IA Project Proposal

Thomas Faulhaber

Summary

For the IA, I'm making a web app that can identify the author of an unlabeled handwriting sample given a set of labeled ones. This will help teachers with a lot of students narrow down potential authors for an assignment with the name missing. I have dubbed this project "author-id."

The Scenario

I'm developing this tool for my client and advisor, Mr. Rice, who is the AP Statistics and IB Economics teacher here at Ingraham. In the back of his classroom he has a tray full of homework assignments that students forgot to write their names on, so I figured it would be convenient for him if he could just take a picture of a nameless assignment and figure out who wrote it. This seemed like an interesting problem to me, and Mr. Rice really liked the idea when I suggested it. The overall goal of the project is to implement a program any teacher can use to identify the author of a homework assignment turned in without a name, even if they don't have every student's handwriting memorized.

Rationale

This is the best solution to Mr. Rice's problem, since it requires merely that he collect one sizable handwriting sample from each of his students. Then, he can just take a picture of an assignment he wants to identify and ask the student whose name it returns if they wrote it. Both Mr. Rice, in my most recent consultation with him, and Mr. Britton expressed concern about the collecting of student data for the development of this app. Such collection will not be necessary for development, though it may eventually be necessary for testing and demonstration. A dataset licensed in Creative Commons which is suited to my task already exists here.

I also think that Mr. Rice is a good choice for my advisor because his expertise in statistics could come in handy for the machine learning aspect of this task. I know he has some experience doing statistical modeling in the R language. He will probably be able to provide at least some insight into the development of the machine learning model.

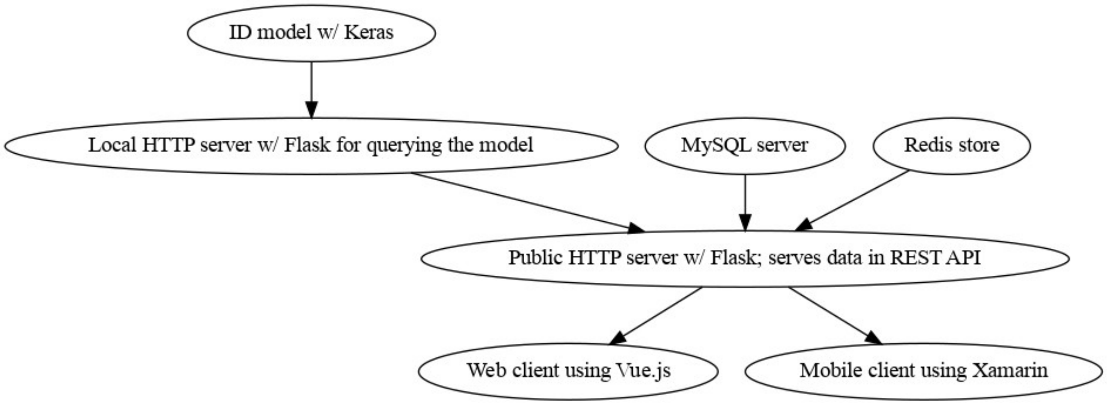
All of the software I plan to use is free and open-source. The server-side of the application will be written in Python using the Flask library for creating web services. It will query a MySQL server for data, though memory may be cached in a Redis key-value store. The machine-learning model for identification will be encapsulated from the rest of the program, accessed through its own local HTTP server, and it will use the Keras deep learning library with TensorFlow as its backend.

I have two different frontends planned for the app: one for the desktop, using the Vue JavaScript framework, and one for mobile using Xamarin, an open-source platform for building mobile apps using .NET languages. If I only have time to write one, I'll write only the first one.

Design

I've already detailed my full technology stack, but I'll go into more detail about how the different parts interact here.

Here's a nifty graph I made with Graphviz:


ID model w/ Keras
Local HTTP server w/ Flask for querying the model
MySQL server
Redis store
Public HTTP server w/ Flask; serves data in REST API
Web client using Vue.js
Mobile client using Xamarin

The ID model will be the same across all instances of this app. It will never be tuned to a specific user's samples. Instead, it will produce a locality-sensitive fingerprint for a sample, which can then be compared statistically to a set of labeled samples. A sample which the user (the teacher) asks the app to identify will be plugged into this model, yielding a fingerprint. That fingerprint will then be compared to the fingerprints for all of the labeled samples using a k-nearest-neighbor algorithm (which is simple to implement using the open-source Python package Scikit-Learn). This will produce a ranked list of candidates which can be presented to the user.

The web client will be a single-page Vue application which uses JavaScript's fetch API to access the app's RESTful interface. This way it will be far easier to extend it to serve a mobile frontend (in fact, very little modification will need to be made to the backend code, if any). I mentioned earlier that Xamarin uses .NET languages. Any .NET language will work. I am going to use VB.NET, which is functionally identical to C# but uses BASIC syntax, just because I like it more. It feels retro. I do not expect this to cause any additional difficulty, since it is basically always possible to translate between the two languages line-for-line.

Timeline

1. Build ML model for handwriting identification using the dataset linked in the "Rationale" section. Should be done by Feb 28th. This will probably be the most complicated part, since writing code for deep learning is particularly finicky.
2. Build a simple Flask microservice for querying that model locally. Should be done by March 4th. This part should be pretty simple.
3. Build a REST API with Flask for the full service. Should be done by March 11th.
4. Build a single-page web app with Vue that uses that API. This will be served on the same server, but will be encapsulated from the API. Should be done by March 18th.
5. Build a mobile app with Xamarin that uses that API. This part is optional. Hopefully done by March 25th.

Success criteria

1. The identification model can be queried over an HTTP server running on localhost rapidly and without error.
2. The REST API can be queried over an HTTP server rapidly and without error.
3. The web app is accessible and functional, allowing user uploads and giving responses to queries from the client.