454 Project Proposal

(1) Team Info

<table>
<thead>
<tr>
<th>Name</th>
<th>Email</th>
<th>Github</th>
</tr>
</thead>
<tbody>
<tr>
<td>Greg Brandt</td>
<td><a href="mailto:brandtg@uw.edu">brandtg@uw.edu</a></td>
<td>brandtg</td>
</tr>
<tr>
<td>Corneliu Suciu</td>
<td><a href="mailto:tonis821@uw.edu">tonis821@uw.edu</a></td>
<td>corneliusuciu</td>
</tr>
</tbody>
</table>

(2) Problem

When using a mobile app, sifting through a lot of information is costly. Users could more effectively use mobile services if they minimize the amount of information they have to sift through. Moreover, the cost of sifting through information is higher on a mobile device than on a traditional computer because of screen size and processing constraints.

This problem can be averted if the user finds what he or she wants within the first few results, without having to dig too deeply into the meta data of any given result.

Our project is to implement a restaurant selection tool that learns the user’s preferences (i.e. most relevant meta data to the user), which orders query results and selectively displays meta data such that the information is most relevant to that particular user.

The tool presents two restaurants to the user, and allows the user to eliminate the less desirable of the two before being presented with another query result. The score by which a restaurant can be ranked is related to how many of these “rounds” a restaurant survives. The meta data associated with that restaurant are then considered more important to the user, and result ordering is updated accordingly.

(3) Artifacts

- Web-only HTML5 + CakePHP application (URL)
- Android app package
- End-user and Developer documentation
- All source code for Web and Android applications

(4) Methods and Techniques

Our server will be implemented in CakePHP and adhere to the MVC design pattern. The server will store an object that performs custom result ordering for each user, as well as perform our machine learning algorithm over session data, which is necessary to create and update these objects. The server will be hosted on Amazon EC2.
Our web interface will be implemented in HTML5, and we will use the socket.io, jQuery and jQuery mobile JavaScript frameworks. There will be both full browser and mobile optimized views of the application. HTML5’s local storage facility will be used to store things like a user identifier to preserve state across sessions. The Android application will be essentially a native implementation of the mobile view.

Queries are passed to the revminer.com service in real time. The HTML5 and Android applications will compute session data; that is, ordering information necessary to update / create the comparison object. These data are periodically sent to the server for processing. This object is used to reorder the results in a way that query results that are more relevant to the user appear before those that aren’t.

(5) Division of Labor
The project is naturally divided into two parts: the core machine learning algorithm that provides intelligent query reordering, and the application infrastructure (i.e. UI, server architecture, communication with external services, etc...). Corneliu will take the lead on the core machine learning algorithm, and Greg will take the lead on the application infrastructure, allowing for overlap as we see fit.

(6) Milestones
1. HTML5 application infrastructure done. Initial, incomplete implementation of ordering / mining algorithm. Infrastructure should allow easy plug-in of this algorithm. Begin learning Android SDK.
2. Android application done. Algorithm “successfully” (on an intuitive level) ordering results.
3. Test tools built: come up with hypothetical users + preferences + desired results. Polish everything.

(7) Evaluation of Success
We will evaluate success primarily on the relative position of relevant and / or specifically desired restaurants in our reordered results. For example, if our query is “good mexican food”, and we eventually end up choosing “Agua Verde”, a successful scenario would be “Agua Verde” appearing 2nd in our results vs. 5th in revminer’s results.