Personalized Recommendation Prototype: Knowledge Requirements

I. Synopsis:

The goal of the project is to prototype a mobile application which leverages user data and preferences to perform content recommendation. The models used are to be kept on-device, quelling privacy concerns by using a decentralized approach. They should also be personalized, in that the decentralized approach requires training and inference to occur using majority (or solely) data centric to the user. Thus, efficient/batch learning should be prioritized for supervised processes. Prototyping should also explore the use of leveraging graph data in both supervised and unsupervised settings where their use can be made efficient on-device.

The current stage of prototyping explores the recommendation of tweets by leveraging a user's twitter profile and involving them in a labeling process. This should branch out towards user recommendations, as well as recommendations in other domains. Primarily, web scraping and recommendation of news, posts, journals, or even other streams of media. The current goal was to prioritize recommendation of academic articles by leveraging the user's involvement and profiling their interest. The balance is to be explored in how much and which data to use (Twitter, user labeling, other accessible accounts, etc…) whilst keeping learning manageable on-device. In short, we prototype a content regulator which can better leverage all the of data kept on a mobile device.

II. Background:

This project stems from endeavors to explore on-device learning, with significant emphasis on graph learning and formulating on-device data. Familiarity with machine learning frameworks is a must, as is the ability to curate and transform data for learning. The current prototyping system written in Python showcases the potential of such an arrangement. It was devised using knowledge of API calling and web crawling for data collection, alongside understanding of data availability on a mobile device. The process was rapidly prototyped as shown with existing knowledge of machine learning frameworks by making the most obvious assumptions given the data.

Once settled on Twitter, basic API calls were used given the prior considerations of what data would be useful for a learning model. Understanding of simple NLP processes in preprocessing, tokenization and embedding were necessary to transform the data itself. Sufficient programming experience was necessary to build this pipeline (vector/matrix manipulation, recursion, exception handling). Given the transformed data, knowledge of deep learning (neural networks) was used to build a simple dense model for binary classification. A simple label tool was built using a GUI, which albeit simple showcases potential for user-involvement and the need for user-facing interface knowledge.

III. Responsibilities:

First steps should explore the embedding approach with respect to user-profiles themselves. Then, exploration into other content domains, namely crawling academic articles using API tools such as Semantic Scholar. Embedding of paper abstracts should be used for classification, alongside any other pertinent information. However, exploration into recycling knowledge of the user from prior learning/embedding (i.e. Twitter profile, or others) to influence recommendations could in other domains may be powerful. Different embedding approaches should be explored with respect to efficiency and accuracy. Cutting edge NLP technology such as Roberta, a personalized approach such as Doc2Vec, or a combination of both are all potential options.

To implement the prototype itself with aim for computation on-mobile devices, strong algorithmic programming and machine learning knowledge is required. Since mobile training and deep learning are not commonplace, theoretical understanding of learning models is needed to optimize and tweak their function for mobile devices. Where existing tools are not sufficient, such as certain graph-based methods, it will be mandatory to understand how to efficiently program these models from scratch. Which classification models to invest in will require monitoring and experimentation given tuning of simple dense models, as well as introduction of less straightforward approaches in graph learning. This requires some understanding of resource management (memory/CPU) to best suit the task to the mobile phone without overloading.

With respect to the broader project, the system prototype is quite small. A framework to reliably crawl for new data (either to recommend or motivate training) will be needed and may leverage learning of its own to selectively filter from sources. Given that a graph structure is present behind many of these data sources, graph neural networks may prove better suited for classification or embeddings tasks or in combination with simpler methods. Node ranking may be applied for filtering or ordering of nodes to recommend, with the motivation of highly efficient models available which will scale for on-device learning.

The overall project should be prototyped into a user application on a mobile device that involves the user data, as well as clever integration of user interaction into training. Strong software development skills are needed to prototype an application which is of sufficient quality for the larger goal of deploying a working version. Specifically, experience with UX/UI or GUI systems is highly valuable to ensure the app is highly usable whilst leveraging user involvement. Although mobile development experience is preferred, strong programming skills in any class-oriented language is sufficient to build the back-end framework.

IV. Required Knowledge

- Strong understanding of machine learning processes (training/evaluation, choosing correct model given data, efficiency aspects of approaches, data tranformsation)
- Theoretical background in machine learning, specifically traditional deep learning (MLP, convolution)
- Natural Language Processing, particularly: preprocessing, tokenization, and text embedding
- Strong algorithmic coding with respect to optimization and complex computation, preferably in C/C++
- Software development paradigms sufficient for prototyping a class driven application, using object-oriented languages such as Java, Swift, etc.
- Background in mathematics, particularly graph and matrix theory
- Experience creating GUI, user-facing applications

V. Preferred Knowledge

- Using API and web crawling technology to compile data for learning, particularly raw HTML/JSON formats
- Knowledge of existing machine learning toolkits (PyTorch, Tensorflow, SciKit) in Python or languages
- Android/iOS application development experience (Java, Kotlin, Swift)
- Ability to use HTML/XML or other markup languages and UI tools with respect to mobile development
- Existing knowledge of graph learning approaches (graph neural networks, clustering, graph classification)
- Knowledge of OS resource management (memory/CPU/power), preferably with respect to a mobile device
- Familiarity with Android/iOS documentation and API with respect to data avalability