedings. American Medical Informatics Association 2006, 2006, 469.
13. Mutabazi E.; Ni J.; Tang G.; et al. A review on medical textual question answering systems based on deep learning approaches. Applied Sciences 2021, 11(12), 5456.

14. Raza S.; Schwartz B.; Rosella L.C. CoQUAD: a COVID-19 question answering dataset system, facilitating research, benchmarking, and practice. BMC bioinformatics 2022, 23(1), 1-28.
15. Singhal K.; Azizi S.; Tu T.; et al. Large language models encode clinical knowledge. arXiv preprint arXiv:2212.13138 2022.
16. Wang H.; Liu C.; Xi N.; et al. Huatuo: Tuning llama model with chinese medical knowledge. arXiv preprint arXiv:2304.06975 2023.
17. McMahan B.; Moore E.; Ramage D.; et al. Communication-efficient learning of deep networks from decentralized data. Artificial intelligence and statistics. PMLR 2017, 1273-1282.
18. Konečný J.; McMahan B.; Ramage D. Federated optimization: Distributed optimization beyond the datacenter. arXiv preprint arXiv:1511.03575 2015.
19. Smith V.; Forte S.; Chenxin M.; et al. CoCoA: A general framework for communication-efficient distributed optimization. Journal of Machine Learning Research 2018, 18, 230.
20. Tan Y.; Long G.; Ma J.; et al. Federated learning from pre-trained models: A contrastive learning approach. Advances in Neural Information Processing Systems 2022, 35, 19332-19344.
21. Tian Y.; Wan Y.; Lyu L.; et al. FedBERT: When federated learning meets pre-training. ACM Transactions on Intelligent Systems and Technology (TIST) 2022, 13(4), 1-26.
22. He K.; Zhang X.; Ren S.; et al. Deep residual learning for image recognition. Proceedings of the IEEE conference on computer vision and pattern recognition 2016, 770-778.
23. Qiu X.; Sun T.; Xu Y.; et al. Pre-trained models for natural language processing: A survey. Science China Technological Sciences 2020, 63(10), 1872-1897.
24. Xu D., Yen I.E.H.; Zhao J.; et al. Rethinking Network Pruning--under the Pre-train and Fine-tune Paradigm. arXiv preprint arXiv:2104.08682 2021.
25. Martinez J.; Shewakramani J.; Liu T W.; et al. Permute, quantize, and fine-tune: Efficient compression of neural networks. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2021, 15699-15708.
26. Liu Y.; Fine-tune BERT for extractive summarization. arXiv preprint arXiv:1903.10318 2019.
27. Hu E.J.; Shen Y.; Wallis P.; et al. Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685 2021.
28. Zhang S.; Zhang X.; Wang H.; et al. Chinese medical question answer matching using end-to-end character-level multi-scale CNNs. Applied Sciences 2017, 7(8), 767.
29. Dong F.; Abbasi A.; Drew S.; et al. WeiAvg: Federated Learning Model Aggregation Promoting Data Diversity. arXiv preprint arXiv:2305.16351 2023.
30. Banerjee S.; Lavie A. METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization 2005, 65-72.
31. Lin C.Y. Rouge: A package for automatic evaluation of summaries. Text summarization branches out 2004, 74-81.
32. Li T.; Sahu A.K.; Zaheer M.; et al. Federated optimization in heterogeneous networks. Proceedings of Machine learning and systems 2020, 2, 429-450.
33. Sheller M.J.; Edwards B.; Reina G A.; et al. Federated learning in medicine: facilitating multi-institutional collaborations without sharing patient data. Scientific reports 2020, 10(1), 12598.
34. Li J.; Zhang Z.; Zhao H.; Self-prompting large language models for open-domain qa. arXiv preprint arXiv:2212.08635 2022.

35. Hermann K.M.; Kocisky T.; Grefenstette E.; et al. Teaching machines to read and comprehend. Advances in neural information processing systems 2015, 28.
36. Xiong W.; Du J.; Wang W.Y.; et al. Pretrained encyclopedia: Weakly supervised knowledge-pretrained language model. arXiv preprint arXiv:1912.09637 2019.
37. Keya M.; Masum A.K.M.; Majumdar B.; et al. Bengali question answering system using seq2seq learning based on general knowledge dataset. 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT). IEEE 2020, 1-6.
38. Chandra Y.W.; Suyanto S. Indonesian chatbot of university admission using a question answering system based on sequence-to-sequence model. Procedia Computer Science 2019, 157, 367-374.
39. Zaib M.; Zhang W.E.; Sheng Q.Z.; et al. Conversational question answering: A survey. Knowledge and Information Systems 2022, 64(12), 3151-3195.
40. Lin C.Y.; Rouge: A package for automatic evaluation of summaries. Text summarization branches out 2004, 74-81.
41. Nguyen D.C.; Pham Q.V.; Pathirana P.N.; et al. Federated learning for smart healthcare: A survey. ACM Computing Surveys (CSUR) 2022, 55(3): 1-37.