A Reporting Framework for Search Session Evaluation

Cathal Hoare  
Computer Science Department, University College Cork, Ireland. Email: hoare@cs.ucc.ie.

Humphrey Sorensen  
Computer Science Department, University College Cork, Ireland. Email: sorensen@cs.ucc.ie

Abstract  
Mobile devices have become ubiquitous, admitting a range of new contexts for information access. Indeed, these devices are now becoming a significant means of conducting information seeking even where desktops and other large screen devices are available. This has required the development of new design patterns that cater for the advantages and disadvantages presented by these devices’ sensors and smaller screens. In turn, understanding how these new features effect information seeking has required development of new evaluation frameworks. This paper presents one such framework, as well as describing our experience when developing and evaluating mobile search user interfaces.

Keywords: search user interface evaluation, reporting tools, mobile user search interface evaluation

Introduction  
Mobile devices have become ubiquitous over the past number of years. Their high rate of adoption is predicted to grow as smartphones and tablets become more affordable\footnote{http://www.gartner.com/newsroom/id/2610015}. These devices are characterised by their portability, startup speed, connectivity and range of sensors including GPS, cameras and motion sensors. These characteristics have admitted new information access features such as query-by-speech/sound (e.g. Shazam and Siri) and query-by-image (e.g. Google Goggles) (Hearst, M. A. (2011)). Their sensors have also facilitated new forms of information presentation that leverage the user’s context to organise information (e.g. presentation of landmarks on a map) (Church, K. et al. (2010)). Connectivity has admitted a social search context where online communities can be used to answer an information need instead of accessing a search engine (Church, K. et al. (2012)). These developments have arrived quickly as developers rush to utilise new hardware features and gain commercial advantage. The pace and complexity of development has outstripped our ability to develop a deep understanding of how users are using these features to discover relevant information. This is especially true when we seek to understand complex search such as exploratory search.

Evaluation of mobile user search interfaces offers both opportunities and disadvantages. Native mobile applications can provide a view of all user interactions during a search session. In addition, users’ context can be monitored. When examining users’ actions ‘in the wild’ - outside of controlled lab conditions - this provides a valuable insight. Creating evaluation tasks and environments for mobile applications, especially when considering contexts such as location or social interactions, is especially difficult; data collection in these environments is also complicated. Recruiting participants also posses difficulties; expecting users to have their own device incurs costs and requires trust on their part in order to allow you deploy your software to their device, while providing a mobile device requires trust on the evaluators’ part.

This paper describes our experience of evaluating mobile user search interfaces, particularly in support of exploratory search tasks. In pursuit of this goal, we developed an evaluation framework that models user interactions across two related dimensions, gain and process. Gain - the amount of useful information retrieved - is represented through a variation of Charnov’s Marginal Gain Theorem (utilised by Pirolli and Card (1999) in Information Foraging Theory, while process - the steps taken to discover relevant information - is represented by a modified version of the process model described by Marchionini (1995). These views are animated and show the development of users’ actions over time rather than presenting the final state achieved. This is especially important since a feature might impact a particular phase of a search session.
This paper continues by examining exploratory search and the effects of mobile contexts on search. In addition, approaches to evaluation that inspired this work are reviewed. The theoretic underpinning of the model is then introduced before describing how the model is implemented; this will include data collection from mobile devices, cleaning this data and preparing it for examination. The paper will conclude with a description of the evaluation interface.

Background
Mobile information seeking is becoming ubiquitous. Smartphones admit

- Search from new contexts - through location awareness and mobile communications, users can query on the move or from locations where search would previously have been unlikely. This admits a range of new contexts.

- Information seeking (or components of seeking) can be conducted in new ways, for example, through social interactions such as question answering or through new forms of query that accept input from the device’s microphone or camera.

- New modes of presentation that take advantage of users’ context make assimilation of information more intuitive. For example, being geographically aware, query results can be presented on a map.

- The always on, always connected nature of these devices allows a user to integrate information seeking into tasks and admits serendipitous curiosities.

This non-exhaustive list indicates that a host of new forms of query are now in use in a range of contexts by users whose domain and system knowledge varies hugely. While these developments are to be expected, the role of mobile devices in static contexts such as home or office is, perhaps, surprising. These devices are used in static contexts, even when a desktop or laptop is available. The ‘always on’, low boot time, and ‘to hand’ nature of mobile devices mean that they are often the tool of choice when a serendipitous curiosity arises, for example, while watching television. Church et al. (2012) found that 29% of mobile searches captured in a user survey were conducted at home, while 24% took place at the workplace; these findings have been reinforced by surveys conducted by the author.

It is also surprising that these devices are used to conduct exploratory search. Exploratory search is characterised by the need to satisfy several information needs, synthesise them into a piece of knowledge that can be used in support of some greater task. Exploratory search is dynamic, and often characterised by an early exploratory phase where users learn about their task, knowledge space and information need. These discoveries often cause the information need to develop, and inform more focused queries that occur later (as shown in Figure 1, reproduced from White and Roth (2008)). A Nielsen Report2 indicated that many mobile searches are not standalone, but are associated with follow-up actions (including further search). This finding is reinforced by Church et al. (2012), who find that tasks that “assist an activity or task” make up 60% of mobile search tasks captured in their survey. This type of activity is often not well supported by search user interfaces. Mobile search has assumed that a user is mobile while searching and not in a static context. This has, for example, manifested itself through the provision of answers built into search results and the inclusion of maps and other information; this is useful for those on the move, but often useless to those in a static context such as home. These findings indicate a need to support many forms of search on a mobile device, and not just search while mobile; deciding presentation modality based on device type is no longer sufficient. It also indicates the need to develop features and evaluate their impact through the search process.

Evaluation of exploratory search is considered difficult. Many variables impact user actions, and simulating tasks and information domains is complex (Kules, and Capra (2008)). Furthermore, interface and system components must be evaluated while bearing in mind that a component may only improve certain parts of a search session or for search in particular contexts, and may have no impact on others; for example, maps are useful when a user is in a mobile context but may be useless if a user is wholly unfamiliar with a location. It is therefore necessary to carefully construct realistic tasks over a range of contexts. It is also necessary to understand user’s actions and relate these to information gain. These views need to be maintained over the entire lifecycle of the search session.

Several systems have been developed to support exploratory search evaluation. Janson et al. (2006) developed the Wrapper system which was developed to collect user

---

Figure 2: (A) Sequential Search Process Model (B) Combined Model (C) Sample Process Views
interactions across multiple applications and computers and report these to a server where analysis could be conducted. Capra (2011) introduced the HCI Browser system, which provided a management interface for exploratory evaluation, presenting tasks to participants and logging their actions as they complete tasks on web browsers and also presents them with pre- and post-task questionnaires. A system with similar goals, called Search-Logger, was presented by Signer et al. (2011). This system managed deploying tasks to participants, collecting their responses and provided an analysis interface to examine results. Numerous models of exploratory and other types of search have been proposed; these have been examined elsewhere. Individually, they provide a relatively narrow view of a search session from a particular context; together they provide a detailed view of the same session from many perspectives. The power of combining models was demonstrated by Wilson et al. (2008) where two models were combined to provide a deeper view of a search process to identify the strengths and weaknesses of search user interfaces and quantify how well they support various user tactics and strategies.

**Evaluation Model**

The authors chose to combine two established, generic and expressive models to capture both gain and the process followed by participants while executing experimental tasks (shown in Figure 2, Part B). The models are a general seeking model - described by Marchionini (1995) (reproduced in Figure 2, Part A) and a component of Pirolli and Card’s (1999) Information Foraging Theory, called a Gain Model. Together they provide a view of the search process followed by experiment participants and relate this to their rate of successfully finding relevant information. The results can be viewed overtime to gain an insight into the search session’s development and admit views of individual and aggregations of participants. The models can be adapted to highlight use of particular interface features. The combination of models also admits comparisons between results for users or for different versions of a user search interface.

This process model was arranged to convey contextual information about the states and to emphasise observable transitions between them. In addition, this arrangement helps to make the gain graph above the process model more meaningful by capturing ‘organisational actions’; these are actions that locate and organise retrieved information, while ‘analysis tasks’ are actions concerned with information gain. The accurate state was an addition to the original model. Intended to capture note taking or other information collection activities - a common feature of our mobile applications - this state is an example of how the model can be easily modified to highlight feature types. Transitions are associated with particular sequences of interface actions taken by the user. For example, formulating a query, entering text into a search field, submitting a query, and presenting a series of SERPs would pass through the (re)define/formulate query and formulate query/examine transitions. The model reports on the percentage of each transition type made; this indicates how the search process evolved. A more complete explanation of transitions can be found elsewhere (Hoare and Sorensen, 2010).

Part C of Figure 2 presents some typical search patterns displayed by participants. Typically, during the Exploratory Browsing phase (depicted in Figure 1), users were seen to conduct shallow, rapid searching, formulating queries, brief examining results and either redefining their queries or formulating a new query. Once they have gained an insight into the task, domain and system, they begin to form targeted, exploratory queries. These produce patterns similar to the next process map, where results are examined, information is extracted and relevant information is recorded. This information is used in turn to redefine and evolve queries. The final part of that diagram demonstrates two other patterns that were observed during evaluations. The first, shown as a sequence of blue arrows demonstrated a tendency by some users to rapidly formulate new queries when the first few SERPs returned failed to satisfy their information need. The rate of redefinition was high, often with terms being added in an unplanned way and with little recourse to information retrieved up to that point. Another

![Figure 3. Charnov’s Marginal Gain Theorem](image-url)
ineffective strategy was observed where users paged through results without modifying their query; convinced that their query was correct, these users believed that the system was at fault for not satisfying their query. The ability to observe these patterns admits the possibility of allowing the search system to intervene and recommend other queries or strategies to the user; this remains as future work.

The gain model is derived from a component of Pirolli and Cards’ (1999) Information Foraging Theory. Foraging Theory attempts to explain information seeking behaviour in humans by comparing it to food foraging mechanisms in nature. Here patches of food are analogous to patches of relevant information in an information space. Some patches are more nutritious than others, while others cost more effort to locate and harvest for information. A patch can be exhausted, resulting in no new information being located; this is the point when a seeker should move to another patch. Optimal foraging occurs when the seeker stays in a patch just long enough to consume its nutritious content, before moving to another patch to continue foraging. Chernov’s Marginal Gain Theorem is used to describe the state of foraging in a particular patch. Gain is represented by the area beneath the curve in Figure 3, Part A, while the cost of harvesting that information is the time expended both within patches and seeking those patches. Thus, the rate of gain achieved is equal to the slope of tangent R\(^+\) (Figure 3, Part A). Two types of enrichment can occur, prevalence and profitability. Prevalence can be increased by decreasing the time spent seeking relevant information. This is analogous to creating queries with high recall - a desirable state when conducting the initial Exploratory Browsing phase of exploratory search. Profitability occurs when patches with high nutritional value are browsed; this increases the rate of gain. This is analogous to high precision queries, preferred for the focused search phase of exploratory search. Thus, it is desirable to see a process where initial sequences provide high prevalence and admit queries that provide high profitability. Gain is represented in our visualisation as graph depicting recall over time and precision over time. Other metrics are being investigated.

**Implementing the Model**

We will now examine how this evaluation framework was implemented. The resulting system was composed of five functional areas:

- **Experiment Setup, Deployment and Management** - this component admitted marshaling of metadata about application versions and participants into a database that informs the user segmentation component of the reporting interface.

- **Deployment Function** - in all cases this functionality was managed by Apple’s Developer portal and is concerned with deploying features and applications under evaluation to participant’s mobile devices.

- **Collection of User Metrics** - this component is catered for by Google Analytics for iOS Native Application Tracking which collects user interactions with the app under evaluation.

- **Data Cleaning** - takes sequences of events form the Google Analytic’s repository and translates these into evaluation metrics that can be visualised on the reporting interface.

- **Reporting Interface** - this component consists of a user interface that allows an evaluator to segment cleaned results and visualise these through the gain and process models; the models are represented over time, producing an animated representation of both actions and their effects over an entire search session.

We will now examine each of these stages in greater detail.

The first element of the framework is experiment management (step numbered 1 in figure 4). Experiments that evaluate mobile search interface features must be carefully managed; software versions, participants and contexts are recorded to provide metadata to the reporting interface to allow fine grained user segmentation. It is particularly important to manage software versions and ensure that the correct version is deployed to participants’ phones. The interface components developed by the authors have targeted Apple’s mobile devices, the iPhone and iPad. Code is developed in Apple’s XCode development environment, and applications are deployed through creating an ad-hoc provisioning profile that allows an application to be run on a specific set of devices. This profile and an application deployment bundle can be sent to an experiment participant with instructions on how to deploy these files to their phone through Apple’s iTune’s programme (step 2, figure 4). In addition to recording metadata and managing software deployment, the framework requires experimenters to develop an interface model that maps sequences of interface component use to transitions in the evaluation framework; for example, forming a query using a search box could look like:

```
<textentry:box1::buttonpressed:button1>
```

This would translate to a transition of type 1 - ‘formulate query to examine’ (see figure 2). These labels need to be associated with the interface components during the development process. This is achieved using Google Analytic’s iOS Native Application Tracking development kit.

Google Analytics provides a large set of tools for understanding user interactions with mobile applications. This includes the ability to capture user interactions with the interface. Method calls are added to event handlers for interface components. These include information identifying the component, the action carried out and
information, including a participant’s unique identity and other metadata, about the interaction. These events are written to Google’s collection servers, which can subsequently be accessed from our server through a programatic reporting interface (number 3 in figure 4). Having queried events for an experimental run (using metadata from the experiment management component and user segmentation parameters from the reporting interface), the system must then translate these into a form that can be presented on the reporting interface (number 4 in figure 4). This is done by the ‘cleaner’ component which takes the interface model defined in step 1, and creates a parsing tree which generates a script of transitions that are displayed in the process model. Average Recall and precision measurements are also calculated at regular time intervals for the user segments defined in the reporting interface.

The final element (step 5 of Figure 4) of the framework presents the evaluation model. The producer component consumes the script and meta-data files to produce timed events that are presented on both process and gain models on the interface. The producer is controlled by the playback controls on the interface, controlling speed and other playback features. The interface also provides a control to select experiment runs and fine-grained user segmentation.
This control produced parameters for the ‘cleaner’ component. We will now examine the interface itself in greater detail.

**Reporting Interface**

The reporting interface is composed of three functional parts: a data segmentation feature, a visualisation of the model and interface controls. The interface itself is written in Javascript and HTML/CSS and runs in a web browser, connecting to a server that hosts the producer component that generates script events. The data segmentation component allows users to be segmented according to various rules. Data for individual participants, or whole participant sets for an experiment can be sampled. These can be further divided by imposing rules on the set. For example, individual participants can be filtered by id or by their recall and precision scores. Participants can be chosen from the set meeting these criteria. Similarly, aggregations of participants can be created by imposing similar rules; for example, the interface can generate a report for all participants with a recall score less than some value. Two data segments can be reported on at any time, allowing comparison between the two.

The reporting interface displays the process and gain model described earlier. This visualisation is animated, showing the development search sessions for one or two data segments over time. Each transition in the model is colour coded to indicate the frequency with which it is transited; these can also be clicked on to reveal a popup that provides more detailed statistics for that transition. Two metrics can be shown on the gain model at any one time - for example, precision and recall. The playback controls allow the animated report to be paused and admit adjustments to playback speeds. Snapshots of the model can also be taken for further investigation later.

**Conclusions**

This paper has presented the development of an evaluation framework for mobile exploratory search interfaces. The framework presents the development of a search session over time through the lens of two models representing gain and process. Potential insights provided by this combination were presented. The insights provided by this visualisation admit improvements to the search interface. An implementation of this model was also presented. Implementation of a framework to support data collection and cleaning in support of the model was also presented, as was the interface used to partition experiment data and present results to evaluators.

**REFERENCES**


Curriculum Vitae

Cathal Hoare is a PhD student in Computer Science at University College Cork. He graduated with a BSc in Computer Science from UCC in 1998, after which he worked as a software engineer at Motorola and Comnitel Technologies. On returning to UCC he began to work on applying the benefits of the sensors available on an smartphone to improve search user interfaces by creating query-by-image query interfaces. He has been published widely in the areas of user interface evaluation and search interface design. He has also worked with local companies to conduct early stage research on new products through a variety of Enterprise Ireland grants.

Humphrey Sorensen is a Senior Lecturer in Computer Science at University College Cork, where he has worked since 1983. He was educated at University College Cork (B.E., M.Sc.) and at the State University of New York at Stony Brook (M.S.). He has also worked at the University of Southern Maine and at Colby College. He teaches in the area of database and information systems. His research has largely been in the area of information retrieval, filtering and visualization, where he has collaborated with industrial partners on several funded projects. Latterly, he has researched and published in the areas of multi-agent approaches to complex information tasks, and in the broader fields of artificial life (AL) and multi-agent systems (MAS). He has supervised several M.Sc., PhD and Post-doctoral researchers within these areas.