CrowdLogger as a Community Platform for Searcher Behavior Experiments

Henry Feild

Center for Intelligent Information Retrieval University of Massachusetts Amherst

November 28, 2012







Things we like to do in IR

Observe and model user behavior

Modeling and Measuring the Impact of Short and Long-Term Behavior on Search Personalization

Personalization of Search Results Using Interaction Behaviors in Search Sessions

Improving Searcher Models Using Mouse Cursor Activity

Search, Interrupted: Understanding and Predicting Search Task Continuation

User Evaluation of Query Quality

- Compare search algorithms / interfaces
 - which do users prefer?
 - time to completion
 - which result in more/fewer clicks, etc.

Absence time and user engagement: Evaluating Ranking Functions

Optimized Interleaving for Online Retrieval Evaluation

What's currently done

software:



- make a toolbar from scratch
- modify the Lemur Search Log Toolbar

study:

- recruit some users and conduct controlled lab study
- install on campus computers, observe users in situ
 - well, in situ specifically in a library setting
- This is slow, expensive, and generally a lot of effort

What we want

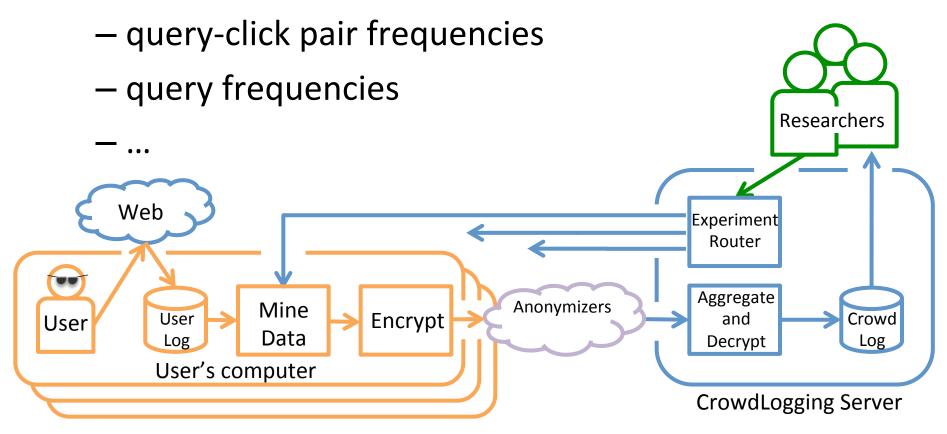
- a common, open source platform that deals with the basics
 - interaction data collection
 - data storage
 - privacy
- a common user base
 - can recruit some new users, but already have a significant pool of participants
- an interface for implementing novel studies



CrowdLogger overview



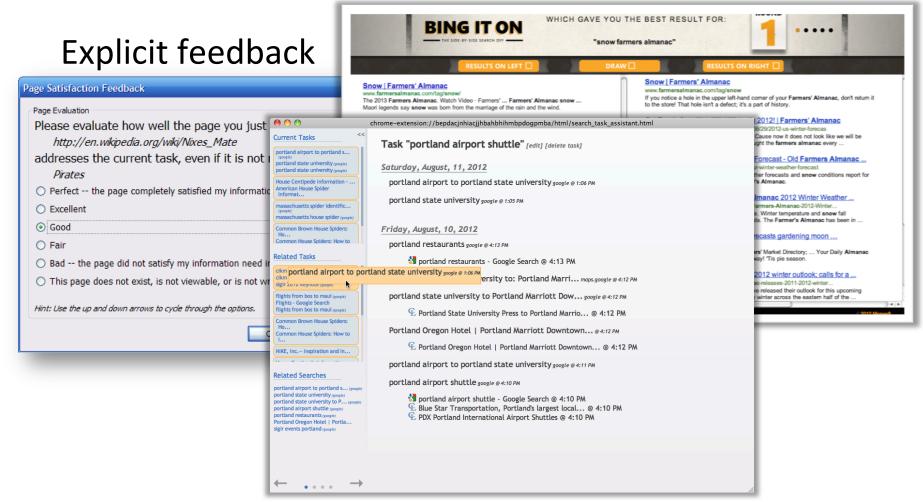
- Current a private data aggregation system
 - query reformulation pair frequencies





2.0 Interactive experiments

System comparisons



Labeling

Challenges

- data management across multiple experiments
- an API that allows researchers sufficient control over accessing user data and implementing experiments
- controlling what data is shared with researchers
- incentivizing users to download the extension and participate in experiments

Data management

What to log

browser interactions

- add/remove/move tab
- back/forward buttons
- favorites/home/minimize/exit ...

web page interactions

- page un/loads
- page focus
- clicks
- scrolls
- mouse movements

SERP interactions

- query
- top 10 results
- urls, summaries, etc.

complex interactions

- opening links in new tabs
- search tasks
- study data

How to log it

Lemur Toolbar format

```
SEARCH 1354022000 google wikipedia

CLICK 1354023824 wikipedia.org google.com/search? q=wikipedia

LOAD 1354023900 wikipedia.org

...
```

A tab-delimited text file

CrowdLogger format

```
{event: search, time: 1354022000, query: wikipedia, se: google} {event: click, time: 1354023824, destUrl: wikipedia.org, srcUrl: google.com/search?q=wikipedia} {event: load, time: 1354023900, url: wikipedia.org} ...
```

Data management

Benefits of JSON: easily extensible

{event: search, time: 1354022000, query: wikipedia, se: google}

```
{event: search, time: 1354022000, query: wikipedia, se: google, results: [{rank: 1, title: Wikipedia, url: wikipedia.org, snippet: "Wikipedia, the free encyclopedia that anyone can edit.}, {rank: 2,...}]}
```

Benefits of IndexedDB

- versioning built in
- entries can be updated in place
 - no need to re-write entire log file
- can build multiple indexes over data store
- HTML5 standard

API Categories

User Data



Historical data

- get all clicks
- get all searches

Real time data

- on new search, do ...

Aggregate User Data



Already collected data

- get all query rewrites
- get all query-click pairs

User Interface



Add to CrowdLogger interface

- add widget to tools page

Stand-alone windows/pages

- present dialog when user searches
- modify search page ranking

Uploading/Privacy



Encryption
Anonymization & aggregation

- upload via anonymizers
- privately aggregate data

Client-server communication



Request server-side computation

- run retrieval algorithm for query

Access server-side data

- send me synonyms for ...

API Layer Options



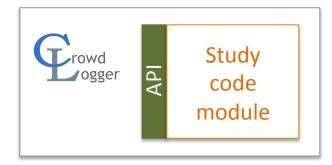
Communicate via inter-extension event calls

Pros:

- flexible study extension

Cons:

- limited communication
- no control over extension
- user has to download separate extension



CrowdLogger executes study code module

Pros:

- single module
- browser independent

Cons:

- remote JavaScript execution
- requires code approval
- potentially complex study code formulation
- less study code flexibility

Privacy controls

- what data get's shared with researchers?
- under what conditions?

What data is being collected and how it will be used

What is minimally useful to researchers

What users are comfortable with

Query rewrites for public release

Whatever users are comfortable with

User 1: only if shared by 9+ other users (k=10)

User 2: k=1 rewrites

Feedback on retrieval system preference for researcher use only

k=1 anonymized feedback from users

User 1: k=5 feedback
User 2: k=1 feedback

Privacy controls

What will be collected:

All search reformulations. For example, if you search for "blueberry pie" and then "blueberry pie recipes", the pair: "blueberry pie", "blueberry pie recipes" will be collected.

How the collected data will be used:

Reformulations will be anonymized and made **publically accessible** and used to, for example, generate search suggestions for you and other users.

Privacy settings:

For each search reformulation collected from you, select the anonymization level: the number of other users that must also share the same reformulation for it to be included in the final data set:

I have read the <u>consent</u> form and agree to participate in this study.

Cancel

Continue

Incentivization

Provide a service

- research prototypes
- visualizations
- re-finding tools
- citizen scientist



Google Search History



Search Task Assistant

YAHOO! ANSWERS



Financial incentives

- gift cards
- virtual currency to 'buy' research apps

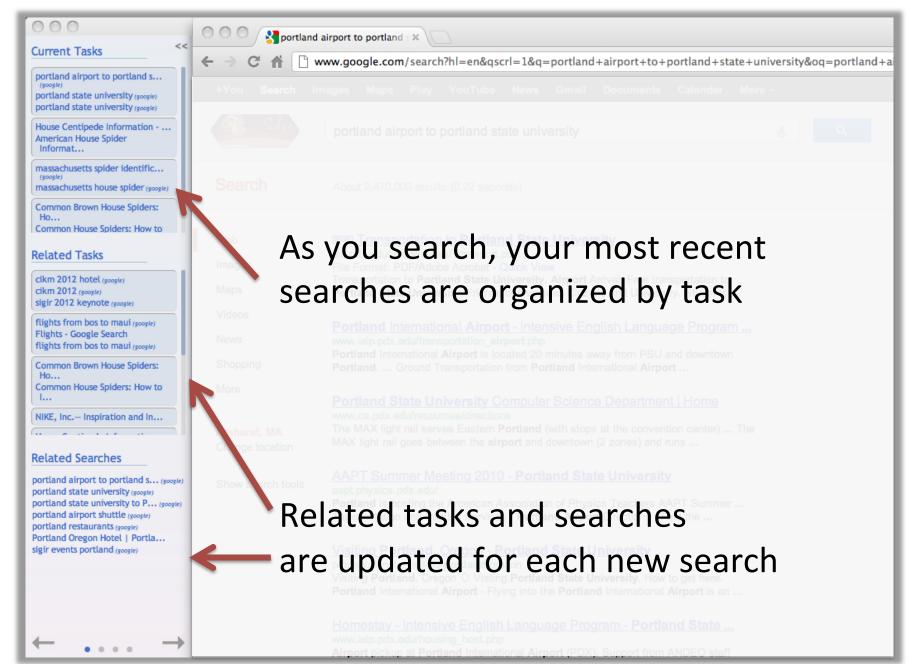
Gamification

- study-specific
- could also be a service

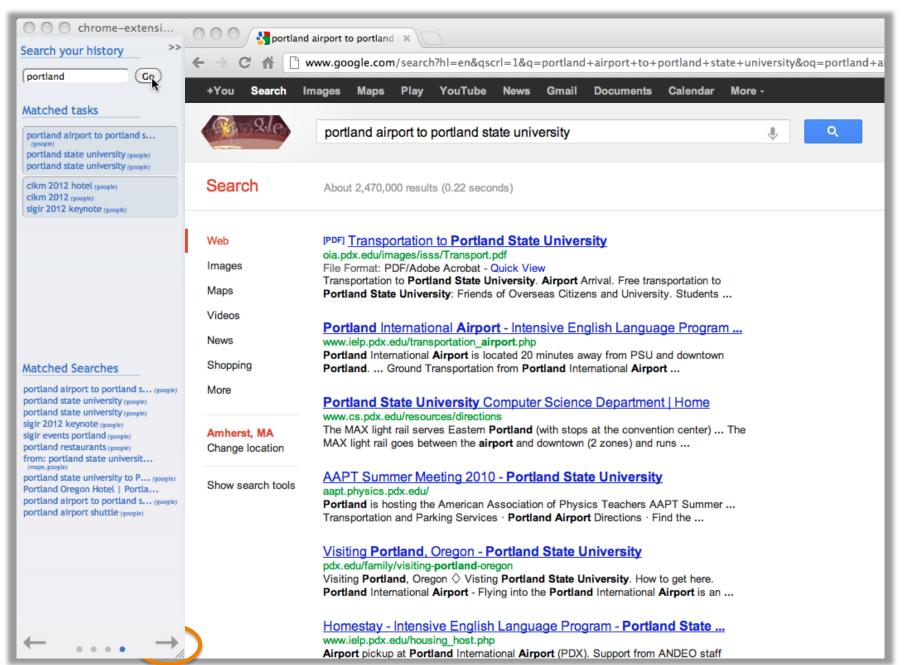


Google-a-day

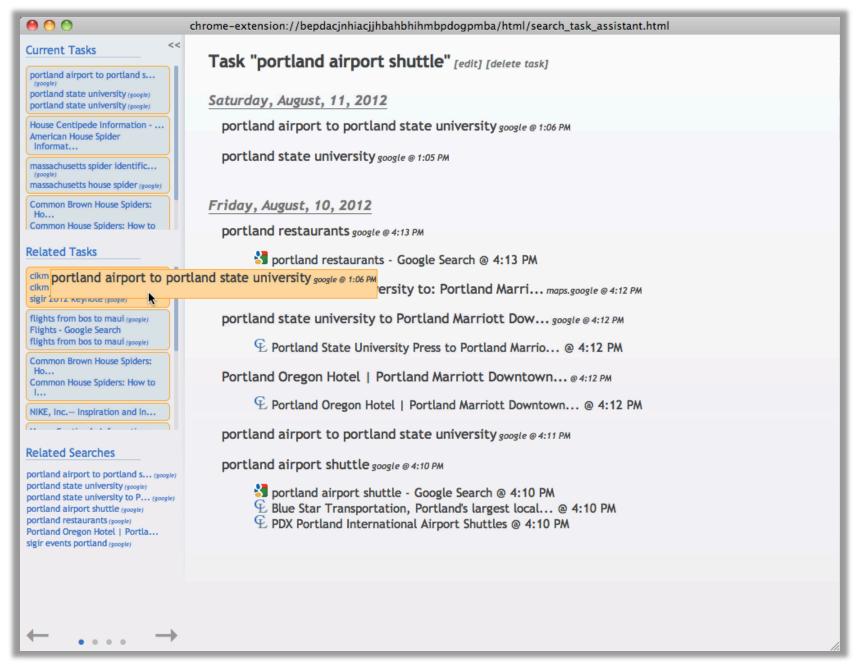
Service: Search Task Assistant



Service: Search Task Assistant



Service: Search Task Assistant



Use case example

- evaluate a query recommendation algorithm
 - APIs:



Extract query reformulations from users

User Data



User Interface



 When a user enters a new query, modify the search page to include a list of query suggestions

 Ask user to rate the suggestions Client-server communication



On each new query, contact a server to compute the recommendations

Uploading/ Privacy



 k=5 for query reformulations

- k=1 for feedback

Use case example: Privacy consent

What will be collected:

Your feedback about the quality of search suggestions generated by our algorithm. In addition, you will have the option of providing the search you entered along with the feedback.

How the collected data will be used:

The feedback will be used only by the researchers in charge of this study (listed in the consent form).

Privacy settings:

For each set of feedback collected from you, select the anonymization level: the number of other users that must share the same feedback for it to be included in the final data set (max to participate: 1):

I have read the <u>consent</u> form and agree to participate in this study.

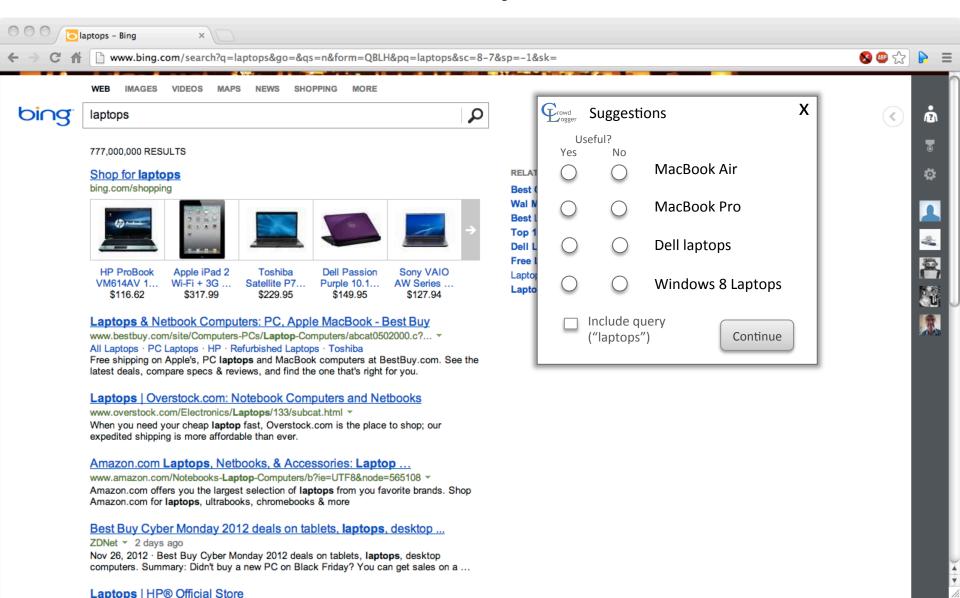
Cancel

Continue

Use case example: code snippet

```
api.userData.onQuery( function(data) {
   var suggestions = getSuggestions(data.query);
   api.ui.addOverlay(queryData.page, showSuggestions);
});
function getSuggestions(query){
   return api.clientServer.callServer(
      QUERY_SUGG_SERVER_URL, query);
function showSuggestions(page, suggestions){
   var overlay = page.jQuery("<div>");
   button.click(onFeedbackSubmitted);
function onFeedbackSubmitted(data){
   api.upload.private(data.toString, {k: 1});
```

Use case example: interface



Other challenges

- cross-device synchronization
- mobile device support
 - requires browsers to allow extensions on mobile platforms
- neutral code review panel