Enhancing Personalized Search by Mining and Modeling Task Behavior

Ryen White, Wei Chu, Ahmed Hassan, Xiaodong He, Yang Song, and Hongning Wang
Microsoft Research, Microsoft Bing, UIUC
Motivation

• Search behavior part of broader search tasks

• Search engines learn from historic queries

• Rich models of task behavior not built or used

Goal: Personalize via current user & others’ task behavior

• Find historic instances of task (same user or others)

• Use on-task behavior to improve relevance
Background

• User behavior mined from search and browse logs
  • *Interest prediction, satisfaction analysis, query suggestion*
  • “Task” has been proposed as robust alternative to session

• Queries for machine-learned ranking (individual, chains)
• Short- & long-term personalization (query, session, topic)
• Groupization (Teevan et al.) - personalize via related users

• Our method:
  • *Personalize/groupize via on-task behavior of current or other users*
  • *Model tasks using info. available to search engines (queries and clicks)*
Task-Based Groupization

Find other users engaged in similar task

For example, for clicked URLs:

\[ s(t, url) = \sum_{t' \in T} (k(t, t') \cdot c(t', url)) \]

where \( c(t', url) \) is the click frequency for URL for similar tasks

Task-based personalization is also possible – using similar tasks only from current user’s history
Realizing Task-based Groupization

• To realize this vision, we need key functionality:
  • *Identify and model search tasks*
  • *Find related tasks from the current user and/or other users*
  • *Learn from on-task information*

• Discuss each of these in this talk

• There are others:
  • *Filter to users from similar cohorts (in paper, not covered in talk)*
  • *Cohorts include: same location and domain expertise*

  • *E.g., to integrate cohorts into our method ...*
Integrating User Cohorts...

1. Identify user cohort

2. Match only against users in particular cohort

- Task Model Builder
- User Cohort Matching (optional)
- Task Similarity Matching
- Feature Generator (Per URL)
- Result Re-ranker

For example, for clicked URLs:

\[ s(t, url) = \sum_{t' \in T} (k(t, t') \cdot c(t', url)) \]

where \( c(t', url) \) is the click frequency for URL for similar tasks

Set of historic tasks from other users (\( T \))
Realizing Task-based Personalization

1. Identify and model search tasks
2. Find related tasks from the current user and/or other users
3. Learn from on-task information
Step 1: Identifying Tasks in Sessions

• Mine sessions from Bing logs (30 min inactivity timeout)
• Use QTC [Liao et al., 2012] to extract tasks via query relatedness and query clustering:
Task Characteristics

- One week of Bing logs
- 1.4M sessions, 1.9M tasks
  - Avg. 1.36 tasks per session
  - Avg. 2.52 queries per session
  - Avg. 1.86 queries per task
- > 25% of sessions have multiple tasks
- Highlights the importance of considering task
- Explore use of task vs. session in paper
  - Not covered in talk
  - Paper shows that task-based models > session-based models
Step 1: Modeling Tasks

• Represent tasks for comparability

• Create four representations:
  • Queries, Clicked URLs, Clicked Web domains
  • Topical Categories (ODP (dmoz.org) using content-based classifier)

• Tasks are represented as:
  • Sets of queries, clicks, Web domains
  • Probability distributions (over ODP topics), e.g.,

\[ P(\text{topic}) \]

- Query
- Click

- Sports/Football
- Sports/Professional
- Arts
- Health
- Business
- News
- Games
- Recreation
Step 2: Task Relatedness – Query

- Find instances of related tasks
- 2 measures of query relatedness between $t$ and $t'$
  - **Syntactic**
  - *Term overlap between queries in each task (all queries, unique queries)*
  - **Semantic** – machine translation models learned from clicks
  - *Queries may be related semantically even if there is no term overlap*

**Semantic similarity model between query $S$ and $Q**

\[
P(S|Q) = \prod_{i=1}^{I} \sum_{j=1}^{J} P(s_i|q_j)P(q_j|Q)
\]

**Learn translation probabilities $P(s|q)$:**
- Treat <query, title of clicked doc> as translation pairs
- Learn IBM Model 1 with EM

\[
P(S|Q, \theta) = \frac{\varepsilon}{(J + 1)^I} \prod_{i=1}^{I} \sum_{j=1}^{J} P(s_i|q_j)
\]
Step 2: Task Relatedness – Clicks

• Satisfied (SAT) clicks (clicks with dwell ≥ 30 seconds)
• Clicks provide information about intent not in queries

• 3 measures of click relatedness between tasks $t$ and $t'$
  • URL similarity – fraction of unique clicked URLs shared
  • Web domain similarity – fraction of unique clicked domains shared
  • Topical similarity – match on ODP category distributions $C_t$ and $C_{t'}$
    • One asymmetric and one symmetric:

  \[
  KL(t', t) = \sum_{c \in C} \ln \left( \frac{P_t'(c)}{P_t(c)} \right) P_t'(c)
  \]

  \[
  \cos(C_{t'}, C_t) = \frac{C_t \cdot C_{t'}}{\|C_t\| \|C_{t'}\|}
  \]
Step 3: Learn from Related Tasks

• For each query, build representation of current task $t$
  • Previous interactions, including current query (but not its clicks)

• Find related tasks from search histories of other users

• For each URL $u$ in top 10 for current query, compute score $s_k$

$$s_k(t, u) = \sum_{t' \in T} (k(t, t') \cdot w(t', u))$$

$k(t, t')$: relatedness between $t$, related task $t'$, computed in different ways

$w(t', u)$: importance of URL in related task (we use click frequency)

• Generate $s_k$ for a range of different $k(t, t')$
Step 3: Re-Ranking Features

- Computed for current task vs. other tasks

- **ClickedTasksCount**: Total number of tasks for which a particular URL $u$ is clicked
  - *URL popularity ind. of task*

- **QueryTranslation** and **CategorySimilarityKL** are asymmetric $\Rightarrow$ include reverse variants

### Feature name | Definition
--- | ---
FullQueryOverlap | Fraction of all queries in the union of $t$ and $t'$ that the two tasks share
QueryTermOverlap | Fraction of all unique query terms in the union of $t$ and $t'$ that the two tasks share
QueryTranslation | Semantic similarity between the queries in $t$ and the queries in $t'$
ClickedURLOverlap | Fraction of clicked URLs in the union of $t$ and $t'$ that the two tasks share
ClickedDomainOverlap | Fraction of clicked domains in the union of $t$ and $t'$ that the two tasks share
CategorySimilarityKL | Kullback-Liebler divergence between ODP distribution from clicks in $t$ vs the same distribution from $t'$
CategorySimilarityCosine | Cosine similarity between the ODP distribution from result clicks in $t$ versus the same distribution from $t'$. 
Research Questions

• **RQ1:** Does task matching outperform query matching?
• **RQ2:** Does task groupization beat task personalization?

• Others answered in paper, briefly in talk:
  • **RQ3:** Is task segmentation needed or is session okay?
    • *Answer: Performance is better with task segmentation*
  • **RQ4:** Do user cohorts help (e.g., those in a particular location or those with good topic knowledge)?
    • *Answer: Slight gains from cohorts – needs more research*
Models

• Competitive* query-centric baselines
  • Query-based Group (QG; same query, all users)
    • Features from queries in all users’ search histories
  • Query-based Individual (QI; same query, same user)
    • Features from queries in current user’s search history
  • Query-based Group & Individual (QGI)

• Task-centric comparator systems
  • Task-based Group (TG; same task, all users)
    • Features from tasks in all users’ search histories
  • Task-based Individual (TI; same task, same user)
    • Features from tasks in current user’s search history
  • Task-based Group & Individual (TGI)

* Uses Bing, which already leverages user behavior
Judgments and Metrics

• **Relevance:**
  - *Personalized judgments via post-query clicks:*

<table>
<thead>
<tr>
<th>Label=2</th>
<th>Label=1</th>
<th>Label=0</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT click (≥ 30 sec dwell)</td>
<td>Quickback click (&lt; 30 sec dwell)</td>
<td>No click</td>
</tr>
</tbody>
</table>

• **Multi-level helped learn nuanced differences between results**

• **Mean Average Precision (MAP):** many clicks
• **Mean Reciprocal Rank (MRR):** first click on relevant item

\[
\text{AvgPrec} = \frac{\sum_{k=1}^{n} \text{Prec}(k) \cdot \text{Rel}(k)}{\sum_{k=1}^{n} \text{Rel}(k)}
\]

\[
\text{MRR} = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{\text{rank}_i}
\]

• **SAT vs. other (binary) in testing** (conservative - could also use NDCG)

• **Coverage:** fraction of results w/ re-rank@1 and fraction of query instances covered by features
Method

• Gathered four weeks of Bing query-click logs
  • Logs collected from an A/B test with no other personalization
• Week 1: Feature generation
  • Compute $s_k$ for clicked URLs
• Weeks 2-3: Learn re-ranking model (LambdaMART)
• Week 4: Evaluation
  • Re-rank top-10 for each query
  • Compute MAP and MRR for re-ranked lists (and coverage stats)

<table>
<thead>
<tr>
<th>Count</th>
<th>Training</th>
<th>Validation</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAT Clicks</td>
<td>2,086,335</td>
<td>2,062,554</td>
<td>2,082,145</td>
</tr>
<tr>
<td>Quickback Clicks</td>
<td>417,432</td>
<td>408,196</td>
<td>413,496</td>
</tr>
<tr>
<td>Tasks</td>
<td>1,165,083</td>
<td>1,126,452</td>
<td>1,135,320</td>
</tr>
<tr>
<td>Queries per Task</td>
<td>1.678</td>
<td>1.676</td>
<td>1.666</td>
</tr>
</tbody>
</table>
Table 3. MAP/MRR gains on the test data (± SEM). Production ranker is baseline. Query-based baselines highlighted.

<table>
<thead>
<tr>
<th>Model</th>
<th>Δ MAP(10^{-2})</th>
<th>Δ MRR(10^{-2})</th>
<th>Rerank@1</th>
<th>Coverage</th>
<th>Win</th>
<th>Loss</th>
<th>Cost Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>QG</td>
<td>0.0888±0.0023</td>
<td>0.1076±0.0024</td>
<td>0.46%</td>
<td>19.10%</td>
<td>28009</td>
<td>27507</td>
<td>98.21%</td>
</tr>
<tr>
<td>QI</td>
<td>0.1425±0.0028</td>
<td>0.1431±0.0029</td>
<td>0.70%</td>
<td>17.87%</td>
<td>26966</td>
<td>23214</td>
<td>86.09%</td>
</tr>
<tr>
<td>QGI</td>
<td>0.1448±0.0028</td>
<td>0.1455±0.0029</td>
<td>0.71%</td>
<td>19.10%</td>
<td>29259</td>
<td>25097</td>
<td>85.78%</td>
</tr>
<tr>
<td>TG</td>
<td>0.1408±0.0029</td>
<td>0.1440±0.0029</td>
<td>0.88%</td>
<td>67.37%</td>
<td>45866</td>
<td>37668</td>
<td>82.13%</td>
</tr>
<tr>
<td>TI</td>
<td>0.1485±0.0028</td>
<td>0.1490±0.0029</td>
<td>0.71%</td>
<td>19.44%</td>
<td>30932</td>
<td>26586</td>
<td>85.95%</td>
</tr>
<tr>
<td>TGI</td>
<td>0.2292±0.0035</td>
<td>0.2318±0.0036</td>
<td>1.22%</td>
<td>67.37%</td>
<td>32753</td>
<td>22292</td>
<td>68.06%</td>
</tr>
</tbody>
</table>

- Small-ish changes – avg. over all q, many q unchanged
- Some key findings:
  - Both query and task match get gains over baseline
  - Task match better, especially when both feature groups used (TGI)
  - Task match better coverage (> 3x) – re-rank@1 ~2x results as query
Effect of Query Sequence in Task

Figure 2. Segment analysis on MAP for queries issued at different points in the task (± SEM).

Some key findings:

- All models clearly outperform QG throughout the task
- TI and TG similar, apart from the first query (effect of re-finding?)
RQ2: Group vs. Individual

Table 4. Comparison on the test data. ΔMAP and ΔMRR denote the MAP and MRR difference from the baseline model (TG) respectively (± SEM).

<table>
<thead>
<tr>
<th>Models</th>
<th>ΔMAP(10^{-2})</th>
<th>ΔMRR(10^{-2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>TI vs. TG</td>
<td>0.0077±0.0033</td>
<td>0.0050±0.0025</td>
</tr>
<tr>
<td>TGI vs. TG</td>
<td>0.0884±0.0026</td>
<td>0.0878±0.0031</td>
</tr>
</tbody>
</table>

• Some key findings:
  • Group and Individual statistically indistinguishable
  • Group has > 3x query coverage
  • Combining group and individual gets relevance gains (vs. TG)
Summary

• Improved search relevance by mining task behavior
• Used on-task behavior from current searcher & others
• Task match > query match (relevance & coverage)
• Task groupization ≈ task personalization (3x coverage)
• Also (in paper), task > session, user cohorts useful

• Future work: explore cohorts and cohort combinations, richer task models – including behavior beyond engine, beyond the Web...