MOBILE APP RECOMMENDATION

Eugene Agichtein, Jaime Arguello, Christina Lioma, and Ryen White

Day 1 – Research Topic Discussion

Mobile apps (short for "mobile applications") are software tools that are specifically designed to accomplish a particular higher-level task from a mobile device. For example, Shazam[1] is an app that automatically identifies a song currently playing in the user's physical space and then gives the user the option to purchase the song/album online. As the usage of smartphones and tablets continues to grow, people are increasingly in situations where they are depending on their mobile devices to assist with task completion. There is an opportunity to provide a service that will recommend a relevant application (or set of applications) to users based on information about their current task gathered implicitly from behavioral data and other contextual signals such as user location and time of day.

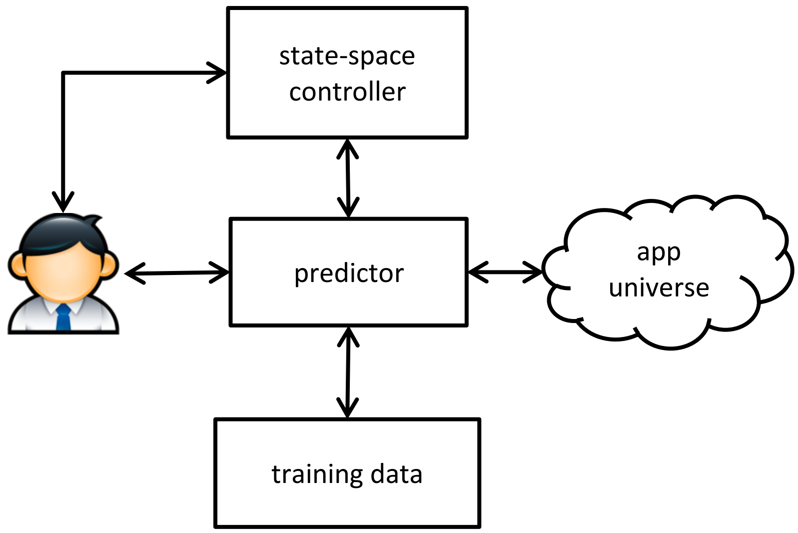
Our group discussion focused on the task of *app recommendation*. Specifically, we focus on apps that help users accomplish tasks that require several subtasks, such as planning an evening out or purchasing a product. The goal for an app recommendation system would be to automatically recognize the task the user is trying to accomplish (using information derived from the user's context) and to recommend apps that might assist the user in accomplishing this task. Given that mobile devices can capture rich information about a user's geo-temporal context (and compare this with historical information from the user and other users), we focused our discussion to the mobile domain. However, the same idea could also generalize to the desktop domain, where operating systems such as Microsoft Windows are also moving toward an application model. Example tasks in the mobile domain include preparing dinner for guests (deciding what to cook, what ingredients to buy and where, and learning how to prepare each item) or planning a day of activities in a new city (deciding on places to visit throughout the day given a set of budgetary constraints). For these tasks, the user may need to collect and synthesize information from multiple sources and leverage different applications to accomplish each of the subtasks.

Our discussion of app recommendation was driven by the following questions:

1. What would the system architecture look like?
2. What types of contextual signals or features might the system use to predict app relevance?
3. What are some of the major challenges in developing such a system?
4. What are some of the related subfields of IR?

The basic system architecture is shown below.

[1] http://www.shazaam.com



We envision the following inputs to the system (the predictor): a universe of retrievable mobile apps (either installed on the user's mobile device or available online), some representation of the user's context (possibly enhanced by the state-space controller), and a set of positive/negative recommendation instances to be used as training data. Each training instance would consist of a context-app pair and a positive/negative relevance label.

The predictor would be a machine learned model trained to predict app relevance as a function of a set of input features. Possible features could be derived from the user's context (including location, time, speed and direction of travel, proximity of friends, etc.), from meta-data associated with the app, and from previous user interactions with the system. From this information, the predictor would need to create a representation of the task that could be used to find related applications. Training labels could be derived from either explicit or implicit user feedback. The training data could be initially gathered from third-party judges who could make a determination based on the appropriateness of the application for a context provided to them, but could then shift to the utilization of online signals based on users' reaction when the recommendations are made (e.g., did they accept or decline the invitation to use the application when it was offered to them?). The goal of the state-space controller would be to derive higher-level contextual features that might narrow the space of possible tasks the user is trying to accomplish. These higher-level features could then be incorporated into the predictor's feature representation. An example higher-level feature might be whether the user is currently in an outlier location. The state-space controller could make such a prediction using the current time and location information and comparing this with the user's historical profile. Knowing that the user is an unexpected location may suggest that the user is trying to accomplish a task that requires finding a business or landmark, or perhaps in addition to their inferred task, they may also need help in situating themselves within their physical environment.

The following are some of the major challenges we identified. At the core, we view app recommendation as a search problem. Given a user who is performing a particular task, the goal for the system is to retrieve one or several apps likely to be relevant to the task. Thus, some of the same challenges associated with any search system apply: modeling the information need (the task in this case), finding features that are correlated with app relevance, estimating recall[2], and

[2] Estimating recall is challenging when relevance is derived from user interactions with search results.

personalizing results in light of data sparseness. However, we believe that other challenges are unique to app recommendation. Our group discussion focused on two unique challenges. First, what is the optimal stage within the task-completion lifecycle to recommend a potentially relevant app? Recommending an app too soon may interrupt the user's cognitive flow. The user may not yet fully understand the steps and processes required to complete the task and may therefore not recognize a relevant app recommendation. On the other hand, recommending an app too *late* may render the recommendation useless. The user may have already committed to accomplishing the task using one or several familiar (and potentially suboptimal) tools and may lose much momentum in switching. Second, how should the system transfer task-state. Recognizing that the user is working on a task implies that the user has already started working on the task. Thus, it seems necessary to be able to transfer information about the already-completed portion of the task to the new app. Otherwise, the user may incur too heavy of a cost in switching. The latter challenge requires some degree of interoperability between apps, perhaps even in the form of a standard for information sharing.

Several areas within IR seem to be related to app recommendation. *Contextual suggestion* is the task of recommending things to do for a user that is in a particular geo-temporal context and has been investigated at TREC 2012 and 2013. This is a simpler problem than app recommendation as the user's task is consistently "entertain me" and the retrievable units are webpages associated with local businesses and landmarks. However, some of the same challenges apply. The system has to isolate recommendations that are appropriate for the user's context and must incorporate prior user feedback in order to personalize results. Other related areas include multimedia retrieval, which involves retrieving items that are inherently associated with text. There is also work recent work on task and user modeling whereby models of users' active search interests and intentions are inferred from their search behavior and used to predict future activity or select relevant search results.

Another area related to app recommendation is *mobile search*. The advanced functionalities of mobile computing devices (built-in cameras, scanners, position aware micro-sensors such as GPS, digital compasses and accelerometers) coupled with the availability of fast internet connections through 3G networks create opportunities for developing mobile web IR systems that are capable of responding to user needs 'on the go'. Limitations in screen real estate, processing speed, and lower user tolerance to bad design, provide motivations for exploring novel mobile user needs and interaction paradigms, where IR and HCI know-how must be considered alongside. Several spatially aware mobile applications that can search for information in the proximity of a point of interest exist (e.g. AroundMe for iPhones[3] and GeoVector[4]). App recommendation in mobile devices is a challenging research subarea in that domain.

Day 2 – Preliminary Study Design

During our Day 2 discussion, we set out to design a user study to investigate how two factors affect users' willingness to accept an app recommendation: task completion stage and prior knowledge about the task. More specifically, the study is motivated by the following three research questions:

- RQ1: Are users more likely to accept an app recommendation during different stages of task completion?
- RQ2: Are you users more likely to accept an app recommendation when they have greater prior understanding of the sequence of steps required to complete the task?

[3] http://www.aroundme.com/
[4] http://www.geovector.com/

- RQ3: Is there an interaction effect between these two variables? That is, does prior knowledge of the task steps moderate the effect of task completion stage on a user's willingness to accept a task recommendation?

The study would have two experimental variables: (1) task completion (early, middle, late) and (2) prior knowledge (yes, no). The study would also have four outcome measures: (1) app recommendation acceptance (Did the user accept the app recommendation and use it to complete the task?), (2) post-task satisfaction, (3) task completion time, and (4) effort (measured as the number of actions required to complete the task).

The study could have a within-subject design. Each participant would complete six tasks using a mobile device. For three of the tasks, the participant would have prior knowledge of the steps required to complete the task. For the other three, the participant would have no prior knowledge. For two of the tasks, the app would be recommended in the first stage, second stage, and third stage, respectively. Participants would have the option of completing the task using the recommended app, or to complete the task using a set of baseline mobile tools familiar to the participant.

Our main hypothesis is that users are more likely to accept an app recommendation in the early stages of task completion (but not the first stage since they may be happily focused on their task and not in need of assistance). We also hypothesize that prior knowledge of the stages required to complete the task moderates the likelihood that they will accept the app recommendation early. In other words, we conjecture that a user may be more likely to accept the recommendation if they have prior knowledge of the task stages and are more aware its complexity *a priori*.

The following are some possible example tasks:

cook:
1. pick cuisine (early)
2. get recipes (middle)
3. find ingredients (late)

restaurant:
1. find restaurant (early)
2. book table (middle)
3. get directions (late)

travel:
1. choose destination (early)
2. book travel (middle)
3. find things to do (late)