A Heuristic Architecture for Federated Question Answering System for Personal Health Knowledge Graphs

Yuxuan Wang, Oshani Seneviratne
Rensselaer Polytechnic Institute

ABSTRACT. Federated Learning (FL) is proven to be a promising technique for allowing edge devices to participate in model training processes while keeping users' privacy-sensitive data on the devices. An application that leverages FL to deliver dietary recommendations that utilize personal health data from edge devices remains to be seen. We describe an Information-Retrieval-based (IR) Knowledge Graph Question Answering (KGQA) architecture that can be used for this purpose.

BACKGROUND. Federated learning, first popularized by Google in an application of next word prediction in remote mobile keyboards [1], has gained much popularity in recent years as an effective solution for multi-party machine learning where participants train a shared model under the orchestration of a secure central server. However, it is challenging to develop a KGQA system that adopts the FL paradigm, mainly due to client devices' limited computational and storage resources. Therefore, we designed an FL-KGQA architecture with the following design goals: (1) Any user-specific information should be agnostic to the service provider, (2) Computational expensive operations, such as query execution, should happen on the server-side, and (3) Storage consumption on the client-side should be minimal.

THE ARCHITECTURE. One key innovation is that we decompose most IR-based KGQA models into a *Candidate Generator* and an *Answer Selector*. Another innovation is that each edge device only caches and actively manages a knowledge base subset pertinent to previous questions. Therefore, we design the FL system as a composition of three submodules that can operate asynchronously to support the KGQA service:

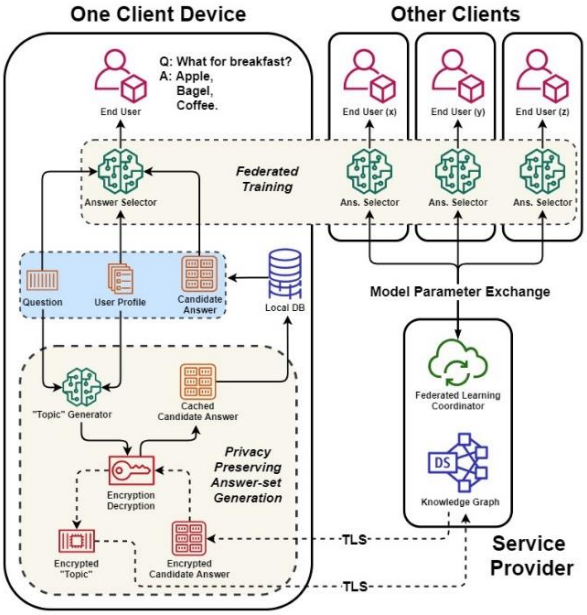

Figure 1: Fed-KGQA Architecture for Answering Personalized Questions

- **Candidate Generator:** The module first parses a natural language question into a query that encodes the given question's gist, e.g., the entities associated with the topic. It then sends the query to the server and retrieves a set of possible answers. Lastly, a triple consisted of the question, user profile, and the candidate set is cached locally.
- **Answer Selector:** We can formalize a selector as a function that takes the question-profile-answers triple as input, select a subset from the candidates, and rank them. Since we have the triples stored on the client device, we can update the model following the standard FL paradigm.
- **Data Manager:** Two primary responsibilities for a data manager are: (1) regularly discard the cache for data freshness, security, and efficient storage, (2) work as a daemon program that manages cache received from the generator and prepares the input triples for answer selector when the user asks questions or when the edge device is sampled for an FL training session.

References:

[1] H Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, et al. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629, 2016.