PITT at TREC 2013 Session Track

Different Effects of Click-through and Past Queries on Whole-session Search Performance

Jiepu Jiang (University of Massachusetts Amherst)
Daqing He (University of Pittsburgh)
• Analysis of an old method using alternative evaluation approaches
  • Are we really improving the performance?
  • Whole-session relevance?
  • Past query vs. click-through
MOTIVATION

• Using past queries and past click-through data as relevance feedback
  • Pretty old idea
    e.g. context-sensitive RF (Shen, Tan & Zhai, SIGIR ’05)
  • Seemingly very good performance
    • e.g. our systems in 2011 and 2012 (a variant of context-sensitive RF) were ranked at the top (by nDCG@10 of the last query)
MOTIVATION

- Using past query and past click-through as relevance feedback
  - Probably making results similar to previous results
MOTIVATION

• Are we really improving the performance?
  • The improvement of nDCG@10 may come from retrieving relevant documents found by previous queries?

• We cannot answer the question without
  • using whole-session evaluation methods
  • considering novelty in evaluation
**ALTERNATIVE EVALUATION**

- Evaluate whole-session search performance

- **Procedure**
  - A static session \{q_1, q_2, \ldots, q_n\}
  - For each \(q_k\), generate results \(R_k\) based on \{q_1, \ldots, q_k\}
  - Evaluate \{R_1, R_2, \ldots, R_n\} for whole-session performance

- **Simulation of user querying behavior: no simulation**
  - User will not change the next query according to the previous results of systems & behaviors (e.g. click).
METRICS

• Macro-average nDCG@10

\[
\frac{1}{m} \cdot \sum_{i=1}^{m} \left( \frac{1}{n-1} \cdot \sum_{j=2}^{n} nDCG@10(R_{ij}) \right)
\]

• Starting at the 2nd query of each session
METRICS

• nsDCG@10
  • Concatenate top 10 results of each query
  • Combine as a whole rank list for evaluation
    
    *see details in session track overview of 2010*

• There are more complex methods
  • Kanoulas, Carterette, D Clough, & Sanderson in SIGIR’11
METRICS

• Instance recall (instRec)
  • Used in old TREC interactive tracks
  • An instance is similar to a “nugget”
  • instRec measures the recall of all judged relevant instances (nuggets) all over the session
METRICS

- Our calculation of instRec
  - A document is considered as an instance (because no judgments of instance)
  - Concatenate top 10 results of each query

\[ D_F = \bigcup_{i=1}^{n} \{D_i\} \]

- Calculate recall of the concatenated results

\[ \text{instRec} = \frac{|D_F \cap D_R|}{|D_R|} \]
METRICS

• **Instance recall gain (instRecGain)**
  - Evaluates each query’s contribution to the session’s instance recall
  - The instance recall contributed by the kth query’s results $D_k$ is:
    \[
    \text{instRecGain}(D_k) = \text{instRec}\left(\bigcup_{i=1}^{k}\{D_i\}\right) - \text{instRec}\left(\bigcup_{i=1}^{k-1}\{D_i\}\right)
    \]
  - Then, we compute the macro-average instRecGain
METRICS

• nDCG@10 (macro-average), nsDCG@10
  • Do no consider novelty of results

• instRec and instRecGain
  • Do no consider ranking & graded relevance
METRICS

• Macro-average inDCG@10
  • (Jiang, He, Han, Yue, & Ni, CIKM’12)
  • Discount utility of relevant documents in a session based on their rankings in previous results
  • Then, calculate nDCG@10 of each query based on the discounted utility of documents at that moment

• (Shokouhi, White, Bennett, Radlinski, SIGIR’13)
  • “sometimes the repeated results should be promoted, while some other times they should be demoted.”
METRICS

• Average Jaccard Similarity (AvgJaccard)
  • Not a performance measure, but helpful for analyzing novelty of search results.
  • For each unique pair of queries in the session, calculate the top 10 results’ Jaccard similarity, and then calculate the mean value.
ANALYSIS ON AN OLD METHOD

- context-sensitive RF (Shen, Tan & Zhai, SIGIR ’05)
- The “FixInt” method

\[
P(w | \theta_k) = \alpha P(w | q_k) + (1-\alpha) \left[ \beta P(w | H_c) + (1-\beta) P(w | H_q) \right]
\]

\[
P(w | H_c) = \frac{1}{k-1} \sum_{i=1}^{k-1} P(w | C_i)
\]

\[
P(w | H_q) = \frac{1}{k-1} \sum_{i=1}^{k-1} P(w | q_i)
\]
ANALYSIS ON AN OLD METHOD

• context-sensitive RF (Shen, Tan & Zhai, SIGIR ’05)

• Past queries
  • Can lead to serious decline of results’ novelty (Jaccard similarity can increase from 30% to 80%)
  • When we optimize the system by nDCG@10, FixInt gets 10% - 20% improvements on nDCG@10, but also about 20% increase in avgJaccard and 10% decline of instRec.
  • No significant improvements on instRec
    • 0.1079 ➔ 0.1104 (max) in 2011 dataset
    • 0.0881 ➔ 0.0896 (max) in 2012 dataset
ANALYSIS ON AN OLD METHOD

- context-sensitive RF (Shen, Tan & Zhai, SIGIR '05)
- Click-through
  - Slight increase of avgJaccard (less than 10%)
  - Improvements of nDCG@10 comparable to those using past queries (10% - 20%)
  - About 10% Improvements on instRec
    - 0.1079 ➔ 0.1169 (max) in 2011 dataset
    - 0.0881 ➔ 0.1007 (max) in 2012 dataset
  - Still, when we optimize the system by nDCG@10, we cannot get maximum performance on instRec
  - Parameters are not stable in 2011 & 2012 (probably due to the different distribution of session types)
ANALYSIS ON AN OLD METHOD

• context-sensitive RF (Shen, Tan & Zhai, SIGIR ’05)

• Metrics
  • Pearson’s r of metrics’ values on 121 parameter settings

<table>
<thead>
<tr>
<th></th>
<th>TREC 2011</th>
<th>TREC 2012</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>nDCG@10</td>
<td>instRec</td>
<td>nDCG@10</td>
</tr>
<tr>
<td>nDCG@10</td>
<td>1.000</td>
<td>-0.235</td>
<td>1.000</td>
</tr>
<tr>
<td>nsDCG@10</td>
<td>0.985</td>
<td>-0.244</td>
<td>0.994</td>
</tr>
<tr>
<td>inDCG@10_88</td>
<td>-0.013</td>
<td>0.956</td>
<td>0.496</td>
</tr>
<tr>
<td>inDCG@10_85</td>
<td>0.227</td>
<td>0.874</td>
<td>0.703</td>
</tr>
<tr>
<td>inDCG@10_58</td>
<td>0.483</td>
<td>0.719</td>
<td>0.773</td>
</tr>
<tr>
<td>inDCG@10_55</td>
<td>0.686</td>
<td>0.530</td>
<td>0.875</td>
</tr>
<tr>
<td>instRec</td>
<td>-0.235</td>
<td>1.000</td>
<td>0.245</td>
</tr>
<tr>
<td>instRecGain</td>
<td>-0.226</td>
<td>0.979</td>
<td>0.228</td>
</tr>
<tr>
<td>avgJaccard</td>
<td>0.413</td>
<td>-0.957</td>
<td>0.180</td>
</tr>
</tbody>
</table>
TAKE HOME MESSAGES

• Click-through vs. past queries
  • If you are also using past queries as positive relevance feedback information, probably you should re-evaluate your “improvements”.

• Metrics
  • We may need to consider novelty, no matter the task is a single-query task or a whole-session search task (considering people may wrongly use past queries to enhance nDCG@10)

• Optimization
  • Optimizing the parameters for nDCG@10 is risky, usually you cannot balance other evaluation metrics such as instRec
• Thank you!