UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Distributed Open-Domain Answer Sentence Selection by Federated Learning

by

Weikuan Wang

A THESIS SUBMITTED
IN FULFILLMENT OF THE
REQUIREMENTS FOR THE DEGREE

Master by Research
Under the supervision of
A/Prof. Guodong Long
Dr. Tao Shen
Dr. Jing Jiang

Sydney, Australia

2023

Certificate of Original Authorship

I, Weikuan Wang, declare that this thesis, is submitted in fulfilment of the requirements for the award of Master by Research, in the School of Computer Science, Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program.

Signature: Production Note: Signature removed prior to publication.

Date:2024.3.4

ABSTRACT

Distributed Open-Domain Answer Sentence Selection by Federated Learning

by

Weikuan Wang

Natural Language Processing (NLP) has achieved huge success, largely attributed to the use of large pre-trained language models. Open-Domain Question Answering (OD-QA), a task of significant importance within the industry, has also experienced substantial advancements through the application of these large-scale pre-training models. A specialized subset of Open-Domain Question Answering, Open-Domain Answer Sentence Selection (OD-AS2), seeks to provide an answer to a query from a sentence within a document collection. An excellent application of this technology is the deployment of OD-AS2 models on edge devices such as computers and smartphones, thereby creating a personalized, intelligent question-answering assistant derived from a user's personal documents.

Recently, Dense Retrieval has garnered interest from both academic and industrial society as a novel approach to OD-QA/OD-AS2. The Dense Retrieval models play an indispensable role by striking a balance between efficiency and performance across various solution paradigms. However, their effectiveness largely depends on the availability of ample labeled positive QA pairs and a diverse range of hard negative samples in training. Fulfilling these requirements is challenging in a privacy-preserving distributed scenario, where each client possesses fewer in-domain pairs and a relatively small collection, unsuitable for effective Dense Retrieval training.

To address this issue, we introduce a new deep-learning framework for Privacy-preserving Distributed OD-AS2, dubbed as PDD-AS2. Drawing upon the principles of Federated Learning, this framework incorporates a client-customized query encoding method for personalization and a cross-client negative sampling method to enhance learning effec-

tiveness called Fed-Negative. To assess our learning framework, we initially construct a novel OD-AS2 dataset, termed Fed-NewsQA, utilizing NewsQA as the base to simulate distributed clients with varying genre/domain data. Experimental results indicate that our learning framework outperforms baseline models and demonstrates impressive personalization capabilities.

Dissertation directed by A/Prof. Guodong Long, Dr. Tao Shen and Dr. Jing Jiang
Australian Artificial Intelligence Institute
Faculty of Engineering and IT
University of Technology Sydney

Acknowledgements

The completion of my Master by Research would not have been achievable without the inspiration and encouragement received from many people.

Firstly, I want to talk about my supervisors, Associate Professor Guodong Long, Dr. Tao Shen, and Dr. Jing Jiang. Associate Professor Guodong Long, serving as my principle supervisor, has consistently offered the guidance I required. He is a generous, patient, and tolerant person. I am glad to be his student. Dr. Tao Shen is the one who patiently guided me through the details of my scientific research process. From grammar and formatting errors in the paper, to the fine details of writing experimental code, and presenting results, he took time out of his busy work schedule to patiently answer these questions for me, sometimes into the late night. I still remember a few times when I made some basic mistakes that upset him, and I want to take this opportunity to apologize to him again. Dr. Jing Jiang has provided me with a great deal of assistance in the submission and publication of my paper. After I initially struggled with my submission, it was she who suggested a list of other conferences I could try. Ultimately, my paper was successfully published in one of the conferences she recommended.

I want to extend special thanks to my friend, Dr. Zhuowei Wang, who has resolved many of my doubts in both life and academia. In times of low spirits, he has also patiently advised and accompanied me.

I also want to particularly thank my parents, Bing Wang and Jianwei Shi. Without their support and companionship, I could not have embarked on this journey. Words can hardly express my gratitude to them, and these few lines cannot adequately describe my feelings for them.

Weikuan Wang

Sydney, Australia, 2023.

List of Publications

Conference Papers

C-1. **Weikuan Wang**, Tao Shen, Michael Blumenstein and Guodong Long. "Improving Open-Domain Answer Sentence Selection by Distributed Clients with Privacy Preservation," Advanced Data Mining and Applications, 2023. (Accepted on 15th June, 2023)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

The recent success of deep-learning [1] can be attributed primarily to two factors: the rapid development of computational power, and the discovery and utilization of massive amounts of data. Natural Language Processing, a field regarded as the jewel in the crown of artificial intelligence, has also greatly benefited from the development of deep learning. In Natural Language Processing (NLP), tasks are complex due to the nuanced and variable nature of human language, which involves diverse syntax, semantics, and context. This complexity surpasses that of fields like computer vision and robotics, which often deal with more structured and predictable data. They not only require the model to have a certain level of knowledge about the real world, but also high-level demands on the model's advanced cognitive activities such as reasoning and induction. However, in recent years, a number of deep learning models, such as BERT [2], have achieved, and even surpassed, human-level performance in many tasks in Natural Language Processing.

Open-Domain Question Answering (OD-QA) [3–6] is widely studied in academia and industry due to its extremely high commercial value. OD-QA is capable of answering user queries by searching millions of documents. Therefore, search engines utilize such technologies to make their search results more intelligent. As shown in this image, modern commercial search engines improve their search results through OD-QA. Compared to normal search results, the OD-QA returned content is more intuitive and specific, eliminating the need for users to click into web pages to find answers themselves. Additionally, some mobile or computer operating systems also carry such technologies to

help users intelligently search the documents stored on their devices. Typically, the task form varies slightly depending on the granularity of the answer. Some tasks only require returning the article where the answer is located, while others require accurately returning the phrase or word that constitutes the answer. **Open-Domain Answer Sentence Selection**(OD-AS2) [7–11], as a subtask of OD-QA, has achieved a balance in task granularity and answer accuracy by responding in the form of the sentence where the answer is located. Therefore, this subtask has also attracted much attention in the industry. This thesis will focus on answer sentence selection. In traditional OD-QA [12–14] methods, a con-

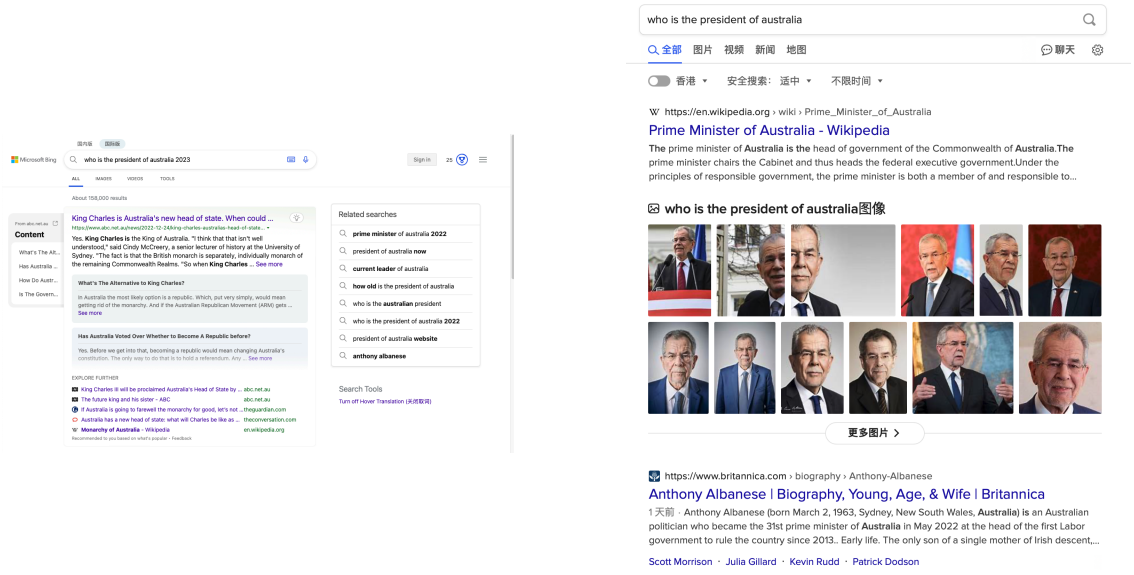

Figure 1.1 : Comparison between modern commercial search engines using OD-QA technology and traditional search engines. Modern search engines (left) return direct answers, while traditional search engines (right) provide only related page links.

text retriever is responsible for extracting documents related to the question from a lot of documents. During this process, some traditional encoding methods, such as BM25 [15] and TF-IDF [16], are used to encode the documents. After document encoding, an answer retriever is responsible for extracting answers from these documents. Typically, Named Entity Recognition(NER) [17, 18] plays a very important role in this phase.

With the surgiing of pre-trained language models [19, 20], OD-QA has benefited greatly. Initially, a two-stage model framework was proposed for OD-QA. This framework achieves tasks through the collaboration of a reader model and a retriever model. In the retriever-reader framework, retriever can be used for screening documents related to the question from a large number of documents, and the reader is responsible for extracting answers from these documents. The early retriever-reader [21] framework still used sparse encoding, such as TF-IDF and BM25, to retrieve articles. Subsequently, some work [22, 23] switched to using pre-trained models to obtain dense encodings of documents to improve model performance. The reader part usually uses pre-trained language models, in the form of a Machine Reading Comprehension(MRC) model [24–28], to extract possible answers from these articles separately. Although this framework does indeed make significant progress in accuracy and other performance metrics compared to traditional methods, it is not feasible in practice due to its large overhead (each question requires the model to encode all documents in real-time).

Dense Retrieval [5,29,30], recently proposed as a task paradigm for solving OD-QA, has received extensive research from both the industry and academia. This framework reaches a balance between model performance and operational efficiency. In Dense Retrieval, the model pre-process the documents into candidate answers, then encodes and stores them. In inference, the framework only needs to encode the question with little latency and then search for answers from the encoded document embeddings through a lightweight evaluation matrix [5, 30], such as cosine similarity. As a result, this method significantly reduces the time required for inference. Dense Retrieval models have been shown can quickly find answers with little latency from tens of millions documents. Since the model requires both positive and negative samples to differentiate between correct and incorrect answers, the correct samples are labeled by humans, while high-quality negative samples are generally composed of the correct answers labeled for other queries or sentences that are semantically similar but not the correct answer. Therefore, the quantity of

labeled data and the richness of the corpus are closely related to the success of Dense Retrieval. Some research [31, 32] shows that without enough training data, Dense Retrieval models' performance can drop by as much as 30%. This thesis will also investigate the challenge of training Dense Retrieval models in data-scarce environments.

Federated learning [33, 34] is a newly proposed distributed machine learning framework with privacy preserving. In recent years, the total amount of data has experienced exponential growth, and data security and privacy have become new focus issues. Many countries have introduced specific data privacy protection laws to ensure that private data is not leaked or exploited. However, the training of deep learning models inevitably requires a large amount of data, and existing open-source datasets cannot meet the ever-increasing data demand. Therefore, Federated Learning, which use a privacy-protecting method to utilize private data owned by various companies or individuals has been proposed.

In Federated Learning, there is a central server for coordinating training, and entities with private data and participating in the training are called clients. During the training process, the training data on the client will not be uploaded and stay locally on device. The deep learning model will be trained locally, and only the model weights or training gradients will be transmitted from each client to the central server and to other clients. The process is illustrated in . Combining Federated Learning with natural language processing is widely noticed not only in the industry but also in academia. Google [35] was the first to apply Federated Learning on a large scale in the industry, using it to train input method prediction models. Other applications [36–41] include word prediction, text classification, named entity recognition, etc.

As a task with high industrial value, OD-QA can train an intelligent QA robot based on documents owned by each user's device. However, the amount of data on each user's device is usually too small to train deep learning models. Also, it is unacceptable to upload

user's private data out of the local device. Therefore, this thesis studies the combination of Federated Learning and OD-QA, and uses the personal data on each client under the premise of privacy protection to cooperate in training and enhance its performance.

Another issue with Federated Learning is personalization. In the vanilla Federated Learning framework, all the models running on the clients share the same weights, while the data distribution on these clients varies greatly. As a consequence, comparing with using Federated Learning, training with only local data on some clients can yield better results. Some studies [42, 43] have shown that clients with fewer local training samples benefit from Federated Learning, while clients with more samples actually suffer a performance loss in federated learning. One solution is that, compared to using a unified global model on all clients, the model on each client should combine both the local and global models to obtain a model suitable for its own data. There are also academic efforts [44–47] exploring Personalized Federated Learning, which train Personalized Federated Learning models through fine-tuning on local training data, or via transfer learning and knowledge distillation. This thesis also investigates the issue of Personalized Federated Learning in the context of OD-AS2.

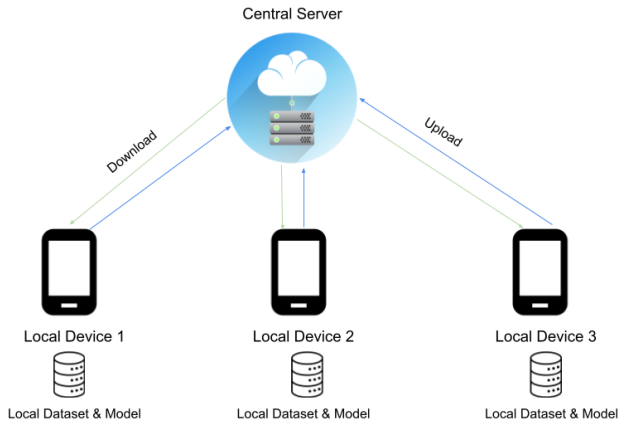

Figure 1.2 : Classical Federated Learning architecture. The local model is uploaded to the server to do the aggregation, and each client gets the aggregated model from the server.

However, Federated Learning is not omnipotent, and its protection of privacy is not guaranteed. Recently, there have been many works [48–51] dedicated to studying data leakage issues in federated learning. Some of these studies have found that in some tasks of natural language processing, even if Federated Learning is applied, any party involved in the training can still reconstruct part of the training data from the training gradients. However, research on privacy issues in federated learning primarily focuses on server-side attacks to obtain model weights or intercepting the gradient updates transmitted between the server and clients to inversely reconstruct the training data. These two attack methods are effective against all federated learning systems because they all require a certain degree of information transmission. Therefore, their research scope goes beyond this thesis.

In our proposed method, due to the presence of information exchange between clients, which is not involved in traditional federated learning, and because we transmit document embeddings in a manner different from most federated learning algorithms, we need to investigate the new privacy issues introduced by our method.

1.2 Research Problems

As introduced in the previous chapters, how to train Open-Domain Answer sentence selection (OD-AS2) in a distributed setting through privacy preserving is an urgent and yet unresolved issue. Second, if Federated Learning is used to resolve the task of training a distributed OD-AS2 framework, how to conduct effective personalized model training in distributed OD-AS2 has not yet been solved. This thesis addresses these two main problems, which cannot be addressed by the previous methods.

- **Research Problem 1**: *Training OD-AS2 models on private data in a distributed scenario*: The usual practice is to train a local model using local training data. The result of doing this is that the model performance would be very poor on clients with scarce data. If all training data is centralized and trained on a central server, it

would cause serious privacy leakage.

- **Research Problem 2**: *Solve the limitations of single global model setting in the distributed OD-AS2 scenario*: In the context of OD-AS2 models acting as personal intelligent QA assistants, it is necessary to conduct personalized training on local data. Moreover, in distributed OD-AS2, clients with scant data would result in a severe performance drop of the local model, while clients with a large amount of data would not benefit from distributed training.

To address the aforementioned research problems, we will develop efficient algorithms to train high-performance Personalized Distributed OD-AS2 tasks under the premise of privacy preserving. Specifically, we break down this task into two core tasks.

- **Task 1 (to Problem 1)**: *Utilizing the private data on various clients, an OD-AS2 model is trained under the premise of privacy protection*. Recent studies have used Federated Learning to train models in natural language processing tasks that require privacy protection. However, no work has yet been done on solving OD-AS2 in a distributed scenario. As a task of great industrial value, it is necessary to study on how to build a practical and high-performance distributed OD-AS2 framework.
- **Task 2 (to Problem 2)**: *Personalized distributed OD-AS2 model training*. We hope to achieve a balance between the unified global model and local models by personalize global model in local dataset. This approach provide clients which have large local dataset with better performance in distributed training. At the same time, it can also enhance the performance of those clients' models where the local dataset is small or data distribution greatly deviates from other clients.

1.3 Major Contributions

The major contributions of this thesis are summarized below.

- **Contribution 1 (in Task 1)**:We proposed the distributed OD-AS2 scenario and demonstrated its necessity and importance for research. We propose a Privacy-preserving Distributed OD-AS2 method, dubbed PDD-AS2, this method combines the vanilla FedAvg with the AS2 approach to address distributed OD-AS2 problem. In this method, we utilizes training data on different clients while eliminating the need to transfer the raw data between clients.

  We further test our method on a new Federated OD-AS2 benchmark based on the CNN News dataset. This benchmark simulates the differences in data distribution across various clients in federated learning based on the types of news. This CNN dataset is widely used in various NLP benchmarks. The experiment demonstrates that our method can greatly improve model's performance on OD-AS2 under distributed settings by leveraging training data on different clients in a privacy-preserving way.

- **Contribution 2 (in Task 2)**: We propose a personalization method applicable to the distributed OD-AS2 scenario. This method is implemented after the traditional stage of federated learning. In this method, we personalize a client-customized query encoder for each client. This approach allows the query encoder to adapt to each user's different language habits and question styles while keeping the document encoder unchanged, maintaining the model's robustness in handling various documents. However, this method does not address the decline in the quality of negative samples and the resulting decrease in training effectiveness caused by the scarcity of data. At the same time, we also propose a negative sampling method called Fed-Negative. This method shares training data by transmitting context embeddings on other clients. Experiments show that our proposed method can greatly enhance the our distributed OD-AS2 model.

1.4 Thesis Organization

This thesis is organised as follows:

- Chapter 2: This chapter present a literature review of this thesis, including Open-Domain Question Answering, Answer Sentence Selection, Federated Learning.
- Chapter 3: This chapter introduces a framework for solving the training of OD-AS2 in a distributed setting while preserving privacy (PDD-AS2) based on Federated Learning. It also explores its performance, how its performance varies in scenarios with scarce data, and other issues encountered during training. Finally, experiments show that our method is better other comparison methods on the dataset.
- Chapter 4: This chapter presents a personalization training method for OD-AS2 cooperated with Federated learning. This approach optimizes for each client user's unique queries by fixing the context encoder and training the query encoder locally. Moreover, we introduce Fed-Negative, a method to optimize personalization training in a Federated Learning scenario. This method boosts model performance by swapping context embeddings between clients. In this chapter, we test our proposed personalized method on benchmark dataset. The results show that our method can further improve the performance of the OD-AS2 model in a distributed scenario. In addition, in this chapter, we also investigate whether our proposed Fed-Negative would cause privacy leakage when swapping context embeddings.
- Chapter 5: This chapter makes an conclusion of the thesis and discusses recommendations for future work.

Chapter 2

Literature Review

2.1 Open-Domain Question Answering

In this task, the model answers a given question using a collection of documents. It does not require a specified context. In practice, the system first retrieves relevant documents from a collection. The collection often consists of local documents or web archives. Then, an answer is generated from relevant documents as a final answer. Generally speaking, most OD-QA methods can be divided into three classes: 1. Traditional OD-QA methods, 2. Two-stage methods of Retriever-Reader, 3. Bi-encoder based Dense Retrieval.

Traditional OD-QA methods. Traditional OD-QA system often consist of a multi-stage method, i.e., query analysis, context retrieval, and answer retrieval [12–14].

In query analysis, a query is used to generate search queries. The search queries are therefore facilitated in the following steps. First, some linguistic methods are used to extract keywords from the query by using pos-tagging [17], stemming [17], and parsing [52]. Afterward, the type of the query is classified by some pre-set types (e.g, when, how).

In context retrieval, the system uses Information Retrieval(IR) methods to select relevant contexts or passages from the document collections using search queries. Tf-idf [16] and BM25 [15], which use probability to calculate the score between queries and documents, are two of the most successful methods in context retrieval.

In answer retrieval, the answer is extracted by using previously retrieved documents

and processed search queries. Therefore, the performance of this stage is highly influenced by the result of previous stages. Traditional OD-QA systems often use factoid questions so that the answers are usually a special name-entity or text-span in the documents. Thus, these systems rely heavily on Named Entity Recognition (NER) methods [17, 18]. In addition, web search engines are often used to validate the answer candidates by a simple principle: a good query and answer pair can return many documents which contain useful information or element about query and answer [12].

Retriever-Reader methods. In a Retriever-Reader system, the Retriever aims at retrieving query-related documents or contexts. The Reader aims to retrieve the answer to the query from the previously retrieved documents or contexts. Retriever is usually an IR model, and Reader model often has a form of MRC model. The early Retriever-Reader methods employed traditional Information Retrieval (IR) techniques such as Tf-idf and BM25 to search for relevant articles or content. DrQA [21] is the first method to integrate traditional IR techniques with modern Neural Machine Reading Comprehension (MRC) models. The similarity between documents and queries is calculated using Tf-idf.

$$[CLS]\ Query\ [SEP]\ Context\ [SEP], \tag{2.1}$$

where $[CLS]$ and $[SEP]$ are special tokens in BERT. Afterwards, they apply a dense layer on the last layer output of $[CLS]$ token. Therefore, the probability that context p is a relevant document to query q can be denoted as:

$$P(q,p)_{positive} = Softmax(logit_{positive}(h_{[CLS]})), \tag{2.2}$$

where $logit_{positive}$ is the logit represents positive label in a binary classification network, $h_{[CLS]}$ is the embedding of $[CLS]$ token. Reader is the other core component in a Retriever-Reader OD-QA system. Reader is usually implemented by a neural MRC model. The goal of the Reader model is to find the answer of the query from documents. Most of the Reader models are extractive reader, which aims at predicting the start index and end index of the answer in the given documents [24, 26–28, 53]. We show the

formulation of answer extraction in below, given a query q and the previously retrieved candidate passages p_i, the inference procedure is described as :

$$P_{start,i}(s) = Softmax(h_{p_i} W_{start}), \tag{2.3}$$

$$P_{end,i}(t) = Softmax(h_{p_i} W_{end}), \tag{2.4}$$

$$P_{selected,i} = Softmax(\hat{h_{[CLS]}}^T W_{selected}), \tag{2.5}$$

where $W_{start}, W_{end}, W_{selected}$ are learnable parameters, $\hat{h_{[CLS]}}$ is the set of the embedding of $[CLS]$ token for each p_i. However, these methods usually yield tremendous computational costs. In real-world applications of OD-QA, the number of documents can be millions or billions. Therefore, a more efficient method is needed for the implementation of OD-QA methods.

Bi-encoder based Dense Retrieval Bi-encoder, also called dual-encoder or two-tower encoder, is the architecture that employs two identical but independent encoders. These encoders are usually called query encoders and context encoders, respectively. The query and context are encoded separately by two encoders in this system. Then, the similarity score is computed by some scoring metric (e.g., cosine similarity). ORQA [54] employs a bi-encoder retriever with two independent BERT-based encoders. They represent query and contexts by the logit of $[CLS]$ token. Then, the similarity score is computed by their embeddings. Given a query q and a context p, the relevance $s(q, p)$ can be denoted as :

$$h_q = Enc_q(q)[CLS], \tag{2.6}$$

$$h_p = Enc_p(p)[CLS], \tag{2.7}$$

$$s(p, q) = h_q^T h_p, \tag{2.8}$$

where Enc_q is the query encoder, Enc_p is the context encoder, respectively. DPR [26] gets rid of the expensive pre-training stage by learning on paired queries and answers sorely. With this aim, DPR proposes introducing negative samples from the whole documents corpus. They proposed several methods for sampling these negatives, including

random selected documents, top documents returned by BM25, and the gold documents paired with other queries. These negative samples improved the performance of Dense Retrieval significantly.

Negative Sampling. Negative sampling, which pairs each training query with wrong candidate passages or contexts in training, is crucial to the success of Dense Retrieval. From these generated negative samples, the encoder learns to represent queries and contexts with more describable dense vectors. Therefore, the quality of the negative samples determines the performance of Dense Retrieval. In the early days, random negatives, where an incorrect answer was randomly selected from all text fragments, were widely used. Another similar sampling method is called gold negative. This method randomly selects from the correct answers of other questions to serve as negatives. There is also a very efficient sampling method called in-batch negatives. In-batch negatives was used in [55, 56] as an effective method that boost the number of training examples in training. The basic idea of in-batch negative is to reuse the negative samples of other queries in the same batch. However, normal negative sampling methods which pair each query with random negatives or gold negatives are proved to be sub-optimal in some work [57]. Hard negative sampling, which samples top-K documents or contexts as negatives yields semantically similar negative samples. Some early methods utilize IR techniques to find semantically similar negative samples. Recently, some research has been using trained models themselves to obtain semantically similar samples. [58] propose to retrieve the documents with highest similarity score as hard negatives before the training with a Dense Retrieval model warmed up on other negative-sampling methods.

2.2 Answer Sentence Selection

Answer sentence selection (AS2), which is one of the essential tasks in question-answering, has attracted much interest since the recent development of intelligent assistants. The definition of AS2 can be described as: with a question q and many candi-

dates answer sentences set S. The task aims to choose a sentence s_k to answer the query q. Some machine reading comprehension datasets also provides sentence-level answers, such as Natural Questions and Hotpot QA. The first appearance of AS2 task was in the TREC competition [59]. With the surge of neural networks, it has significantly improved, such as [60–62].

Previous work usually experiments on small datasets. In TREC [59], there are only 3000 QA pairs, and each query has around 20 candidate answer sentences. [11] proposes a relatively larger dataset ASNQ with 30000 QA pairs. Each query is paired with four candidate answers. However, in our proposed distributed OD-AS2 setting, a query may have millions or even billions of candidate answers. The queries in the client, which has the fewest documents, are paired with thousands of candidate answers, which is much greater than those datasets. However, now existing methods are not efficient for solving such complicate settings.

With the surge of transformer-based models, many works apply transformer-based models in AS2 task such as [11, 63]. However, these methods involve a computational-heavy inference process, which is not applicable in our setting. In our work, we propose a Dense Retrieval based method for distributed OD-AS2. This method is efficient for real-time inference and maintains a usable performance.

2.3 Federated Learning

Federated learning was first raised by [64]. It allows clients such as organizations or personal devices to train a model together in a private-preserving manner. In this diagram, the central server only coordinates the training process, while all the training data is decentralized in each client. Let θ represent global model and $\{\theta_k\}_{k=1}^{k}$ represent k local models. Let $D_k = \{(x_{k,i}, y_{k,i})\}_{i=1}^{n_k}$ be the local dataset where $x_{k,i}$ is the i-th sample in k and $y_{k,i}$ is its ground truth answer.In each round r, a subset of k_A clients are chosen to participate in local training. The global model then collect all local updated weight

$\theta^{r+1} \leftarrow \frac{1}{k_A}\sum_{k=1}^{k_A}(\theta_k^r)$ and send new weight to local models. In FL, the global model θ tries to optimized the total loss, i.e.,

$$\min_{\theta} \mathcal{L}(\theta) \triangleq \sum_{k=1}^{K} \frac{n_k}{N}\mathcal{L}_k(\theta),\ N = \sum_{k=1}^{K} n_k, \tag{2.9}$$

where $\mathcal{L}_k(\theta)$ represent the loss on $\tilde{D}_k$ for client-k, i.e.,

$$\mathcal{L}_k(\theta) = \frac{1}{n_k}\sum_{i=1}^{n_k} \mathcal{L}_{CE}\left(\tilde{y}_{k,i}, F(x_{k,i};\theta)\right), \tag{2.10}$$

where $\mathcal{L}_{CE}$ is the cross-entropy loss and $F(\cdot;\theta)$ is the probabilities of each class by model prediction. The update on client-k begins from $\theta_k \leftarrow \theta$ and then optimize the local loss $\mathcal{L}_k$ by gradient descent on $\tilde{D}_k$ with each step as

$$\theta_k \leftarrow \theta_k - \eta\nabla_\theta \mathcal{L}_k(\theta_k), \tag{2.11}$$

where η is the local learning rate. Then the server collects the local models by:

$$\theta \leftarrow \sum_{k=1}^{K} \frac{n_k}{N}\theta_k. \tag{2.12}$$

The server then transfer θ to all clients as the starting point for the next round of local update.

FedAvg [65] is the most famous and widely used federated learning algorithm. It performs SGD in parallel on some selected clients, and then uploads the model weights of each client to a central server. The central server averages all the weights and then distributes them back to the clients. In this thesis, the methods we propose are mainly based on FedAvg. Improvements to the Federated Learning training framework and discussions of issues are beyond the scope of this thesis.

Applications of Federated Learning Recently, some work has proposed to solve language modeling in a Federated setting. In industry, mobile keyboard is one of the most popular applications of Federated Language Modeling. In this application, the model predicts the user inputs using a language model. Given the limited hardware resource on

mobile devices and the need for an inference time within 20 milliseconds, a small and efficient network is often used. Most works [35–39] use variants of LSTM as the client model. CIFG [66] is a popular solution used in many works [35, 38] as it ensures inference latency and task performance. Classification is another basis of NLP tasks. Many tasks can be solved by using text classification methods such as sentiment analysis, question answering, and topic labeling. [41] apply a standard FedAvg to TextCNN. Speech recognition recognizes speech in audio and then converts it into text. Most of the modern intelligent assistants, such as Siri and Alexa, are equipped with a wake-up words detection function. This function process user's audio in real-time and on-device to detect a specific wake-up word such as 'Hey, Siri. This setting is a perfect fit for Federated Learning: which requires a robust and local model on each device for local-inference, and the training data is too sensitive for uploading to a central server. [67] propose to use Federated Learning on wake-up words detection. [68] study on the non-iid problem in speech recognition with Federated Learning. However, for the distributed OD-QA/OD-AS2, which requires many user privacy data, no research has yet used Federated Learning to address this problem. In this thesis, we propose a framework based on Federated Learning to solve the distributed OD-AS2 task.

Personalized Federated Learning. The primary purpose for clients to participate in Federated Learning is to obtain a better model compared with local training models. However, many works [42, 43] show that not all clients can benefit from Federated Learning. Clients who have insufficient local data can benefit more from the collaboration. On the opposite, those clients who have sufficient local data found the final global model even worse than their local model. Consequently, a simple global model is not enough for many cases in Federated Learning. As a solution, personalization provides methods that can use both the global shared model with the individual local models to get better performance in each client. Transfer learning [45] lets models utilize knowledge learned from a task to solve problems in another task. [44] propose to use transfer learning as personal-

ization with a Federated setting. Meta-learning trains a model on multiple tasks and aims to learn a robust model for any kind of task. And the model only needs a small number of data when adapting to new tasks. [46] propose that Federated Learning is very similar to Reptile [69], a famous Meta-learning method. Knowledge distillation is also used to train a Personalized Federated Learning model. [47] use Knowledge distillation to transfer knowledge from the teacher(global) model to student(local) model on each client. In this thesis, we investigate and propose a method for personalizing the OD-AS2 model in the context of Federated Learning.

Privacy in Federated Learning. However, Federated Learning does not guarantee that there are no risks of privacy leakage. In fact, the academic community has made significant efforts to study the privacy leakage issues caused by the application of Federated Learning in natural language processing. In Federated Learning, model weights or training gradients are transmitted through networks, and attackers may potentially reconstruct users' private data used for training from this information. For instance, some studies have attempted to reconstruct training data by analyzing the variations in weights of the word-embedding layer in large-scale pre-trained models [51]. Other research endeavors have explored techniques such as inferring training data from model gradients [48, 49] or modifying model weights [50]. However, since these privacy concerns are not the central focus of this thesis, we will only discuss new privacy issues arising from the novel approach we propose in this thesis.

Chapter 3

PDD-AS2: a Framework of privacy-preserving distributed OD-AS2

3.1 Introduction

As stated in the literature review, common OD-QA models use a Dense Retrieval model [24, 26–28, 53] structure of a dual-encoder (also known as a bi-encoder or two-stream encoder). Dense Retrieval encodes the question and candidate answers simultaneously into dense vectors and stores them. During inference, the model uses a lightweight metric such as dot-product to calculate semantic similarity. These model architectures are widely studied in academia and industry for their balance between performance and inference efficiency.

However, training an effective Dense Retrieval model in OD-AS2 requires a large amount of data, human-generated question-answer pairs, and an extremely large-scale document library based on real user data. However, the substantial requirement for user privacy data makes it a formidable challenge to directly apply Dense Retrieval to real-world scenarios, such as in-house data inquiry, individual email searches, and personal intelligent assistants. If we adopt the existing methods, such as training Dense Retrieval models locally, underfitting would occur due to the scarcity of training samples. In the Dense Retrieval training process, a large number of diverse negative samples are also needed to help the model learn the relationship between correct and incorrect samples. Some research [31, 32] has found that, in the case of insufficient sample size, the performance of Dense Retrieval models may drop by 20%. Therefore, these questions have driven us to seek a framework for training Dense Retrieval models using personal privacy

data distributed across various clients, under the premise of privacy protection. Based on federated learning, we propose a **P**rivacy-preserving **D**istributed O**D-AS2**, called PDD-AS2. Although we only tested the implementation based on FedAvg in our experiments, our framework is also compatible with other Federated Learning algorithms. Through this framework, we can utilize the privacy data stored on each device for distributed training of the OD-AS2 model. Our main contributions of this work can be summarized as follows:

- We highlight a promising setting of open-domain answer sentence selection (OD-AS2) for real-world industrial applications and propose a privacy-preserving distributed OD-AS2 (PDD-AS2) learning framework towards effectiveness.
- We construct a new distributed OD-AS2 dataset upon NewsQA, dubbed Fed-NewsQA to evaluate the effectiveness of our framework and its baselines.

In Section 3.2, we give a brief introduction of our proposed method. Section 3.3 presents the training and inference process of our proposed method in detail. Then, we conclude the new benchmark we proposed and experiments results in Section 3.4 and give insightful conclusions about our method in Section 3.5. The chapter summary is in Section 3.6.

3.2 The Overview of PDD-AS2

In this chapter, we present PDD-AS2, a privacy-preserving OD-AS2 framework powered by Federated Learning. The idea behind our framework is that we protect privacy by training the OD-AS2 model locally on individual's private data, only uploading the model weights/training gradients. During the local training process, we specifically identify the training into two different steps: In the first step, we use similar samples selected by BM25 as the negative samples for a certain question to train, this stage can be seen as the

cold start of the whole model. In the second step of training, we use the already trained model itself to find the negative samples for each question in the training set. In these two stages, we do not change our loss function and hyper-parameters, only the negative samples and their corresponding quantities will differ. During these two stages, the model weights of each client will be uploaded to the server and then redistributed to each client's local device in the end of each round. During the inference process, the local model on each client runs independently for inference locally, without the need for participation from the central server. Next, we will first give a detailed definition of the OD-AS2 task, then we will elaborate on the framework of our proposed PDD-AS2.

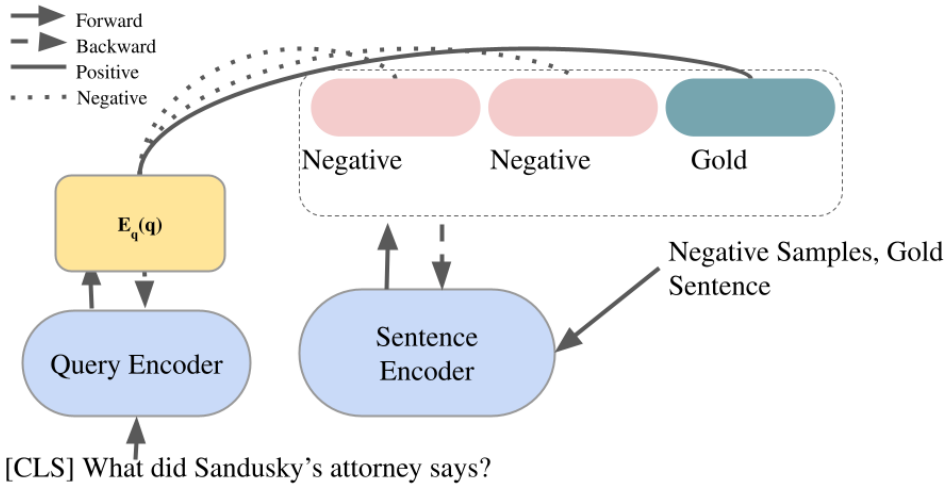

Figure 3.1 : The training process of our proposed PDD-AS2. Query embeddings and negative embeddings are generated in real-time. Then, the loss is calculated and gradient is used to train both query encoder and sentence encoder.

3.2.1 Framework formulation.

In line with existing works [5, 8, 11, 70], we first formulate Open-Domain Answer Sentence Selection (OD-AS2) under distributed setting as follows: For each client $c^i \in \mathbb{C}$ with its large-scale sentence collection $\mathbb{S}^i = \{s_1^i ... s_n^i\}$, it aims to fetch potential answer

sentence(s) s_k^i from $\mathbb{S}^i$ that answers a given query $q \in \mathbb{Q}$. In the OD-AS2 setting, the sentence set $\mathbb{S}^i$ contains sentences from all passages in c^i. If no confusion is caused, we omit the superscript 'i' for a specific client in the remainder.

Usually, a query q and its answer sentence s_q^+ are often provided as positive training data in each client. Hence, it is necessary to sample a set of negative for q to construct negative samples, i.e.,

$$\mathbb{N}_q = \{d | d \sim P(\mathbb{S})\}, \tag{3.1}$$

where $P(\cdot)$ denotes a distribution over $\mathbb{S}$. For simplicity, we omit the query-specific subscript indicator, q.

Then, a contrastive learning framework is usually employed to learn an efficient retrieval model. Formally, a representation learning module is first used to embed q and each $s \in \{s^+\} \cup \mathbb{N}$ and then derive a probability distribution over $\{s^+\} \cup \mathbb{N}$. Specifically,

$$P(\{s^+\} \cup \mathcal{N} | q; \Theta) = \frac{1}{Z} \exp\left(\langle \text{enc}(q; \Theta^{(q)}), \text{enc}(s; \Theta^{(s)}) \rangle\right) \tag{3.2}$$

where $\Theta = \{\Theta^{(q)}, \Theta^{(s)}\}$, Z denotes $\mathrm{softmax}$ normalization term, Θ parameterizes a text encoder for a single vector representation, $<,>$ denotes a lightweight relevance metric (say, a dot product) for their similarity score. Here, $\Theta^{(q)}$ and $\Theta^{(s)}$, whether tied or not, compose a dual-encoder structure for efficient dense retrieval. Lastly, the training loss of contrastive learning can be defined to optimize Θ, i.e.,

$$L^{(\text{ct})}(\mathcal{Q}; \Theta) = -\sum_{q \in \mathcal{Q}} \log P(\text{s} = s^+ | q, \{s^+\} \cup \mathcal{N}; \Theta) \tag{3.3}$$

where $P(\cdot | q; \Theta)$ denotes the probability distribution over $\{s^+\} \cup \mathbb{N}$ for q by Eq.(3.2).

Subsequently, considering the distributed setting of OD-AS2, the overall training loss can be defined as

$$L(\{\mathbb{Q}^i\}_i; \{\Theta^i\}_i) = \sum_i L^{(\text{ct})}(\mathbb{Q}^i; \Theta^i). \tag{3.4}$$

However, directly optimizing Eq.(3.4) cannot deliver a satisfactory performance for each client i since both labeled question-answering pairs and the collection are too scarce

to effectively learn. Therefore, we adopt a popular federated learning method, FedAvg [71], as the backbone of our framework. It will leverage the training data distributed in each client in a privacy-preserving way. We denote the weight of global model as Θ^{global}. For each $c \in \mathbb{C}$ with model weight Θ^i, we update Θ^i with a learning rate of α locally by

$$\Theta^i = \Theta^i - \alpha \nabla L(\mathbb{Q}^i; \Theta^i), \tag{3.5}$$

where L is the loss function of local training objective defined in Eq.3.4. After local updates, each client sends their weights Θ^i to the central server. Central server aggregate the weights by

$$\Theta^{global} = \sum_{i=1}^{k} \frac{|\mathbb{D}_i|}{\sum_{i=1}^{k} |\mathbb{D}_i|} \Theta^i, \tag{3.6}$$

where k is the number of clients, $\mathbb{D}_i$ denotes the volume of the dataset on each client. Note that our PDD-AS2 framework is also compatible with other Federated Learning methods.

Algorithm 1 PDD-AS2

Input: Clients set $\mathbb{C}$, Training set D_i on client c_i, global model weight Θ^{global}, learning rate α
Begin: Initialize the global model Θ^{global} .
for $r = 0, 1, \ldots, R$ **do**
  for Client $c_i \in \mathbb{C}$ **in parallel do**
    Initialize local model $\Theta^i \leftarrow \Theta$.
    for batch *b* in D_i **do**
      Send queries $q_b \in b$ to other clients $c_j \in \mathbb{C}$
      Receive negative samples $\mathbb{N}_{q_b}$
      $\Theta^i \leftarrow \Theta^i - \eta \nabla \mathcal{L}(s^+ \cup \mathbb{N}; q; \Theta^i)$
    end for
  end for
  Server optimize Θ model weights
end for

3.3 Training Pipeline of PDD-AS2

Finally, we introduce the overall training pipeline of our PDD-AS2 framework which adapted from some prevailing works [5, 70]. In this framework, we train the encoders with different kinds of negative samples under FedAvg. Due to the instability of the model in the early training stage, we initially sample BM25 negatives N^{BM25} to warm up the model, following the approach of some works [70, 72]. The advantage of the BM25 is that it is an unsupervised method. However, because it cannot extract the deep semantics of the query and the candidate answers, the performance of BM25 is not as good as deep learning models. Therefore, after completing the warm-up, we use the trained model itself to extract negative samples. In the second stage, at the beginning of each training epoch, we use the trained sentence encoder and query encoder to encode all potential answers and questions respectively. Then, we use the cosine-similarity method to find the semantically closest sample to each query as the static hard negatives, excluding the correct answer. Although for optimal performance, hard negatives should be recalculated after each training step, this would result in a huge computational cost. The training method of static hard negatives has also been proven very effective in many works. We update both $(q;\Theta)$ and $(s;\Theta)$ by $\mathcal{L}$ defined in Eq.3.4. The overview of our Federated Learning method is shown in Algorithm.1.

3.3.1 Retrieval Schemes

Our model is compatible with two retrieval schemes: sentence-level retrieval and passage-level retrieval. For sentence-level retrieval, we retrieve the top sentences follow the probability distribution defined in Eq.**??**. For passage-level retrieval, based on the fact that sentences are extracted from their source passages, we retrieve the passage with

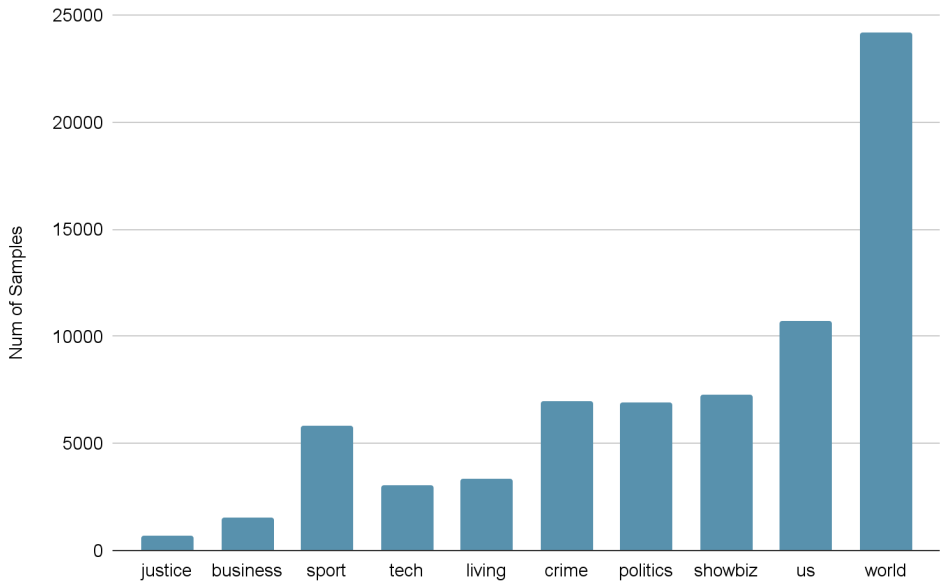

Figure 3.2 : Statistics of each genre in our Benchmark

highest relevance score as

$$f(p,q) := \max_{s \in p} \{ < (q;\Theta), \\ (s;\Theta) > \}, \forall s \in \mathbb{S}, \tag{3.7}$$

where $s \in p$ represents the set of sentences in a given passage p. The additional cost of sorting sentence scores can be ignored [73]. Therefore, the inference speed of our sentence-based passage retrieval is the same as for sentence retrieval.

3.4 Experiment

3.4.1 Fed-NewsQA: A Multi-client OD-AS2 Benchmark

To better evaluate our method in a distributed setting, we propose a multi-client OD-AS2 benchmark based on NewsQA. Recent OD-QA works often use datasets such as SQuAD [74], TREC [7], WebQuestions [75], Natural Questions [76] in their experiments. However, we propose to use NewsQA [77] as our original dataset for two main reasons.

First, to better mimic the difference between each client's personal documents and the data scarcity problem in the real-world cases, we propose to split the dataset into different genres for simulating different clients. Among all these datasets, we find that NewsQA meets our requirements perfectly. We split the dataset into different genres directly from the web-link of each passage. We choose ten genres from NewsQA since the remaining genres do not have enough number of samples in the dev/test set. Each of these genres represents a different client in our Federated Learning setting. The statistics of each genre are shown in the Figure 3.2.

Second, NewsQA significantly outnumbers some other datasets on the distribution of the more difficult reasoning questions, such as SQuAD [77]. We believe inferencing and reasoning queries are essential for OD-QA/OD-AS2 in real-world cases.

3.4.2 Implementation

We use pre-trained DistilBERT [78] by Hugging Face as our model. We use AdamW with a learning rate of 3e-5. We use Faiss [79] to perform the similarity search. We use open-sourced BM25 model in training. Queries and sentences are truncated to a maximum of 32 tokens and 512 tokens, respectively.We represent query embeddings simply using the $[CLS]$ token, and we represent sentence embeddings using the average pooling of word embeddings in the sentence.

The details of our training procedure are described as follows: In the federated static negative training, we pair each query with BM25 negatives and gold-negatives with a batch size of 8 in the warm-up stage. Then we replace them with static hard-negatives. To demonstrate the influence of numbers of negatives, we also experiment with settings with different numbers of negatives. We enable in-batch negative in this stage. We implemented vanilla FedAvg as our Federated learning framework. We aggregate local weights after each epoch.

We report two levels of metrics in our experiments: sentence-level and passage-level.

The retrieval procedure of both levels is defined in section 4.2.4. In both levels, we report the MRR@10, Recall@1,20,100 scores.

3.4.3 Baselines

We conduct experiments to compare of our method with several Dense Retrieval methods, including: (1) Dense Retrieval trained with random negative [29] (2) Dense Retrieval trained with BM25 negative [30]; (3) Dense Retrieval trained with STAR [70].

3.4.4 Experiment Results

Table 3.1 : Results on our Fed-NewsQA Benchmark.

	Sentence-level Retrieval				**Passage-level Retrieval**			
Models	**MRR@10**	**R@1**	**R@20**	**R@100**	**MRR@10**	**R@1**	**R@20**	**R@100**
Upper Bound								
Central-training	0.338	0.284	0.629	0.781	0.502	0.447	0.553	0.821
Sparse Retriever								
BM25	0.172	0.152	0.343	0.533	0.343	0.288	0.345	0.598
Dense Retriever								
dense retrieval-Random Neg	0.194	0.171	0.466	0.62	0.376	0.323	0.401	0.702
dense retrieval-Bm25 Neg	0.188	0.151	0.475	0.639	0.353	0.303	0.388	0.679
dense retrieval-STAR	0.232	0.190	0.535	0.679	0.403	0.350	0.421	0.709
Dense Retriever: Ours								
PDD-AS2	0.261	0.217	0.546	0.695	0.429	0.395	0.479	0.745

The main result of our experiments is shown in Table 4.1. In the first experiment, compared with the Dense Retrieval baselines trained on a single client, our PDD-AS2 outperformed all other methods. This is because the number of documents in some clients are very restricted. Our method can leverage training data on each client in a privacy-preserving way. Therefore, our federated method can achieve better performance than non-Federated methods.

In the second experiment, We explore the influence of $num\_negatives$ in our setting. We experiment with the combinations of different numbers of negatives used in each

method. The result of different $num\_negatives$ is showed in Fig 4.2. We show the impact of $num\_negatives$ on training stages separately. The maximum number of hard-negatives we can test in training is limited due to GPU RAM cost. For BM25 negative sampling and static hard-negative sampling, we train the model with our PDD-AS2 framework from the beginning of our training procedure.

We found that insufficient number of negative samples can lead to much worse performance. This is intuitive since the model saw fewer numbers of samples during training. However, larger numbers of negatives are not affordable even on servers due to hardware limitations. More effective methods are needed to implement more negative samples in the training.

Table 3.2 : Different numbers of negatives in Training

	Sentence-level Retrieval				**Passage-level Retrieval**			
Models	**MRR@10**	**R@1**	**R@20**	**R@100**	**MRR@10**	**R@1**	**R@20**	**R@100**
Dense Retriever with BM25 negatives								
num_negative=2	0.143	0.123	0.302	0.489	0.310	0.247	0.311	0.582
num_negative=8	0.172	0.151	0.343	0.533	0.343	0.288	0.345	0.598
Dense Retriever with STAR								
num_negative=2	0.201	0.160	0.506	0.655	0.352	0.305	0.379	0.705
num_negative=8	0.232	0.191	0.535	0.679	0.403	0.350	0.421	0.709
PDD-AS2								
num_negative=2	0.242	0.193	0.516	0.645	0.392	0.354	0.432	0.719
num_negative=8	0.261	0.217	0.546	0.695	0.429	0.395	0.479	0.745

3.4.5 Influence of dataset Size

In this section, we first want to know if our PDD-AS2 can effectively solve data scarcity problem on each client by leveraging data on different clients. In training, we select different ratios of data randomly. We present the sentence-level R@1 score on our Fed-NewsQA in Figure 3.3. Compared with single-client training, the PDD-AS2 can

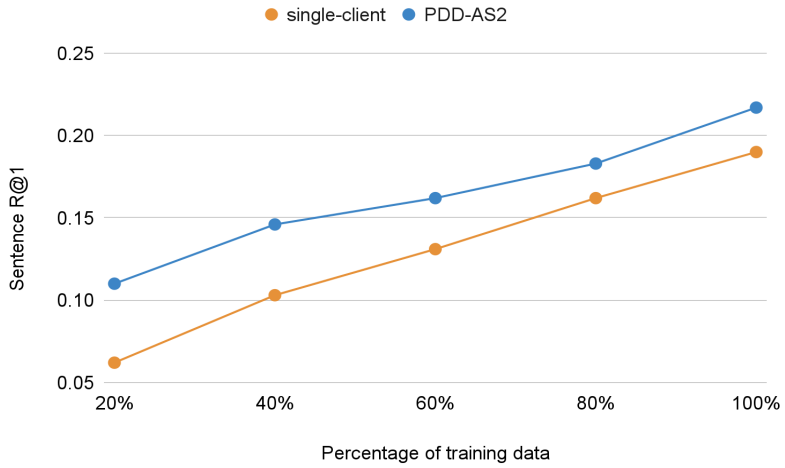

Figure 3.3 : Sentence R@1 of our PDD-AS2 and baseline with single-client training

achieve higher accuracy in all data ratio settings. Moreover, the problem in single-client is more serious when their own local dataset size is small . As a consequence, PDD-AS2 can bring about a more significant performance improvement over single-client training.

Also, we explore to what extent each client benefits from the PDD-AS2. We show the performance improvement in sentence-level R@1 on Fed-NewsQA of each client in Figure 3.4. We found that clients with fewer training data can benefit more from the PDD-AS2 framework. These results indicate that our framework can effectively leverage the training data on different clients. However, performance on some clients with a larger amount of training data was decreased while applying our framework, implying the need for personalization in this scenario.

3.4.6 Influence of query hubness

However, retrieving all top-k hard negatives from similarity search or BM25 engine can lead to a performance drop in some scenarios. The reason is that, not every possible answer for a given query q_i has been labeled as positive. This is very intuitive since most machine reading comprehension datasets only label the answer of the query, which is only in its context passage. However, in OD-AS2, possible answers from all passages must be

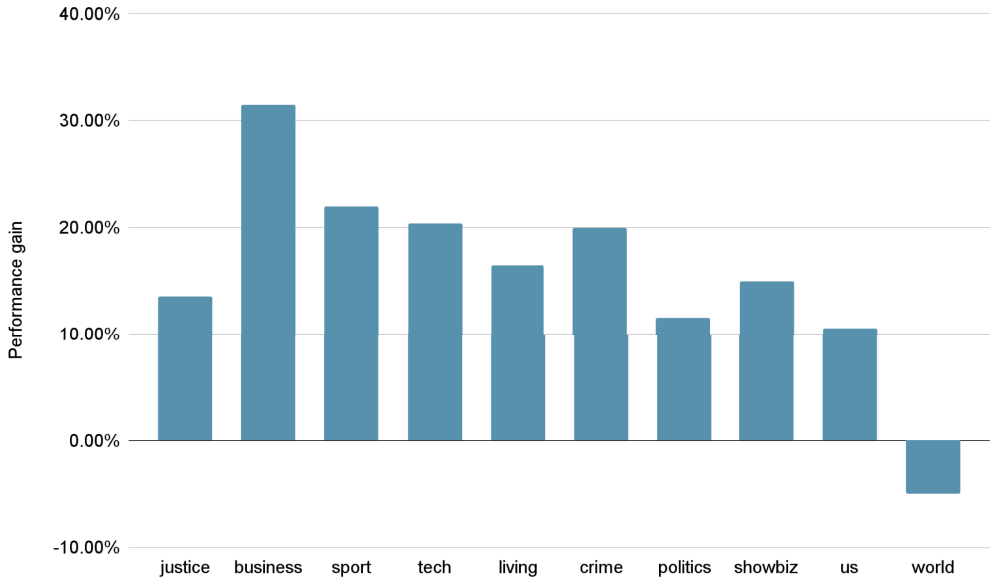

Figure 3.4 : Performance gain on Sentence R@1 of each genre in our benchmark

labled as positive. This problem is more severe when the query is not specific and precise.

As a consequence, for each q_i, if we retrieve all top-k sentences as negative, we actually harm the performance of the model. We conduct a case study in Table 3.4. The case study shows that whether or not the query is specific and precise, the top-k negatives often contain possible answers that were not labeled as positive. We refer to this problem as 'query hubness'. To alleviate this problem, we uniformly sample *n* negatives from *k* candidate where k$\gg$n in our approach. This approach yields better results when we choose a correct k. The difference in model performance in different k is shown in Table 3.3. However, more theoretical insight is needed in query hubness problem.

3.5 Chapter Summary

In this work, we propose a Privacy-preserving Distributed OD-AS2 method, dubbed PDD-AS2. Our method utilizes training data on different clients while eliminating the need to transfer the raw data between clients. We fisrt train both query encoder and sen-

Method	Sentence R@1	Passage R@1
k=10	0.121	0.235
k=50	0.202	0.379
k=100	0.211	0.352
k=300	0.217	0.395

Table 3.3 : Different k while sampling 10 negatives

tence encoder with static hard-negatives under a Federated framework. We further test our method on a new Federated Open-domain Answer Sentence Selection benchmark based on NewsQA. This benchmark better mimics real-world cases than other benchmarks in terms of data distribution and query types. The results show that our method can greatly enhance the performance of OD-AS2 under distributed settings by leveraging training data on different clients in a privacy-preserving way.

	Case 1	Case 2
Question	What did the lawyer say	Who will star in the upcoming ABC pilot “The Manzanis”?
Gold answer	Murray defense lawyer Michael Flanagan, who was in court to defend Dr. White Wednesday, said after the hearing that he believed Murray should be eligible for early release if he is given prison time	When Kirstie Alley cleared the 100 lb. weight-loss hurdle this summer, it was time for a big, fat celebration.
Hard-negative 1	In addition, Anthony’s attorney Charles Greene asserted he would also invoke the Fifth Amendment on her behalf if questioning delved into the 2008 death of her 2-year-old daughter, Caylee.	And she’s ready for her next challenge: “What I’m looking for is to be madly, deeply in love,” says Alley, who will also star in the upcoming ABC pilot, “The Manzanis.”
Hard-negative 2	CNN) – Attorneys representing Casey Anthony invoked her Fifth Amendment right against self-incrimination 60 times during a deposition given in a civil suit against her, according to a transcript of the proceedings.	Kirstie Alley said she’s going to start dating “butt-ugly men” on an episode of “The Ellen DeGeneres Show” airing Friday.

Table 3.4 : Case study of retrieved hard-negatives, all text samples are directly retrived from the original dataset source

Chapter 4

Personalized Distributed Open-Domain Answer Sentence Selection by client-side finetune

4.1 Introduction

In a distributed Open-Domain Answer Sentence Selection (OD-AS2) scenario, Federated Learning [65] can be used to train private data on clients under the premise of privacy protection. By retaining the training data for local training and only synchronizing the model weights or training gradients, Federated Learning can make use of private data. However, there can be substantial differences in the amount of data on clients. Some research [42, 43] has found that in Federated Learning, clients with less local data often see greater improvements, while those with more local data often see smaller improvements, and may even experience performance degradation. On the other hand, given the substantial differences between individual data on all clients, a unified global model may perform poorly in these scenarios. Therefore, we urgently need a solution that can balance the features of the global model and the local model in a distributed OD-AS2 scenario.

Personalization, as a solution, can be integrated with Federated Learning to alleviate this problem. By fine-tuning on the global model with local data, a personalized global model based on local data can be obtained. Many works [45–47] have explored the combination of Federated Learning and Personalization. These methods cover transfer learning, knowledge distillation, etc.

However, the aforementioned methods also present many problems in the context of distributed OD-AS2 scenarios. For example, in the implementation of personalized transfer learning, due to significant differences between local and global samples, and the

scarcity of local samples, many clients tend to overfit and experience catastrophic forgetting during the finetune stage. Moreover, common OD-QA/OD-AS2 algorithms require a large number of negative samples to distinguish between correct and incorrect answers. During local training, if only local data is used, the quantity of available negative samples for OD-AS2 becomes too small, leading to poor training results. If knowledge distillation is employed, due to the insufficient number of local training samples, it is impossible to distill all the capabilities of the teacher model trained with large datasets on multiple clients. Personalization achieved through meta-learning also greatly depends on the size of the local dataset during local training, which is also not the best choice. Therefore, there is a need for a distributed OD-AS2 personalization method that can balance the performance of local models and the need of local datasets.

Therefore, in this chapter, we propose a novel personalized approach for distributed OD-AS2. In this personalized approach, we combine it with the framework we proposed in former chapter, to train the query encoder on local data only, while the sentence encoder‘s weight is fixed without training. A very intuitive explanation is that the difference between personal documents among different clients is relatively small, while personal queries can vary greatly due to language habits, etc. Furthermore, the content of the query is usually easier to understand, while understanding the context/documents requires deeper language comprehension skills. Thus, keep the sentence encoder weights obtained from training on a large amount of client data fixed can avoid catastrophic forgetting caused by fine-tuning.

At the same time, this approach can significantly reduce training overhead while enhancing training effectiveness. In our experiments in previous chapter, we found that the higher the number of negative samples involved in training at each training step, the better the model performance is usually. However, the primary obstacle to the number of negatives in training is memory overhead. Since the sentence encoder needs to participate in training in general training processes, every negative sample encoded with the

sentence encoder needs to calculate gradients and back-propagate, resulting in substantial memory overhead. In our method, since the sentence encoder's weight is fixed, the back-propagation is not required on negative samples. Therefore, in this method, we can introduce more negative samples to aid training.

Based on the above premise, we propose a novel negative sampling method in the context of distributed OD-AS2 scenarios, which introduced a large amount of negative samples picked from other clients, called Fed-Negative. In this method, we enhance personalized training by sharing context embeddings from other clients. In this process, we send the query embeddings from one client to another to look for similar context embeddings, and then send them back to the original client for training. Since our proposed personalized training approach does not require training a sentence encoder, it can greatly increase the number of available local negative samples while maintaining training efficiency and low memory usage. This feature is particularly suitable for distributed scenarios where end devices have limited computational power. Meanwhile, this chapter will also study the privacy leakage issues that this method may bring about.

The rest of this chapter is organized as follows. The review of other Personalized Federated Learning methods is introduced in Section 4.2. Section 4.3 illustrates the overall pipeline of our method and the components of our framework in detail. After that, we analyze the experimental results in Section 4.4 and then conclude our chapter in Section 4.5 .

4.1.1 Review of Personalized Federated Learning Approaches

In this section, we briefly describe some methods of Personalized Federated Learning.

Transfer learning. Transfer learning [45] lets models utilize knowledge learned from a task to solve problems in another task. [44] propose to use transfer learning as personalization with a federated setting. They continue training the shared global model on a local dataset. As a result, the model can keep the knowledge learnt from massive private

training data in Federated Learning stage. However, catastrophic forgetting would happen if we fine-tune the model on the local dataset too much.

Meta-Learning. Meta-learning trains a model on multiple tasks and aims to learn a robust model for any kind of task. And the model only needs a small number of data when adapting to new tasks. [46] propose that federated learning is very similar to Reptile [69], a famous Meta-learning method. In meta-learning, meta-training builds the global model on multiple tasks, and meta-testing adapts the global model separately for different tasks. Therefore, Federated Learning is like meta-training, and personalization is like meta-testing. They further modify the FedAvg to address better results in this two-stage training diagram.

Knowledge distillation. Knowledge distillation [47] a technique that compresses knowledge from single or multiple models to another model. Usually, the former model, called the teacher model, is much larger than the latter model, which is called the student model. In personalization, one of the biggest issues is overfitting caused by an improper personalization process. Some work [33, 43] proposes to mitigate this issue by using knowledge distillation together with transfer learning.

4.2 The Proposed Approach

4.2.1 Fed-Negative: Cross-client Negatives

As mentioned in the above sections, using local negatives samples only cannot fulfill negative samples' needs in terms of quality and quantity in some clients with few document collections. Building on this problem, we propose Fed-Negative: a cross-client negative sampling method inspired by dynamic negative sampling for introducing more diverse negative samples. By introducing additional context embeddings stored on other clients, Fed-Negative has expanded the selection of negative samples available during

model training. Therefore, during the training, the model is more likely to select semantically similar negative samples, thereby enhancing training performance. Given a client c, we first encode q into representations by $(q;\Theta)$. Then we select a subset of clients from the whole client set as

$$C_s = Select(\{C\}), c \notin C_s, \tag{4.1}$$

where the select function can based on network condition or geography distance estimated by client's region. Then we send the query representation $(q;\Theta)$ to each client in C_s.

Once each client receive the query, they did a similarity search on their own sentence embedding matrix to retrieve top *n* sentences embeddings and send them back to c. c choose top *n* negatives from all negatives by the similarity score as

$$N^{fed} = TopK(\{(N_{c_k})\}), c_k \in C_s \tag{4.2}$$

where N_{c_k} is the negative set of q sampled in client c_k.

One concern with this method is the issue of privacy leakage, that is, whether we can restore the original training data from the context embeddings, or extract relevant information. In the subsequent sections of this chapter, we will study the privacy issues of this method.

4.2.2 Client-customized Query Encoding

On top of Fed-Negative, we propose client-customized query encoding inspired by query-side fine-tuning. We aim to provide each client with a personalized query encoder to resolve miscellaneous queries. For this purpose, we personalize $(q;\Theta)$ with local training while fixing the $(s;\Theta)$. $(s;\Theta)$ shares a global weight among all clients. Therefore, by keeping the weight of sentence encoder, which is well trained in Federated Learning, fixed, we retain the model's ability to understand individual documents. Continuing to train the query encoder on the local dataset allows the local model to better adapt to the language habits of different client queries. In this stage, we utilize our proposed Fed-

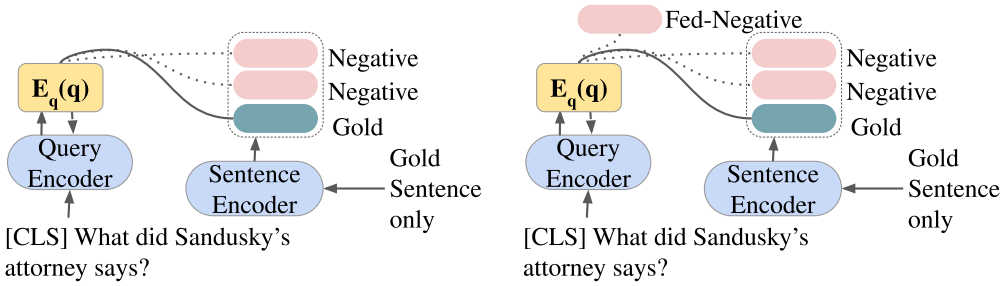

Figure 4.1 : (a) Train query encoder $(q;\Theta)$ and sentence encoder $(s;\Theta)$ with Static hard-negative sampling (b) Personalize the query encoder $(q;\Theta)$ with Fed-Negative

Negative for diverse negative samples.

4.2.3 Training Pipeline.

Finally, we introduce the overall training pipeline of our Personalized PDD-AS2 framework. Compared to traditional FedAvg, we introduce different hard negatives to assist the model in training within the OD-AS2 scenario. As shown in Figure 4.1, we organize our training procedure as two stages adapted from some prevailing works [80, 81]: (Stage 1) ***Federated Static negative training***: we train the encoders with static hard negative sampling N^{static} under FedAvg. Due to the instability of the model in the early training stage, we initially sample BM25 negatives N^{BM25} to warm up the model following some works [80, 82]. We update both $(q;\Theta)$ and $(s;\Theta)$ by $\mathcal{L}$ defined in Eq.3.4.

(Stage 2) ***Client-customized Query Encoding***: Continual from first stage, we samples N^{fed} defined in section 4.2.1 to train a client-customized query encoder follows section 4.2.2. T

4.2.4 Retrieval Schemes

Our model is compatible with two retrieval schemes: sentence-level retrieval and passage-level retrieval. For sentence-level retrieval, we retrieve the top sentences follow the probability distribution defined in Eq.**??**. For passage-level retrieval, based on the fact that sentences are extracted from their source passages, we retrieve the passage with

highest relevance score as

$$f(p,q) := \max_{s \in p} \{ < (q;\Theta), \quad (s;\Theta) > \}, \forall s \in \mathbb{S}, \tag{4.3}$$

where $s \in p$ represents the set of sentences in a given passage p. The additional cost of sorting sentence scores can be ignored [73]. Therefore, the inference speed of our sentence-based passage retrieval is the same as for sentence retrieval.

4.3 Experiments

Baselines. We conduct experiments to compare the performance of our method with several Dense Retrieval methods, including: (1) Dense Retrieval trained with random negative [29] (2) Dense Retrieval trained with BM25 negative [30]; (3) Dense Retrieval trained with STAR [70]. In personalization stage, we compare our proposed Fed-Negative to dynamic hard-negatives in [70].(4) a simple sparse retriever constructed by BM25.

Implementation. We use pre-trained DistilBERT [78] by Hugging Face as our model. We use AdamW with a learning rate of 3e-5. We use Faiss [79] to perform the similarity search. We use open-sourced BM25 model in training. Queries and sentences are truncated to a maximum of 32 tokens and 512 tokens, respectively. We represent query embeddings simply using the $[CLS]$ token, and we represent sentence embeddings using the average pooling of word embeddings in the sentence.

In terms of datasets, we have chosen to continue using the Fed-NewsQA proposed in the previous chapter.

The details of our training procedure is described as follows: In the Federated static negative training, we pair each query with BM25 negatives and gold-negatives with a batch size of 8 in the warm-up stage. Then we replace them with static hard-negatives. To demonstrate the influence of numbers of negatives, we also experiment with settings

with different numbers of negatives. We enable in-batch negative in this stage. We implemented vanilla FedAvg as our Federated learning framework. We aggregate local weights after each epoch.

In the Client-customized Query Encoding, we pair each query with dynamic hard negatives or Fed-Negatives with a batch size of 32. To demonstrate the influence of numbers of negatives, we also experiment on settings with different numbers of negatives. We enable in-batch negatives in this stage.

We report two levels of metrics in our experiments: sentence-level and passage-level. The retrieval procedure of both levels is defined in section 4.2.4. In both levels, we report the MRR@10, Recall@1,20,100 scores.

4.3.1 Experiment Results

Table 4.1 : Results on our Fed-NewsQA Benchmark.

	Sentence-level Retrieval				**Passage-level Retrieval**			
Models	**MRR@10**	**R@1**	**R@20**	**R@100**	**MRR@10**	**R@1**	**R@20**	**R@100**
Upper Bound								
Central-training	0.338	0.284	0.629	0.781	0.502	0.447	0.553	0.821
Sparse Retriever								
BM25	0.172	0.152	0.343	0.533	0.343	0.288	0.345	0.598
Dense Retriever								
dense retrieval-Random Neg	0.194	0.171	0.466	0.62	0.376	0.323	0.401	0.702
dense retrieval-Bm25 Neg	0.188	0.151	0.475	0.639	0.353	0.303	0.388	0.679
dense retrieval-STAR	0.232	0.190	0.535	0.679	0.403	0.350	0.421	0.709
Dense Retriever: Ours								
PDD-AS2	0.261	0.217	0.546	0.695	0.429	0.395	0.479	0.745
+client-customized query encoding	0.289	0.232	0.556	0.711	0.445	0.414	0.489	0.75
+client-customized query encoding with fed-negative	**0.309**	**0.252**	**0.577**	**0.72**	**0.458**	**0.431**	**0.504**	**0.762**

The main result of our experiments is shown in Table 4.1. We conclude that our personalization method with Fed-Negative can outperform the method with local dynamic hard negatives. This is because the scarcity of training data in some clients can lead to a much worse hard-negative sampling result. Compared with static hard negative sampling,

the training of the client-customized query encoder introduces far more negative samples, strengthening the need for hard negatives in terms of quality and quantity. Our method alleviates the problem by leveraging diverse hard negatives on other clients in a privacy-preserving way.

4.3.2 Influence of Numbers of Negatives

We explore the influence of $num\_negatives$ in our setting. We experiment with the combinations of different numbers of negatives used in each method. The result of different $num\_negatives$ is shown in Table 4.2. We show the impact of $num\_negatives$ on both stages of training separately. The maximum number of hard-negatives we can test in stage 1 training is limited due to GPU RAM cost. For BM25 negative sampling and static hard-negative sampling, we train the model with our PDD-AS2 framework from the beginning of our training procedure. In experiments of stage 2 training with Fed-Negative, we continue our training from the model weights trained in previous steps, which follows our training procedure.

We found client-customized query encoder can be steadily improved while feeding much more negatives compared with stage 1 training. This result indicates the need for introducing more hard-negatives with higher quality in stage 2 training, further proving the effectiveness and necessity of our Fed-Negative. What's more, the computational cost does not scale with the $num\_negatives$. As a consequence, client-customized query encoder can benefit from Fed-Negative with little cost.

4.3.3 Privacy

When transferring sentence embeddings between clients, one key concern is whether the user's privacy would be leaked. However, no work has been dedicated to restoring private information from mere sentence embeddings. In order to measure the risk involved, we conducted an experiment to detect whether our transmitted sentence embeddings con-

Table 4.2 : Different num_negative in Training

Models	Sentence-level Retrieval				Passage-level Retrieval			
	MRR@10	**R@1**	**R@20**	**R@100**	**MRR@10**	**R@1**	**R@20**	**R@100**
Dense Retriever with BM25 negatives								
num_negative=2	0.143	0.123	0.302	0.489	0.310	0.247	0.311	0.582
num_negative=8	0.172	0.151	0.343	0.533	0.343	0.288	0.345	0.598
Dense Retriever with STAR								
num_negative=2	0.201	0.160	0.506	0.655	0.352	0.305	0.379	0.705
num_negative=8	0.232	0.191	0.535	0.679	0.403	0.350	0.421	0.709
PDD-AS2								
num_negative=2	0.242	0.193	0.516	0.645	0.392	0.354	0.432	0.719
num_negative=8	0.261	0.217	0.546	0.695	0.429	0.395	0.479	0.745
+client-customized query encoding								
num_negative=10	0.272	0.233	0.557	0.705	0.431	0.415	0.487	0.746
num_negative=200	**0.289**	**0.251**	**0.576**	**0.711**	**0.445**	**0.434**	**0.489**	**0.75**

Method	**Perplexity**
Without training	36.3
CLM without embedding	25.9
CLM with sentence embedding	**25.6**

Table 4.3 : Perplexity of gpt-2 on our dataset

tained information related to the original text.

In this experiment, we used GPT-2, a model that performs well on text generation tasks. In the first part of the experiment, we trained GPT-2 on the language modeling task using our dataset and measured its perplexity on the test set. In the second part of the experiment, we added the sentence embeddings generated by the previously trained sentence encoder in PDD-AS2 to the training and testing procedure. In detail, we feed the sentence embeddings into the GPT-2 as key-value pairs together with the text input. After receiving the input, the model tries to establish the connection between the embedding and the actual sentence it represents through the self-attention structure. Table 4.3 shows

no significant difference in the perplexity between the two groups of experiments. The group with sentence embeddings has slightly lower perplexity on the test set. However, these differences are not statistically significant. To further demonstrate that we cannot obtain private information from the sentence embeddings, we let GPT-2 generate actual sentences directly from their corresponding embeddings without any input and prompts. We show the result in the Table 4.3.3.

We found that GPT-2 could not restore the actual sentence using only the sentence embeddings. Sentence embeddings did have an impact on the generated results. However, these effects are seemingly random and irrelevant to the actual sentence.

Table 4.4 : Case study of sentence-embeddings decoding

Original Sentences	Generated Sentences
Four Australian troops have now died in the conflict in Afghanistan.	”It’s not the first time that we’ve had
It made my stomach turn,” Bertha Lewis, chief executive officer of ACORN, told reporters at the National Press Club in Washington.	”I think it’s important? very important? Very difficult to the one. I think. is, part of me. I the to blame, I don’t blame my
Read the story at the WRTV web site	CNN’s a great-school program that’s not

4.4 Chapter Summary

In this chapter, we propose a personalization method to train a client-customized query encoder for each client. We also propose a new negative sampling method called Fed-Negative. In Fed-Negative, we introduce diverse negatives from other clients to enhance

the training. We further test our method on a new Federated OD-AS2 based on NewsQA. This benchmark better mimics real-world cases than other benchmarks in terms of data distribution and query types.

The experiment results show that our method can effectively improve the performance of OD-AS2 under distributed settings by personalization.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

In this thesis, we deeply studied how to solve distributed OD-AS2 with privacy preserving. Moreover, we propose a personalization method to alleviate the problem occur in our proposed distributed learning framework.

- In Chapter 3, we propose a privacy-preserving distributed OD-AS2 framework called PDD-AS2. We conduct extensive experiments on our benchmark to demonstrate that PDD-AS2 can levearge the training data on each local device. Moreover, we also explored the performance of the framework in scenarios with scarce data. The experiments proved that our framework can effectively improve performance in data-scarce scenarios. We also explored issues exposed by the framework in the experiments, such as the performance decline in some clients after using PDD-AS2, which proved the necessity for further work to resolve this problem.

- In Chapter 4, we first propose a personalized method called Client-customized Query Encoding to train a personalized query encoder. Second, we introduce a new negative sampling method: Fed-Negative to further enhance the effectiveness of our approach. This sampling method allows local models to acquire embeddings from other clients. In view of potential privacy issues this method may raise, we also conducted research. Experiments show that our proposed framework significantly enhances the performance of the distributed OD-AS2 framework, without compromising privacy.

5.2 Limitations

In this section, I will discuss the limitations of our proposed work. Firstly, to simulate different topics and data distributions among clients in real-world scenarios, I faced a scarcity of datasets that meet our standards, hence I only conducted tests based on a single benchmark. This suggests that my conclusions may not hold on other datasets and benchmarks. Secondly, although my proposed fed-negative method improves model performance, I did not measure the network overhead, and there is a possibility that this method could be constrained by communication costs. Lastly, due to limitations in machine costs and performance overhead, we only used a small model, distill-bert, for my experiments, and simulated a scenario with only 10 clients. However, in real scenarios, up to millions of clients might participate in training, which could lead to a dramatic increase in network communication costs and a discrepancy between the data distribution in my experiments and actual conditions, potentially causing biases in my results.

5.3 Future Work

In the this section, we will discuss some potential future works to better improve our method which introduced in before chapters.

For PDD-AS2, the main future work will focus on validating its performance on more datasets and evaluation methods that conform to real-world scenarios. Although we have proposed and built a benchmark that fits real-world scenarios in our thesis, in order to demonstrate the universality of our method, we need to construct more benchmarks to test the performance of PDD-AS2 in similar distributed scenarios. Meanwhile, regarding the query hubness problem we have identified, more work is needed to address it.

One of the future tasks for Client-Customized Query Encoding and Fed-Negative is to clarify the communication overhead of our proposed methods. If embedding transmission is carried out between any two clients, despite the small amount of data transferred

each time, if the number of clients joined in the training is enormous, such as millions, it will cause significant network overhead. If only a portion of the clients are chosen in the embedding exchange, performance will inevitably be affected. We need to find a way to balance communication overhead and performance. Secondly, concerning privacy, we did not conduct a very detailed study of our methods. In this thesis, we only investigated whether our proposed methods would introduce additional privacy issues. However, there is considerable evidence that federated learning can pose privacy leakage risks when applied to NLP tasks, so related research is essential.

Bibliography

[1] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.

[2] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[3] Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Matthew Kelcey, Jacob Devlin, Kenton Lee, Kristina N. Toutanova, Llion Jones, Ming-Wei Chang, Andrew Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. Natural questions: a benchmark for question answering research. *Transactions of the Association of Computational Linguistics*, 2019.

[4] Tri Nguyen, Mir Rosenberg, Xia Song, Jianfeng Gao, Saurabh Tiwary, Rangan Majumder, and Li Deng. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268*, 2016.

[5] Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2020.

[6] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang.

Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, 2020.

[7] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[8] Gehui Shen, Yunlun Yang, and Zhi-Hong Deng. Inter-weighted alignment network for sentence pair modeling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 2017.

[9] Quan Hung Tran, Tuan Lai, Gholamreza Haffari, Ingrid Zukerman, Trung Bui, and Hung Bui. The context-dependent additive recurrent neural net. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[10] Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. A compare-aggregate model with latent clustering for answer selection. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[11] Siddhant Garg, Thuy Vu, and Alessandro Moschitti. Tanda: Transfer and adapt pre-trained transformer models for answer sentence selection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.

[12] Ali Mohamed Nabil Allam and Mohamed Hassan Haggag. The question answering systems : A survey . 2016.

[13] Sanda M. Harabagiu, Steven J. Maiorano, and Marius Pasca. Open-domain textual question answering techniques. *Natural Language Engineering*, 2003.

[14] Marius Paca. Open-domain question answering from large text collections. *Computational Linguistics*, 2003.

[15] Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: Bm25 and beyond. *Found. Trends Inf. Retr.*, 2009.

[16] Juan Ramos. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, 2003.

[17] Julian Kupiec. Murax: a robust linguistic approach for question answering using an on-line encyclopedia. In *SIGIR*, 1993.

[18] Zhiping Zheng. Answerbus question answering system. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, 2002.

[19] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[20] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach. *ArXiv preprint arXiv:1907.11692*, 2019.

[21] Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. Reading wikipedia to answer open-domain questions. *ArXiv preprint arXiv:1704.00051*, 2017.

[22] Yixin Nie, Songhe Wang, and Mohit Bansal. Revealing the importance of semantic retrieval for machine reading at scale. In *EMNLP*, 2019.

[23] Kyosuke Nishida, Itsumi Saito, Atsushi Otsuka, Hisako Asano, and Junji Tomita. Retrieve-and-read: Multi-task learning of information retrieval and reading comprehension. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2018.

[24] Yair Feldman and Ran El-Yaniv. Multi-hop paragraph retrieval for open-domain question answering. *arXiv preprint arXiv:1906.06606*, 2019.

[25] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. *arXiv preprint arXiv:2002.08909*, 2020.

[26] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Yu Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*, 2020.

[27] Yankai Lin, Haozhe Ji, Zhiyuan Liu, and Maosong Sun. Denoising distantly supervised open-domain question answering. In *ACL*, 2018.

[28] Shuohang Wang, Mo Yu, Xiaoxiao Guo, Zhiguo Wang, Tim Klinger, Wei Zhang, Shiyu Chang, Gerald Tesauro, Bowen Zhou, and Jing Jiang. R3: Reinforced ranker-reader for open-domain question answering. In *AAAI*, 2018.

[29] Jui-Ting Huang, Ashish Sharma, Shuying Sun, Li Xia, David Zhang, Philip Pronin, Janani Padmanabhan, Giuseppe Ottaviano, and Linjun Yang. Embedding-based retrieval in facebook search. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020.

[30] Luyu Gao, Zhuyun Dai, Tongfei Chen, Zhen Fan, Benjamin Van Durme, and Jamie Callan. Complement lexical retrieval model with semantic residual embeddings. In *Advances in Information Retrieval*, 2021.

[31] Shuqi Lu, Di He, Chenyan Xiong, Guolin Ke, Waleed Malik, Zhicheng Dou, Paul Bennett, Tie-Yan Liu, and Arnold Overwijk. Less is more: Pretrain a strong Siamese encoder for dense text retrieval using a weak decoder. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, 2021.

[32] Yingqi Qu, Yuchen Ding, Jing Liu, Kai Liu, Ruiyang Ren, Wayne Xin Zhao, Daxiang Dong, Hua Wu, and Haifeng Wang. RocketQA: An optimized training

approach to dense passage retrieval for open-domain question answering. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021.

[33] Daliang Li and Junpu Wang. Fedmd: Heterogenous federated learning via model distillation. *ArXiv preprint arXiv:1910.03581*, 2019.

[34] Peter Kairouz, H. B. McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Keith Bonawitz, Zachary B. Charles, Graham Cormode, Rachel Cummings, Rafael G. L. D'Oliveira, Salim Y. El Rouayheb, David Evans, Josh Gardner, Zachary Garrett, Adrià Gascón, Badih Ghazi, Phillip B. Gibbons, Marco Gruteser, Zaïd Harchaoui, Chaoyang He, Lie He, Zhouyuan Huo, Ben Hutchinson, Justin Hsu, Martin Jaggi, Tara Javidi, Gauri Joshi, Mikhail Khodak, Jakub Konecný, Aleksandra Korolova, Farinaz Koushanfar, Oluwasanmi Koyejo, Tancrède Lepoint, Yang Liu, Prateek Mittal, Mehryar Mohri, Richard Nock, Ayfer Özgür, R. Pagh, Mariana Raykova, Hang Qi, Daniel Ramage, Ramesh Raskar, Dawn Xiaodong Song, Weikang Song, Sebastian U. Stich, Ziteng Sun, Ananda Theertha Suresh, Florian Tramèr, Praneeth Vepakomma, Jianyu Wang, Li Xiong, Zheng Xu, Qiang Yang, Felix X. Yu, Han Yu, and Sen Zhao. Advances and open problems in federated learning. *Found. Trends Mach. Learn.*, 2021.

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

[36] Mingqing Chen, Rajiv Mathews, Tom Y. Ouyang, and Françoise Beaufays. Federated learning of out-of-vocabulary words. *ArXiv preprint arXiv:1903.10635*, 2019.

[37] Shaoxiong Ji, Shirui Pan, Guodong Long, Xue Li, Jing Jiang, and Zi Huang. Learning private neural language modeling with attentive aggregation. 2019.

[38] Swaroop Indra Ramaswamy, Rajiv Mathews, Kanishka Rao, and Franccoise Beaufays. Federated learning for emoji prediction in a mobile keyboard. *ArXiv preprint arXiv:1906.04329*, 2019.

[39] Timothy Yang, Galen Andrew, Hubert Eichner, Haicheng Sun, Wei Li, Nicholas Kong, Daniel Ramage, and Françoise Beaufays. Applied federated learning: Improving google keyboard query suggestions. *arXiv preprint arXiv:1812.02903*, 2018.

[40] Suyu Ge, Fangzhao Wu, Chuhan Wu, Tao Qi, Yongfeng Huang, and Xing Xie. Fedner: Privacy-preserving medical named entity recognition with federated learning. *arXiv preprint arXiv:2003.09288*, 2020.

[41] Xinghua Zhu, Jianzong Wang, Zhenhou Hong, and Jing Xiao. Empirical studies of institutional federated learning for natural language processing. In *FINDINGS*, 2020.

[42] Filip Hanzely and Peter Richtárik. Federated learning of a mixture of global and local models. *ArXiv preprint arXiv:2002.05516*, 2020.

[43] Tao Yu, Eugene Bagdasaryan, and Vitaly Shmatikov. Salvaging federated learning by local adaptation. *ArXiv preprint arXiv:2002.04758*, 2020.

[44] Kangkang Wang, Rajiv Mathews, Chloé Kiddon, Hubert Eichner, Franccoise Beaufays, and Daniel Ramage. Federated evaluation of on-device personalization. *ArXiv preprint arXiv:1910.10252*, 2019.

[45] Lorien Y. Pratt. Discriminability-based transfer between neural networks. In *NIPS*, 1992.

[46] Yihan Jiang, Jakub Konecný, Keith Rush, and Sreeram Kannan. Improving federated learning personalization via model agnostic meta learning. *ArXiv preprint arXiv:1909.12488*, 2019.

[47] Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. Distilling the knowledge in a neural network. *ArXiv preprint arXiv:1503.02531*, 2015.

[48] Jieren Deng, Yijue Wang, Ji Li, Chenghong Wang, Chao Shang, Hang Liu, Sanguthevar Rajasekaran, and Caiwen Ding. Tag: Gradient attack on transformer-based language models. In *EMNLP*, 2021.

[49] Dimitar I. Dimitrov, Mislav Balunovi'c, Nikola Jovanovi'c, and Martin T. Vechev. Lamp: Extracting text from gradients with language model priors. *arXiv preprint arXiv:2202.08827*, 2022.

[50] Liam Fowl, Jonas Geiping, Steven Reich, Yuxin Wen, Wojtek Czaja, Micah Goldblum, and Tom Goldstein. Decepticons: Corrupted transformers breach privacy in federated learning for language models. *ArXiv preprint arXiv:2201.12675*, 2022.

[51] Samyak Gupta, Yangsibo Huang, Zexuan Zhong, Tianyu Gao, Kai Li, and Danqi Chen. Recovering private text in federated learning of language models. *ArXiv preprint arXiv:2205.08514*, 2022.

[52] Cody C. T. Kwok, Oren Etzioni, and Daniel S. Weld. Scaling question answering to the web. *ACM Trans. Inf. Syst.*, 2001.

[53] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*, 2020.

[54] Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. Latent retrieval for weakly supervised open domain question answering. *arXiv preprint arXiv:1906.00300*, 2019.

[55] Wen tau Yih, Kristina Toutanova, John C. Platt, and Christopher Meek. Learning discriminative projections for text similarity measures. In *CoNLL*, 2011.

[56] Daniel Gillick, Sayali Kulkarni, Larry Lansing, Alessandro Presta, Jason Baldridge, Eugene Ie, and Diego Garcia-Olano. Learning dense representations for entity retrieval. *arXiv preprint arXiv:1909.10506*, 2019.

[57] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, M. Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[58] Lee Xiong, Chenyan Xiong, Ye Li, Kwok-Fung Tang, Jialin Liu, Paul Bennett, Junaid Ahmed, and Arnold Overwijk. Approximate nearest neighbor negative contrastive learning for dense text retrieval. *arXiv preprint arXiv:2007.00808*, 2021.

[59] Mengqiu Wang, Noah A Smith, and Teruko Mitamura. What is the jeopardy model? a quasi-synchronous grammar for qa. In *Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL)*, 2007.

[60] Hua He and Jimmy Lin. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies*, 2016.

[61] Liu Yang, Qingyao Ai, Jiafeng Guo, and W Bruce Croft. anmm: Ranking short answer texts with attention-based neural matching model. In *Proceedings of the 25th ACM international on conference on information and knowledge management*, 2016.

[62] Shuohang Wang and Jing Jiang. A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*, 2016.

[63] Luca Soldaini and Alessandro Moschitti. The cascade transformer: an application for efficient answer sentence selection. In *ACL*, 2020.

[64] H. B. McMahan, Eider Moore, Daniel Ramage, and Blaise Agüera y Arcas. Federated learning of deep networks using model averaging. *arXiv preprint arXiv:1602.05629*, 2016.

[65] H. B. McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, 2017.

[66] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, and Jürgen Schmidhuber. Lstm: A search space odyssey. 2017.

[67] David Leroy, Alice Coucke, Thibaut Lavril, Thibault Gisselbrecht, and Joseph Dureau. Federated learning for keyword spotting. 2019.

[68] Dhruv Guliani, Françoise Beaufays, and Giovanni Motta. Training speech recognition models with federated learning: A quality/cost framework. 2021.

[69] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *ArXiv preprint arXiv:1803.02999*, 2018.

[70] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[71] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-Efficient Learning of Deep Networks from Decentralized Data. In *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, 2017.

[72] Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022.

[73] Jinhyuk Lee, Mujeen Sung, Jaewoo Kang, and Danqi Chen. Learning dense representations of phrases at scale. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2021.

[74] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

[75] Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. Semantic parsing on Freebase from question-answer pairs. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 2013.

[76] Steven Bird, Ewan Klein, and Edward Loper. *Natural language processing with Python: analyzing text with the natural language toolkit*. 2009.

[77] Adam Trischler, Tong Wang, Xingdi Yuan, Justin Harris, Alessandro Sordoni, Philip Bachman, and Kaheer Suleman. Newsqa: A machine comprehension dataset. In *Proceedings of the 2nd Workshop on Representation Learning for NLP*, 2017.

[78] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*, 2019.

[79] J. Johnson, M. Douze, and H. Jegou. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*, 2021.

[80] Jingtao Zhan, Jiaxin Mao, Yiqun Liu, Jiafeng Guo, Min Zhang, and Shaoping Ma. Optimizing dense retrieval model training with hard negatives. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2021.

[81] Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. *ArXiv preprint arXiv:2004.04906*, 2020.

[82] Luyu Gao and Jamie Callan. Unsupervised corpus aware language model pre-training for dense passage retrieval. *arXiv preprint arXiv:2108.05540*, 2021.