applied sciences

MDPI

Article

FedQAS: Privacy-Aware Machine Reading Comprehension with Federated Learning

Addi Ait-Mlouk *, Sadi A. Alawadi, Salman Toor and Andreas Hellander

Department of Information Technology, Division of Scientific Computing, Uppsala University, 75236 Uppsala, Sweden; sadi.alawadi@it.uu.se (S.A.A.); salman.toor@it.uu.se (S.T.); andreas.hellander@it.uu.se (A.H.)

* Correspondence: addi.ait-mlouk@it.uu.se

Abstract: Machine reading comprehension (MRC) of text data is a challenging task in Natural Language Processing (NLP), with a lot of ongoing research fueled by the release of the Stanford Question Answering Dataset (SQuAD) and Conversational Question Answering (CoQA). It is considered to be an effort to teach computers how to "understand" a text, and then to be able to answer questions about it using deep learning. However, until now, large-scale training on private text data and knowledge sharing has been missing for this NLP task. Hence, we present FedQAS, a privacy-preserving machine reading system capable of leveraging large-scale private data without the need to pool those datasets in a central location. The proposed approach combines transformer models and federated learning technologies. The system is developed using the FEDn framework and deployed as a proof-of-concept alliance initiative. FedQAS is flexible, language-agnostic, and allows intuitive participation and execution of local model training. In addition, we present the architecture and implementation of the system, as well as provide a reference evaluation based on the SQuAD dataset, to showcase how it overcomes data privacy issues and enables knowledge sharing between alliance members in a Federated learning setting.

Keywords: machine reading comprehension; natural language processing; question answering; data privacy; federated learning; transformer

Citation: Ait-Mlouk, A.; Alawadi, S.; Toor, S.; Hellander, A. FedQAS: Privacy-Aware Machine Reading Comprehension with Federated Learning. *Appl. Sci.* **2022**, *12*, 3130. https://doi.org/10.3390/app12063130

Academic Editors: Arturo Montejo-Ráez and Salud María Jiménez-Zafra

Received: 25 January 2022
Accepted: 14 March 2022
Published: 18 March 2022

1. Introduction

Machine reading comprehension (MRC) is a sub-field of natural language understanding (NLU) that aims to teach machines to read and understand human languages (text). A user can ask the machine to answer questions based on a given paragraph or text document. Generally, MRC requires modeling complex interactions between the context and the query in a specific domain. It could be used in many NLP applications such as dialogue systems and search engines as shown in Figure 1—a Google search engine with MRC techniques can directly return the correct answers to questions rather than a list of content and web pages. These kinds of techniques have been based on hand-crafted rules that need substantial human effort and resources. However, with the rise of artificial intelligence, there has been an explosion of various MRC benchmark datasets and models that contribute to a better understanding of the task and show their ability to exceed human performance. Despite this rapid progress on MRC datasets and models, most of the existing work has focused on algorithms for improving model performance.

At present, several MRC models have already surpassed human performance on many of the MRC datasets [1], but there is still a limit in terms of data availability due to privacy concerns, collaborative training, resource consumption, and communication overhead due to data transfer. Hence, there is a need for extending existing MRC models in a way that keeps data private on its generated location and allows several participants to train a machine learning model by sharing only model parameters. This will let cross-silo

(companies) or cross-device (phones, IoT devices) participate in the training process and let the model learn from large distributed datasets by sharing knowledge between different local models. To address these gaps, we proposed a privacy-aware approach based on federated learning to learn new global models in a geographically distributed manner using our FEDn framework [2], build more challenging MRC models by integrating with private data generation and labeling for local accurate training as well as an incremental learning approach to strengthening the model performances during collaborative training without compromising the data.

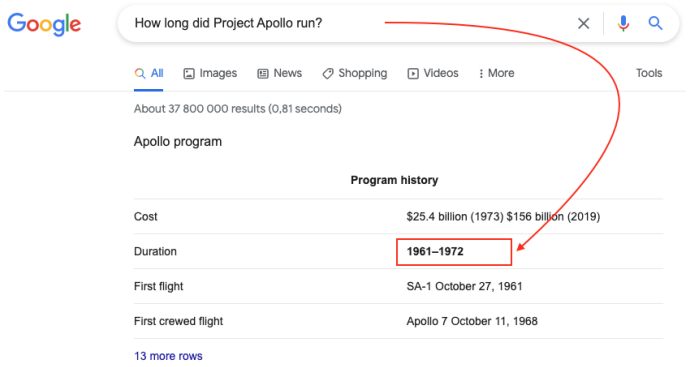

Figure 1. An example of Google search engine with machine reading comprehension techniques.

The remainder of this paper is organized as follows. Section 2 surveys related work. Section 3 details the proposed approach and architecture of FedQAS, with an emphasis on its privacy and scalability properties. In Section 4, we demonstrate the frameworks potential in an evaluation based on the SQuAD dataset. Finally, Section 4 concludes the work and outlines future work.

2. Related Work

Machine reading comprehension was proposed for the first time in 1977 by Lehnert, who built a question answering program called the QUALM [3]. In 1999, Hirschman et al. [4] built a reading comprehension system using a corpus of 60 development and 60 test stories of 3rd to 6th grade material. Because of the lack of benchmark datasets in that period, most MRC systems were rule-based or statistical models [5,6]. In recent years, many benchmark datasets have been released and focused on MRC by answering questions, see Table 1. Since these datasets were made available, there has been considerable progress on MRC tasks. The Stanford Question Answering Dataset (SQuAD) is one of the most well-known reading comprehension datasets, consisting of 100,000 questions posed by crowd-workers on Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading context. Important progress based on SQuAD concerns include the attention method [7] and Bi-Directional Attention Flow (BiDAF) [8], which considerably improved the question answering performance. These two methods compute Context to Question attention and Question to Context attention using a similarity matrix computed directly from context and question. Authors in [9] describe a novel hierarchical attention network for reading comprehension style question answering, which aims to answer questions for a given narrative paragraph. In their work, attention and fusion are conducted horizontally and vertically across layers at different levels of granularity between question and paragraph. In recent work in language modeling, authors in [10] incorporate explicit contextual semantics from pre-trained semantic role labeling and introduce an improved language representation model, Semantics-aware BERT (SemBERT), which is capable of explicitly absorbing contextual semantics over a BERT backbone. Moreover, Zhuosheng et al. [11] propose using syntax to guide the text modeling by incorporating explicit syntactic constraints into the attention mechanism for better linguistically motivated word representations. Recently, there has been an

explosion of various MRC benchmark datasets that leads to a variety of models such as BiDAF [12] and other models based on BERT [13], RoBERTa [14], XLNet [15], ELMo [16] and transformer [17]. Other relevant works have been proposed, including [18–20].

Table 1. List of some existing MRC datasets.

Dataset	Answer Type	Domain-Specific
MCTest [8]	Multiple choice	Children's stories
CNN/Daily Mail [21]	Spans	News
Children's book [22]	Spans	Children's stories
MS MARCO [23]	Free-form text	Web Search
NewsQA [24]	Spans	News
SearchQA [25]	Spans	Jeopardy
TriviaQA [26]	Spans	Trivia
SQuAD [27]	Spans	Wikipedia
SQuAD 2.0 [28]	Spans, Unanswerable	Wikipedia
CoQA [29]	Free-form text,	News, Reddit Wikipedia

All these proposed approaches required a very large amount of data for training, which is not always available in some cases, in particular when the text data is sensitive, private (medical text, business, social media), and very big. In this context, we here propose the use of federated learning as a method for distributed and collaborative machine learning. Organizations maintain and govern their data locally and participate in learning a new global, federated model by sending only their model updates (model weights) to a server for aggregation into the global model. Hence, all participants (clients) can benefit from a newly trained model without exposing their data publicly.

Our contributions in this paper can be summarized as follows: (1) We propose federated learning models for MRC using a transformer architecture, (2) we design and develop the FedQAS system for collaborative training, (3) we preserve data privacy (4) we improve the local training with incremental learning scheme and private data generation, and (5) our analyses of the models respect data privacy regulations and outperforms the baseline model on SQuAD after a couple of rounds.

3. Proposed Approach

The overall architecture of our proposed FedQAS system is shown in Figure 2. The main modules are private data pipeline, federated learning settings, question answering, and incremental learning. The private data pipeline module allows local users (clients) to process and prepare their data locally to be used by federated learning methods. The federated learning module enables multiple private clients to form an alliance to collaboratively train machine learning/deep learning models and send parameters to the server for global model generation (aggregation of local models). Afterward, the system allows the client to add new data locally and train the model incrementally through a defined number of rounds to improve the performance using incremental learning techniques. Finally, participating clients can use the global model for question answering system. The system is implemented using the FEDn federated learning framework [2] and a web interface using Flask (https://flask.palletsprojects.com/, accessed on 24 January 2022). FEDn provides a highly scalable federated learning run-time, and Flask is used to develop interactive and user-friendly interfaces for the different processes in the workflow. The list of available datasets related to question answering used for the demo is placed in the local data sources. Moreover, the developed system is scalable, flexible, and can be expanded with new clients/data sets on-demand (without the need to re-train the federated model).

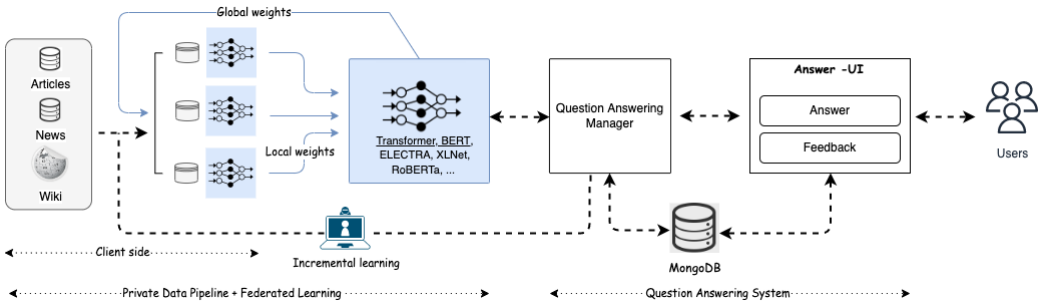

Figure 2. Overview of the FedQAS architecture. The FedQAS approach is organized in three main logical layers, the first one is for collaborative privacy-preserving training, the second one is a federated question answering manager, and the third is for incremental learning and private data generation.

3.1. Data Processing (Client Side)

Stanford Question Answering Dataset (SQuAD) [30] is a machine reading comprehension dataset, consisting of questions posed by crowd-workers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, alternatively the question might be unanswerable. SQuAD2.0 combines the 100.000 questions in SQuAD1.1 with over 50,000 unanswerable questions written adversarially by crowd-workers to look similar to answerable ones. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering (https://rajpurkar.github.io/SQuAD-explorer/, accessed on 24 January 2022), see Figure 3 for an example of passage, questions, and answers. Consider the question "How many countries does Shell operate in?" posed in the passage. To answer the question, one might first locate the relevant part of the passage "It has operations in over 90 countries", then reason that "under" refers to a cause (not location), and thus determine the correct answer: "over 90".

Shell was vertically integrated and is active in every area of the oil and gas industry, including exploration and production, refining, distribution and marketing, petrochemicals, power generation and trading. It has minor renewable energy activities in the form of biofuels and **wind**. It has operations in **over 90** countries, produces around 3.1 million barrels of oil equivalent per day and has **44,000 service stations** worldwide. Shell Oil Company, its subsidiary in the United States, is one of its largest businesses.

Question 1: Aside from biofuels what other renewable energy activities is Shell involved with? wind
Question 2: How many countries does Shell operate in? over 90
Question 3: How many services stations does Shell have? 44,000 service stations

Figure 3. A paragraph from Wikipedia and three associated questions together with their answers, taken from the SQuAD dataset.

3.2. Private Data Pipeline Module

To train a model with a high level of accuracy, machine reading comprehension models require large datasets to 'learn' from, however, data might be sensitive and private. To preserve data privacy, different anonymization techniques have been used. The most relevant are k-anonymity [31], l-diversity [32], and t-closeness [33]. In k-anonymity, specific columns (e.g., name, religion, sex) are removed or altered (e.g., replacing a specific age with an age span). L-diversity and t-closeness are extensions of k-anonymity, which are used to protect attribute disclosure, these anonymization techniques are applied before data is shared for training. However, with the rise of AI, this form of anonymizing personal data is not enough to protect privacy because the data can often be reverse-engineered using machine learning to re-identify individuals [34]. In question answering systems, there might be sensitive documents, personal data that needs to be processed for MRC task, without exposing data. To handle this issue, we propose a question answering

system based on federated learning methodology to protect data leakage and ensure secure collaborative training. The proposed system follows a federated learning paradigm in which participating clients are required to train their local models and then share the gradient (model parameters) for an eventual aggregation strategy in a central server. This approach ensures input data privacy, enables collaborative training, low-cost training by distributing the workload across clients instead of training a large model individually, sharing the local learning model within the alliance (training clients) without compromising private data, and improving local learning by using incremental learning and local data generation pipeline.

3.3. Federated Machine Learning Module

Federated learning is an emerging technology enabling multiple parties to jointly train machine learning models on private data. These parties could be mobile and IoT devices (cross-device FL), or organizations (cross-silo). Data remain locally at each party, only the parameter updates are communicated with a server and other parties. In our system, we use FL to develop FedQAS based on transformer architecture for question answering. FedQAS trains a global model (Algorithm 1) on large amounts of data from multiple geographically distributed parties. Each party trains a local transformer model on its data (Algorithm 2) and sends parameters W_t to the central server for aggregation (FedAVG [35]) instead of the whole model. In the aggregation part, the aggregator (running in the *combiner* in FEDn [2]) combines parameters and generates a single global model $M(W_t)$ for each round using federated incremental averaging [35].

Algorithm 1: Incremental FedAVG algorithm. **k**: Number of clients, **r**: Number of rounds, W_i: Local model weights and **M**: Global model weights.

Input: W_t
Output: $M(W_t)$

Server executes:
initialized W_0
Function `IncrementealFedAVG`(k, W_{t-1}, W_t)**:**
  foreach $t \leftarrow 1$ **to** r **do**
    $S_t \leftarrow$ (sample a random set of clients)
    foreach *client* $k \in S_t$ ***in parallel*** **do**
      $W_{t+1}^k \leftarrow ClientUpdate(k, W_t, N_l)$
      $W_{t+1} \leftarrow \sum_{k=1}^{k} \frac{n_k}{n} W_{t+1}^k$
    end
    $W_t \leftarrow (W_{t-1} + (W_t - W_{t-1})/t)$
  end
  return $M(W_t)$

Algorithm 2: Local client update, **k**: Number of clients, $\boldsymbol{D^k}$: Client k local dataset, **e**: Number of local epochs, and η is the learning rate.

Output: W_t
// Run on client k
Function `ClientUpdate`(k, W_t)**:**
  $\beta \leftarrow$ (split D^k into mini batches)
  for *local epoch* $e_i \in 1, \ldots e$ **do**
    for *batch* $b \in \beta$ **do**
      $W_t \leftarrow W_t - \eta \nabla l(W_t, b)$
    end
  end
  return W_t

3.4. Transformer Model

In this paper, we develop a FedQAS using a Transformer architecture [17] to handle questions. The transformer is a model architecture eschewing recurrence and instead relying on an attention mechanism to draw global dependencies between input and output. The transformer follows the encoder and decoder architecture using stacked self-attention. The encoder maps an input sequence of symbol representations $(x_1, ..., x_n)$ to a sequence of continuous representations $z = (z_1, ..., z_n)$. The decoder then generates an output sequence $(y_1, ..., y_m)$ of symbols. Both encoder and decoder are composed of layers and sub-layers that can be stacked on top of each other multiple times. The first is a multi-head self-attention mechanism and the second is a simple, position-wise fully connected feed-forward network. In this paper, by applying the self-attention mechanism, we aim at capturing the long dependencies in the input sentence, the inputs and outputs are first embedded into an n-dimensional space.

3.5. Incremental Learning Module

Incremental learning is a machine learning case in which input data is continuously used to extend the existing model's knowledge, i.e., to further train the model. It attempts to improve a model's performance while adding the fewest samples possible. In the proposed system, adding data locally by clients is an important task to improve the local model performance first, then propagating these improvements into the global model after new training rounds in a privacy-preserving manner. We have engineered an intuitive process for each local client to contribute to the adding of new samples on top of their local data (Figure 4). The first step is to add a new data point that will remain on the local site, this allows the user to add their private data, questions, and correct answers. The incremental learning module will process and transform the private data locally and generate training points to be used in the local training. This process enables collaborative data generation between organizations in a private way in order to strengthen data protection and avoid unnecessary sharing within the alliance. In addition, a database layer is used to store user queries and global model predictions as feedback to enhance and improve the performance for further usage.

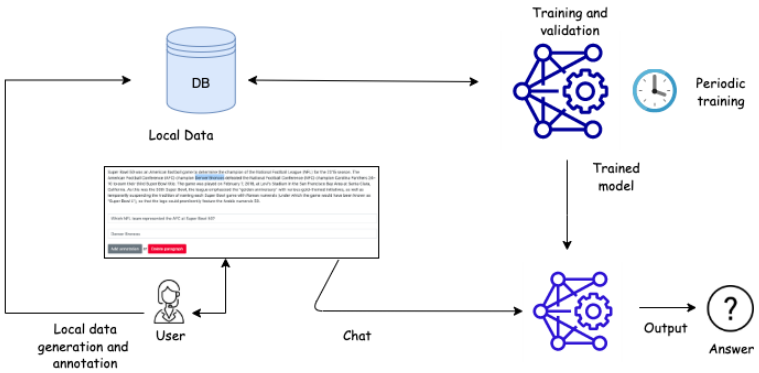

Figure 4. Federated incremental learning process. Clients can add data continuously to extend the existing model's knowledge locally while training a global model, the generated data can be stored locally on the client-side.

4. Experiment and Results

SQuAD is a reading comprehension dataset made up of questions posed by crowd workers on a collection of high-quality Wikipedia articles. It covers a wide range of topics from music celebrities to abstract notions. When comparing SQuAD with other datasets, SQuAD is one of the most popular question answering datasets (it's been cited over 4096 times) because it's well-created and improves on many aspects that other datasets fail to address. Other reading comprehension datasets such as MCTest [8] and Deep Read [4] are too small to support intensive and complex models. Hence, we conduct our

experiment on the SQuAD 1.0 dataset, which contains 100.000+ question-answer pairs. To ensure collaborative training, we randomly select and split data over 5 clients with 20% for validation dataset for all clients. We used FedAVG [35] for the aggregation of model parameters, see Table 2.

Table 2. Federated training configuration.

Rounds	Total Number of Clients	Update Size	Total Number of Parameters
5	5	400 MB	109.483.776

For the fine-tuning in our task, we used the BERT base as an encoder to build our model and the implementations are based on the public TensorFlow implementation from Keras (https://github.com/tensorflow/tensorflow, accessed on 24 January 2022) we set the initial learning rate to 5×10^{-5}. The batch size is set to 8. The maximum number of epochs is set to 1. Texts are tokenized using Wordpieces [36] with a maximum length of 384. Table 3 presents the hyperparameters used in our experiments. We used three input layers, two dense layers, two flatten layers, and Adam optimizer. We run the model for optimizing the cross-entropy loss between the output probabilities and the output answers.

Table 3. Experimental model parameters.

Hyper-Parameter	Range	Value
Epochs	[1–3]	1
Batch size	[8–128]	8
Learning rate	[0.001–0.004]	5×10^{-5}
Optim. method	Adam, SGD, RMSProp	Adam
MAX_SEQ_LENGTH	[1–1000]	384

4.1. Framework Evaluation

For the evaluation, we used exact match (EM) and F1 score, the main metrics commonly used for question answering systems. These metrics are computed on individual (question, answer) pairs. In case of multiple correct answers for a given question, the maximum score over all possible correct answers is computed. In the EM metric, for each pair (question, answer), if the characters of the model's prediction exactly match the characters of (one of) the True Answer(s), EM = 1, otherwise EM = 0.

The Accuracy represents the percentage of the questions that an MRC system accurately answers. Each question corresponds to one correct answer. For the span prediction task, the accuracy is the same as Exact Match and can be computed by the Formula (1) as follows:

$$Accuracy = EM = \frac{Number\ of\ correct\ answers}{Number\ of\ questions} \quad (1)$$

The precision represents the percentage of token overlap between the tokens in the correct answer and the tokens in the predicted answer, while the recall is the percentage of tokens in a correct answer that have been correctly predicted in a question. The True Positive (TP) denotes the same tokens between the predicted answer and the correct answer, the False Positive (FP) denotes the tokens which are not in the correct answer but the predicted answer, while the False Negative (FN) presents the tokens that are not in the predicted answer but the correct answer. Precision and Recall can be computed by the Formulas (2) and (3) as follows:

$$Precision = \frac{N(TP)}{N(TP) + N(FP)} \quad (2)$$

$$Recall = \frac{N(TP)}{N(TP) + N(FN)} \tag{3}$$

The F1 score is a measure of a test's accuracy. It is the weighted average between precision and recall. The formula for this score is given in (4). In our case, it's computed over the individual words in the prediction against those in the True Answer. The number of shared words between the prediction and the truth is the basis of the F1 score.

$$F1\ score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \tag{4}$$

To demonstrate the benefit of FedQAS, we partitioned the SQuAD dataset into 5 equal chunks, so that each client has "20%" of the total dataset. We then compare the federated scenario to centralized model training. Table 4 lists the available metrics for different training rounds of the global model. Our implemented model baselines show similar EM and F1 scores with the global model during the first rounds and slightly outperform the baseline with respect to data privacy and knowledge sharing across participants. Overall, the result shows that question-answering in federated learning settings performs well compared to centralized settings. This is due to the used hyper-parameters in the federated learning setting, see Figure 5 for the convergence of accuracy (EM) and Figure 6 for the convergence of F1.

Table 4. Comparisons with equivalent parameters on the validation set of SQuAD1.0.

Model	F1 Score	Accuracy (EM)
Baseline	0.31	0.75
FedQAS	0.33	0.81

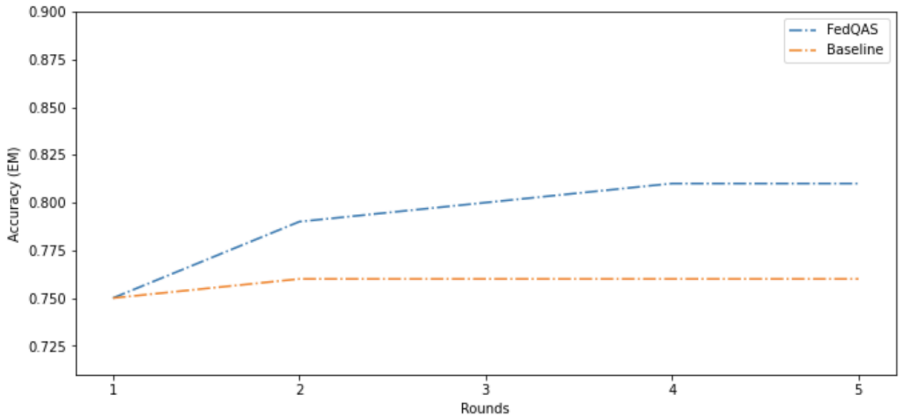

Figure 5. Convergence of accuracy (Exact Match) on the SQUAD dataset with 1 combiner, 5 clients and 5 rounds.

In terms of resources, the result proves the fact that model architecture affects client training time and combiner round time. Hence, training a large model (400 MB) in a centralized way requires more resources than the federated setting. For demonstration, we consider a FEDn network consisting of a single, high-powered combiner (8 VCPU, 32 GB RAM) with connected clients (8 VCPU, 32 GB RAM) instances in SSC (SNIC Science Cloud [37]) and measure the average round time over five global rounds. Figure 7 shows round time for global model training, since the model size affects both the training time at clients and the cost for data transfer and model aggregation, we show the mean training time for reference.

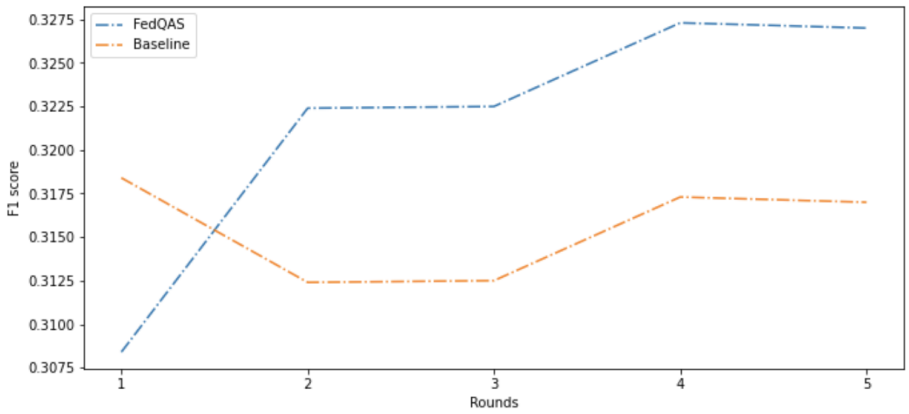

Figure 6. Convergence of F1 score on SQUAD dataset with 1 combiner, 5 clients and 5 rounds.

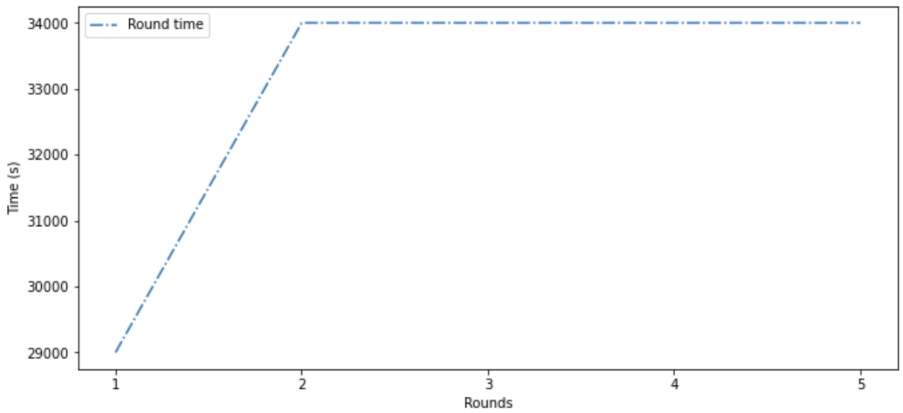

Figure 7. Round times for global model training (FEDn network).

To gain an intuitive observation of the predictions, we give a prediction example on SQuAD1.0 from both the baseline and federated model in Table 5, which shows that FedQAS works better at answering the question on a given passage. Hence, the proposed approach has contributed overall to a better understanding of QA, preserving data privacy, and contributed to low-cost training as well as a collaborative question answering system task using large models.

Table 5. Comparison of answer prediction on test data.

Title: Project Apollo
Passage: The Apollo program, also known as Project Apollo, was the third United States human spaceflight program carried out by the National Aeronautics and Space Administration (NASA), which accomplished landing the first humans on the Moon from 1969 to 1972. First conceived during Dwight D. Eisenhower's administration as a three-man spacecraft to follow the one-man Project Mercury which put the first Americans in space, Apollo was later dedicated to President John F. Kennedy's national goal of landing a man on the Moon and returning him safely to the Earth by the end of the 1960s, which he proposed in a 25 May 1961, address to Congress. Project Mercury was followed by the two-man Project Gemini. The first manned flight of Apollo was in 1968. Apollo ran from 1961 to 1972, and was supported by the two man Gemini program which ran concurrently with it from 1962 to 1966. . .
Question 1: How long did Project Apollo run?
Gold answer (human): 1961 to 1972 **Google search engine answer:** see Figure 1 **Baseline model answer:** 1961 to 1972 **FedQAS answer:** 1961 to 1972

Table 5. *Cont.*

Question 2: What program was created to carry out these projects and missions?
Gold answer (human): Apollo program **Baseline model answer:** National Aeronautics and Space Administration **FedQAS answer:** Apollo program
Question 3: What year did the first manned Apollo flight occur?
Gold answer (human): 1968 **Baseline model answer:** 1968 **FedQAS answer:** 1968 **Question 4:** What President is credited with the original notion of putting Americans in space?
Gold answer (human): John F. Kennedy **Baseline model answer:** John F. Kennedy **FedQAS answer:** John F. Kennedy
Question 5: Who did the U.S. collaborate with on an Earth orbit mission in 1975?
Gold answer (human): Soviet Union **Baseline model answer:** Soviet Union **FedQAS answer:** Soviet Union
Question 6: How long did Project Apollo run?
Gold answer (human): 1962 to 1966 **Baseline model answer:** 1961 to 1972, and was supported by the two man Gemini program which ran 1966 **FedQAS answer:** 1962 to 1966
Question 7: What program helped develop space travel techniques that Project Apollo used?
Gold answer (human): Gemini **Baseline model answer:** Gemini **FedQAS answer:** Gemini
Question 8: What space station supported three manned missions in 1973–1974?
Gold answer (human): Skylab **Baseline model answer:** Skylab **FedQAS answer:** Skylab

4.2. Implementation and Demo Environment

Designing and developing question answering in a privacy-aware manner is not a trivial task. It requires design strategies to comply with data governance and privacy regulations. Several third-party frameworks have been proposed for federated learning; providing open-source building blocks that help to collaborate in training machine learning models. The present QAS application framework needs to provide scalability, large models training, and production-grade features such as robustness to failure. Based on these requirements, we chose to design and develop our proposed FedQAS system on top of private data using the FEDn framework [2]. FEDn is an open-source, modular, and model agnostic framework for federated machine learning. We developed interactive and user-friendly interfaces using the Flask framework (https://flask.palletsprojects.com, accessed on 24 January 2022), which make it easy for a third party to contribute to data annotation and then participate in training global models directly from their location site. The proposed FedQAS has the following features:

- Privacy-preserving: sharing only model parameters with a central server (cloud) and keeping data private on the client side,
- Incremental learning: improving the global models by attaching more clients and adding new data points,
- Robust: robust enough to deal with natural language tasks (e.g., question answering, chatbot, etc.) and large models in a geographically distributed manner,

- Multilingual: language agnostic, can be trained on any language,
- Standalone: multiple platforms (i.e., guarantee for low disk and memory footprint). It can be run production-grade on a standard laptop having two cores and 2GB of RAM,
- Accuracy and F1 score: achieve competitive performance compared with centralized training and the used baseline model (see experiment and evaluation section).

The proposed FedQAS is composed of three main components: FEDn for collaborative training, MongoDB (https://www.mongodb.com, accessed on 24 January 2022) as a NoSQL [38] database and question answering UI for prediction and local incremental learning. The system is interactive, scalable, suitable for secure collaborative training and data privacy-preserving, and can be used both in the cloud, on edge nodes and in a standalone mode. It is accessible from different platforms to engage a wide range of users, and it is also optimized for both desktop and mobile. The source code is publicly available on Github via this link https://github.com/aitmlouk/FEDn-client-FedQAS-tf.git, accessed on 24 January 2022.

FedQAS is, to the best of our knowledge, the only approach for question answering that supports data privacy and knowledge sharing through federated learning. Its main value is to provide an environment to quickly ensure data privacy and low-cost training by collaborative training. Nearly every deep learning application can benefit from data privacy and knowledge sharing across a different client in a federated learning setting. The transformer model used in FedQAS can be improved by tuning parameters and using transfer learning for new pre-trained models (GPT-2 [39], GPT-3 [40], etc.).

5. Conclusions

In this paper, we have proposed FedQAS, a high-quality question answering (FedQAS) approach, to address the data-sharing issue in machine learning. Validation experiments for FedQAS was implemented based on 5 rounds of training with a transformer neural network. The system consists of several components including the private data pipeline, collaborative training and private incremental learning. Experiments on the SQuAD dataset using the transformer architecture demonstrate that our FedQAS significantly outperforms the baseline model performances, protecting data privacy and sharing knowledge within an alliance. The proposed FedQAS allows collaborators (collaborative training participants) to have overall control of their sensitive and private data while collaboratively training question answering models. The integration of federated learning within machine reading comprehension provides a sustainable solution by preserving data privacy and ensuring low-cost training. We conclude that the application of FL to NLP tasks such as question answering can contribute to solving the problem that arises when using machine learning in the context of data protection and privacy. The system actively supports end-users in joining training and improving the performance through incremental learning on a various range of local clients.

In future work, we aim to extend the FedQAS to cover more datasets in a geographically distributed manner and test the model with other aggregation algorithms (e.g., FedOPT, FedProx, etc.) for federated learning. We also plan to fine-tune the pre-trained models such as BERT large, GPT-2 for MRC and particularly investigate their effectiveness in federated learning settings especially when it comes to large private documents (text).

Author Contributions: Formal analysis, A.A.-M.; investigation, A.A.-M.; methodology, A.A.-M.; project administration, A.H.; writing—review and editing, S.A.A., S.T. and A.H. All authors have read and agreed to the published version of the manuscript.

Funding: Funding has been provided by the eSSENCE strategic collaboration on eScience (Ait-Mlouk, Alawadi, Toor, and Hellander) and the Swedish Innovation Agency Vinnova grant no. 2019-02819 (awarded to Scaleout Systems AB).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data used to support the findings of this work can be found on this link https://github.com/aitmlouk/FEDn-client-FedQAS-tf/tree/main/data, accessed on 24 January 2022.

Acknowledgments: The computations were enabled by resources provided by the Swedish National Infrastructure for Computing (SNIC) at C3SE partially funded by the Swedish Research Council through grant agreement no. 2018-05973.

Conflicts of Interest: The authors declare no conflict of interest regarding the design of the study; the collection, analyses, or interpretation of data; the writing of the manuscript, or the decision to publish the results.

References

1. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv* **2016**, arXiv:1606.05250.
2. Ekmefjord, M.; Ait-Mlouk, A.; Alawadi, S.; Åkesson, M.; Stoyanova, D.; Spjuth, O.; Toor, S.; Hellander, A. Scalable federated machine learning with FEDn. *arXiv* **2021**, arXiv:2103.00148.
3. Lehnert, W. The Process of Question Answering. Ph.D. Thesis, Yale University, New Haven, CT, USA, 1977.
4. Hirschman, L.; Light, M.; Breck, E.; Burger, J.D. Deep Read: A Reading Comprehension System. In Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics, ACL '99, College Park, MD, USA, 20–26 June 1999; Association for Computational Linguistics: Stroudsburg, PA, USA, 1999; pp. 325–332. [CrossRef]
5. Riloff, E.; Thelen, M. A Rule-Based Question Answering System for Reading Comprehension Tests. In Proceedings of the 2000 ANLP/NAACL Workshop on Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Sytems—Volume 6, ANLP/NAACL-ReadingComp '00, Seattle, WA, USA, 4 May 2000; Association for Computational Linguistics: Stroudsburg, PA, USA, 2000; pp. 13–19. [CrossRef]
6. Charniak, E.; Altun, Y.; de Salvo Braz, R.; Garrett, B.; Kosmala, M.; Moscovich, T.; Pang, L.; Pyo, C.; Sun, Y.; Wy, W.; et al. Reading Comprehension Programs in a Statistical-Language-Processing Class. In Proceedings of the ANLP-NAACL 2000 Workshop: Reading Comprehension Tests as Evaluation for Computer-Based Language Understanding Systems, Seattle, WA, USA, 4 May 2000; Association for Computational Linguistics: Stroudsburg, PA, USA, 2000.
7. Wang, Z.; Hamza, W.; Florian, R. Bilateral Multi-Perspective Matching for Natural Language Sentences. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, Melbourne, Australia, 19–25 August 2017; pp. 4144–4150. [CrossRef]
8. Richardson, M.; Burges, C.J.; Renshaw, E. MCTest: A Challenge Dataset for the Open-Domain Machine Comprehension of Text. In Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, Seattle, WA, USA, 18–21 October 2013; Association for Computational Linguistics: Stroudsburg, PA, USA, 2013; pp. 193–203.
9. Wang, W.; Yan, M.; Wu, C. Multi-Granularity Hierarchical Attention Fusion Networks for Reading Comprehension and Question Answering. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; Volume 1: Long Papers, pp. 1705–1714. [CrossRef]
10. Zhang, Z.; Wu, Y.; Zhao, H.; Li, Z.; Zhang, S.; Zhou, X.; Zhou, X. Semantics-aware BERT for Language Understanding. *arXiv* **2020**, arXiv:1909.02209.
11. Zhang, Z.; Wu, Y.; Zhou, J.; Duan, S.; Zhao, H.; Wang, R. SG-Net: Syntax-Guided Machine Reading Comprehension. *arXiv* **2019**, arXiv:1908.05147.
12. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bidirectional Attention Flow for Machine Comprehension. *arXiv* **2018**, arXiv:1611.01603.
13. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. 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, Minneapolis, MN, USA, 2–7 June 2019; Association for Computational Linguistics: Stroudsburg, PA, USA, 2019; Volume 1 (Long and Short Papers), pp. 4171–4186. [CrossRef]
14. Zhuang, L.; Wayne, L.; Ya, S.; Jun, Z. A Robustly Optimized BERT Pre-training Approach with Post-training. In Proceedings of the 20th Chinese National Conference on Computational Linguistics, Hohhot, China, 13–15 August 2021; Chinese Information Processing Society of China: Beijing, China, 2021; pp. 1218–1227.
15. Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R.; Le, Q.V. XLNet: Generalized Autoregressive Pretraining for Language Understanding. *arXiv* **2020**, arXiv:1906.08237.
16. Peters, M.E.; Neumann, M.; Iyyer, M.; Gardner, M.; Clark, C.; Lee, K.; Zettlemoyer, L. Deep Contextualized Word Representations. In Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, New Orleans, LA, USA, 1–6 June 2018; Volume 1 (Long Papers); Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 2227–2237. [CrossRef]
17. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention Is All You Need. *arXiv* **2017**, arXiv:1706.03762.

18. Yamada, I.; Asai, A.; Shindo, H.; Takeda, H.; Matsumoto, Y. LUKE: Deep Contextualized Entity Representations with Entity-aware Self-attention. *arXiv* **2020**, arXiv:2010.01057.
19. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A Lite BERT for Self-supervised Learning of Language Representations. *arXiv* **2020**, arXiv:1909.11942.
20. Zhang, Z.; Yang, J.; Zhao, H. Retrospective Reader for Machine Reading Comprehension. *arXiv* **2020**, arXiv:2001.09694.
21. Hermann, K.M.; Kočiský, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching Machines to Read and Comprehend. *arXiv* **2015**, arXiv:1506.03340.
22. Hill, F.; Bordes, A.; Chopra, S.; Weston, J. The Goldilocks Principle: Reading Children's Books with Explicit Memory Representations. *arXiv* **2016**, arXiv:1511.02301.
23. Bajaj, P.; Campos, D.; Craswell, N.; Deng, L.; Gao, J.; Liu, X.; Majumder, R.; McNamara, A.; Mitra, B.; Nguyen, T.; et al. MS MARCO: A Human Generated MAchine Reading COmprehension Dataset. *arXiv* **2018**, arXiv:1611.09268.
24. Trischler, A.; Wang, T.; Yuan, X.; Harris, J.; Sordoni, A.; Bachman, P.; Suleman, K. NewsQA: A Machine Comprehension Dataset. In Proceedings of the 2nd Workshop on Representation Learning for NLP, Vancouver, BC, Canada, 3 August 2017; Association for Computational Linguistics: Stroudsburg, PA, USA, 2017; pp. 191–200. [CrossRef]
25. Dunn, M.; Sagun, L.; Higgins, M.; Guney, V.U.; Cirik, V.; Cho, K. SearchQA: A New Q&A Dataset Augmented with Context from a Search Engine. *arXiv* **2017**, arXiv:1704.05179.
26. Joshi, M.; Choi, E.; Weld, D.S.; Zettlemoyer, L. TriviaQA: A Large Scale Distantly Supervised Challenge Dataset for Reading Comprehension. *arXiv* **2017**, arXiv:1705.03551.
27. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. SQuAD: 100,000+ Questions for Machine Comprehension of Text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, Austin, TX, USA, 1–5 November 2016; Association for Computational Linguistics: Stroudsburg, PA, USA, 2016; pp. 2383–2392. [CrossRef]
28. Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. *arXiv* **2018**, arXiv:1806.03822.
29. Reddy, S.; Chen, D.; Manning, C.D. CoQA: A Conversational Question Answering Challenge. *Trans. Assoc. Comput. Linguist.* **2019**, *7*, 249–266. [CrossRef]
30. Rajpurkar, P.; Jia, R.; Liang, P. Know What You Don't Know: Unanswerable Questions for SQuAD. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, Melbourne, Australia, 15–20 July 2018; Volume 2: Short Papers; Association for Computational Linguistics: Stroudsburg, PA, USA, 2018; pp. 784–789. [CrossRef]
31. Sweeney, L. K-Anonymity: A Model for Protecting Privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
32. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. L-Diversity: Privacy beyond k-Anonymity. *ACM Trans. Knowl. Discov. Data* **2007**, *1*, 1–52. [CrossRef]
33. Li, N.; Li, T.; Venkatasubramanian, S. t-Closeness: Privacy Beyond k-Anonymity and l-Diversity. In Proceedings of the 2007 IEEE 23rd International Conference on Data Engineering, Istanbul, Turkey, 15–20 April 2007; pp. 106–115. [CrossRef]
34. Rocher L., H.J.; de Montjoye YA. Estimating the success of re-identifications in incomplete datasets using generative models. *Nat. Commun.* **2019**, *10*, 3069. [CrossRef]
35. McMahan, H.B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-Efficient Learning of Deep Networks from Decentralized Data. *arXiv* **2017**, arXiv:1602.05629.
36. Wu, Y.; Schuster, M.; Chen, Z.; Le, Q.V.; Norouzi, M.; Macherey, W.; Krikun, M.; Cao, Y.; Gao, Q.; Macherey, K.; et al. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *arXiv* **2016**, arXiv:1609.08144.
37. Toor, S.; Lindberg, M.; Falman, I.; Vallin, A.; Mohill, O.; Freyhult, P.; Nilsson, L.; Agback, M.; Viklund, L.; Zazzik, H.; et al. SNIC Science Cloud (SSC): A National-Scale Cloud Infrastructure for Swedish Academia. In Proceedings of the 2017 IEEE 13th International Conference on e-Science (e-Science), Auckland, New Zealand, 24–27 October 2017; pp. 219–227. [CrossRef]
38. Pokorny, J. *NoSQL Databases: A Step to Database Scalability in Web Environment*; iiWAS '11; Association for Computing Machinery: New York, NY, USA, 2011; pp. 278–283. [CrossRef]
39. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language Models are Unsupervised Multitask Learners. *OpenAI Blog* **2019**, *1*, 9.
40. Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners. *arXiv* **2020**, arXiv:2005.14165.