Untangling Result List Refinement and Ranking quality:

A Framework for Evaluation and Prediction

Jiyin He, Marc Bron, Arjen de Vries, Leif Azzopardi, and Maarten de Rijke
Batch Evaluation

- **Cost effective** evaluation: prediction of search effectiveness based on a series of assumptions on how users use a search system

- Requirements:
  - A collection of documents
  - A set of test queries
  - Relevance judgements
  - An evaluation metric
Evaluation metrics and user interaction

• Evaluation metrics (Carterette, 2011)
  • A user interaction model: How users interact with a ranked list
  • Associating user interactions with effort or gain
  • Current batch evaluation metrics: boils down to the ranking quality of the results
Beyond a ranked list

**Q**: how do we evaluate and compare systems under varying conditions of **ranking quality**, **interface elements**, as well as different **user search behaviour**?
Our solution

• An effort/gain-based user interaction model
  - How users interact with a ranked list and the RLR elements
  - Associating user interactions with effort and gain

• Applications
  - **Prediction**: system performance w.r.t. a particular application and user group
    - Model parameters derived from user studies
  - **Simulation**: whole system evaluation under varying conditions: ranking quality, interface elements, user types
    - Model parameters based on hypothesised values
Modelling user interaction: with a ranked list

E.g., following assumptions of user behaviour as in RBP

Parameter: continuation

Decision point: when to stop?
Modelling user interaction: with result list refinement

- Result refinement = switching between different filtered versions of the ranked list (sublists)

Decision point:
- to stop browsing?
- to switch or to continue examining?
- which one to select next?

Combinatory number of possible user paths:
- A Monte-Carlo solution
Modelling user interaction: with result list refinement

• Action path constraints
  - In each sublist, users browse top-down
    - common assumption; reducing possible paths from n! to constant
  - Users skip and only skip documents already seen
    - preventing inflated relevance and infinite switching
  - Deterministic quitting point
    - gain based: quit when certain amount of effort is spent
    - effort based: quit when certain amount of gain is achieved
Modelling user interaction: with result list refinement

- **Action path constraints**
  - In each list, users browse **top-down**
  - Users **skip and only skip** documents already seen
  - **Deterministic quitting point**
    - gain based: quit when certain amount of effort is spent
    - effort based: quit when certain amount of gain is achieved

![Diagram]

Parameter 1: continuation

Parameter 2: List selection
User actions, efforts, and gain

• From user action paths to user efforts and gain
  - Each action is associated with an effort
  - Each action may or may not result in a gain, i.e., finding relevant document

• User actions
  - Examine result, refine a list, pagination

• Simple assumption about effort and gain
  - Equal unit effort for all actions  \[ \text{Total effort} = \# \text{ actions} \]
  - Equal unit gain for all relevant documents  \[ \text{Total gain} = \# \text{ relevant docs found} \]
Validation of prediction

• RQs
  - Does the predicted effort correlate to user effort derived from usage data?
  - Can we accurately predict when a RLR interface is beneficial, compared to a basic interface?

• 3 Steps
  - Obtaining usage data from user study
  - Measuring (real) user effort
  - Predicting user performance by calibrated user interaction model

• Data
  - TREC 2013 Federated Search track
  - 50 topics with retrieved web pages and snippets, all judged
  - Results from 108 verticals, each associated with one or more categories
Obtaining usage data: study design

- User task (He et al., 2014)
  - Finding 10 relevant documents
    - Manageable effort, potential for considerable effort save
  - within 50 clicks
    - Preventing randomly clicking all results
  - Snippet based relevance judgement with user feedback
    - Reduced user variability in relevance judgement

- Experiment design
  - Between subject
  - Randomised topic and interface assignment
Obtaining usage data: interfaces

Basic interface

RLLR interface
# Obtained usage data

<table>
<thead>
<tr>
<th></th>
<th>Basic</th>
<th>RLR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Completed task instances</td>
<td>145 (Median p. task: 2)</td>
<td>255 (Median p. task: 3)</td>
</tr>
<tr>
<td>#Participants</td>
<td>49</td>
<td>48</td>
</tr>
<tr>
<td>#Uncompleted task instances</td>
<td>35</td>
<td>28</td>
</tr>
</tbody>
</table>
Measuring (real) user effort

- Examine result: mouse hover over a result snippet

  - # results visited on a SERP =
    - all results in a page before a “pagination” action +
    - up to the last clicked result on the last visited page

Mild position bias: as a result of snippet-based result examination
Predicting user effort with calibrated interaction model

Parameter 2: List selection
- Per topic, the relative frequency that a filter is chosen
- Default selection: “All categories”

Parameter 1: continuation
Probability a result is visited @ rank $K$
Q1: Does the predicted effort correlate to user effort?

- Predicted effort: an approximation of real user effort

  Correlation as a measure for the accuracy of approximation

Pearson correlation between the predicted effort and user effort: \textbf{0.79} (p-value < 0.01)
Q2: Can we accurately predict when a RLR interface is beneficial?

- **Basic user effort - RLR user effort** (difference between user effort on two interfaces)
- **Basic user effort - RLR predicted effort** (difference between actual user effort on basic interface and predicted user effort on RLR interface)

### Accuracy of prediction

<table>
<thead>
<tr>
<th></th>
<th>P</th>
<th>R</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td>0.85</td>
<td>0.55</td>
<td>0.66</td>
</tr>
<tr>
<td>RLR</td>
<td>0.52</td>
<td>1.00</td>
<td>0.68</td>
</tr>
</tbody>
</table>
Validation of prediction: conclusions

• Our RLR user interaction model is able to accurately predict user effort

• Different interfaces are suitable for different queries (i.e., of different ranking quality)

• Model allows prediction of which interface is most suitable
Whole system evaluation: hypothesised users

• RQs
  - When does an RLR interface help to save user effort compared to a basic interface?

• Study whole system performance under varying conditions:
  - Ranking quality
  - Sublist characteristics
  - User behaviour
Hypothesised user parameter setting

• Intuition: some users are more patient than others

• Parameter 1: continuation
  • at each rank $r$, draw a decision as a Bernoulli trial
  • Bernoulli parameterised by an exponential decay function to approximate the empirical distribution of rank biased visit

$$P(E = 1 | r, u) = e^{-\lambda r}$$
Hypothesised user parameter setting

- Intuition: some users make better selection of sublists than others

- Parameter 2: list selection
  - draw a decision vector from a categorical distribution
    \[ f_1, \ldots, f_k \sim \text{Cat}(K, \bar{c}) \]
  - setting user prior knowledge of the candidate lists with its conjugate prior
    \[ \bar{c} \sim \text{Dir}(K, \bar{\alpha}) \]

**Uniform**: no idea what to select

**NDCG**: informed selection
Factors influencing RLR effectiveness

- Query difficulty for the basic interface (Dq)
  - Efforts needed to accomplish a task with basic interface
- Sublist relevance (Rq)
  - Averaged NDCG score over sublists of a query
- Sublist entropy (Hq)
  - Entropy of relevant documents distributed among sublists
- User accuracy (U)
  - Controlled by the amount of smooth added to the prior of list selection
  - Level 1 (oracle based on NDCG);
  - 15% (level 2), 50% (level 3), 67% (level 4) less accurate compared to level 1
- User task
  - to find 1, 10, or “all” relevant documents
Method

- Fit a generalised linear model (logistic regression)

- **DV**: whether a RLR interface outperforms (i.e., save efforts) a basic interface

- **IVs**: factors outlined above

- **Model selection**: forward and backward selection with Bayesian information criterion (BIC)

- Explain the relation between DV and IVs and their interactions
**Main effects**

- **Find - 1**: none of the main effects are significant
- **Find - 10 / all**: users need to know which sublists to pick
- **Find - all**: having sublists with relevant documents ranked high is useful.

<table>
<thead>
<tr>
<th>Coefficients</th>
<th>Find-1</th>
<th>Find-10</th>
<th>Find-all</th>
</tr>
</thead>
<tbody>
<tr>
<td>intercept</td>
<td>-7.340</td>
<td>-10.437</td>
<td>-0.534</td>
</tr>
<tr>
<td>Dq</td>
<td>0.106</td>
<td>-0.069</td>
<td>0.002</td>
</tr>
<tr>
<td>U-level2</td>
<td>3.223</td>
<td>-2.131</td>
<td>-5.106</td>
</tr>
<tr>
<td>U-level3</td>
<td>1.559</td>
<td>-5.528</td>
<td>-8.014</td>
</tr>
<tr>
<td>U-level4</td>
<td>-2.319</td>
<td>-8.194</td>
<td>-8.014</td>
</tr>
<tr>
<td>Hq</td>
<td>-1.044</td>
<td>3.635</td>
<td>-1.649</td>
</tr>
<tr>
<td>Rq</td>
<td>-</td>
<td>-49.792</td>
<td>114.940</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Interaction</th>
<th>Find-1</th>
<th>Find-10</th>
<th>Find-all</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dq : U-level2</td>
<td>-1.655</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dq : U-level3</td>
<td>-2.004</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dq : U-level4</td>
<td>-2.068</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Dq : Hq</td>
<td>1.310</td>
<td>-0.097</td>
<td>-</td>
</tr>
<tr>
<td>Dq : Rq</td>
<td>-</td>
<td>3.263</td>
<td>0.091</td>
</tr>
<tr>
<td>Hq : Rq</td>
<td>-</td>
<td>13.968</td>
<td>-57.277</td>
</tr>
<tr>
<td>Dq : Hq : Rq</td>
<td>-</td>
<td>-0.842</td>
<td>-</td>
</tr>
</tbody>
</table>
Interaction effects

Dq: Rq for Find -10

- When query is difficult for basic interface, sublists and users do not need to be very accurate for RLR to be more effective.
- When query is easy for basic interface, higher quality of sublists and user accuracy are necessary.
Interaction effects

Dq:Rq:Hq for Find -10

(a) Dq: high; Rq: high
(b) Dq: high; Rq: low
(c) Dq: low; Rq: high
(d) Dq: low; Rq: low

- When query is difficult for basic interface, RLR is likely to be beneficial especially when few sublists contain most of the relevant documents.

- When query is easy for basic interface, very specific conditions with respect to user accuracy, sublist relevance, and sublist entropy need to be met for RLR to be beneficial.
Relation to traditional metrics

<table>
<thead>
<tr>
<th>Metric</th>
<th>user effort</th>
<th>predicted effort</th>
<th>Δ user effort</th>
</tr>
</thead>
<tbody>
<tr>
<td>nDCG@10</td>
<td>-0.21</td>
<td>-0.19</td>
<td>0.02</td>
</tr>
<tr>
<td>nDCG@all</td>
<td>-0.42*</td>
<td>-0.34</td>
<td>0.00</td>
</tr>
<tr>
<td>NRBP</td>
<td>-0.41*</td>
<td>-0.33</td>
<td>0.08</td>
</tr>
<tr>
<td>AP</td>
<td>-0.63**</td>
<td>-0.54**</td>
<td>0.02</td>
</tr>
<tr>
<td>binary nDCG@10</td>
<td>-0.54**</td>
<td>-0.44**</td>
<td>0.02</td>
</tr>
<tr>
<td>binary nDCG@all</td>
<td>-0.72**</td>
<td>-0.59**</td>
<td>0.04</td>
</tr>
<tr>
<td>Our model</td>
<td>0.79**</td>
<td>—</td>
<td>-0.49**</td>
</tr>
</tbody>
</table>

Pearson’s linear correlation; p-value < 0.01 (*); <0.001 (**)  

- Query difficulty alone is not sufficient to predict whether a RLR interface will be beneficial
Whole system evaluation: conclusions

• When ranking quality is low, sub-lists and user sublist selection do not have to be of high quality for RLR to be more effective than basic;

• When ranking quality is high, only under specific conditions RLR may be beneficial, i.e., quality sub-lists, recall oriented task, accurate users;

• Implication for HCIR experiments: are your queries, user tasks, and ranking algorithms appropriate to study the properties of the interface?
Conclusions

• A user interaction model for evaluating search systems with result refinement elements

• By instantiate the model with parameter values derived from real usage data, we have validated the predictive power of our model

• By simulating users with hypothesised parameter values, we can investigate whole system performance under varying conditions concerning ranking quality, interface differences, user types, and task types