Ahmed Hassan, Ryen White, Yi-Min Wang

Toward Self-Correcting Search Engines: Using Underperforming Queries to Improve Search
Search Satisfaction Prediction

• Behavioral Modeling
  - Hassan et al, WSDM 2010
  - Ageev et al., SIGIR 2011
  - Hassan, SIGIR 2010

• Engine Switching
  - Heath and White, WWW 2008
  - White and Dumais, CIKM 2009
  - Guo et al., SIGIR 2011
Search Satisfaction Prediction

- Applications:
  - Metrics
  - Finding Dissatisfied searches (DSATs)
Bringing Order to DSATs

• A Bottom-Up approach to mining DSAT Segments

• A **DSAT Segment** is a coherent set of DSAT queries that can be characterized by a small number of attributes
DSAT Segments

Every bubble is a DSAT Segment
DSAT Segments

Bubble size: Number of DSATs in the segment
DSAT Correlation: A measure of segment dsatness.
Agenda

• A Process
  – Generate DSATs and DSAT Segment

• A Metric
  – Measuring performance across segments

• A Method
  – for tackling consistent search quality problems
System Overview

The Process

General Ranker (GR) → Search Logs → DSAT Finder → Individual DSATs → DSAT Segment Mining

Hybrid Ranker → Combine SSR and GR

Segment Specific Ranker (SSR) → Build a Segment Specific Ranker (SSR)

Additional Investment

START

Microsoft Research
System Overview

The Metric

General Ranker (GR)

Search Logs

DSAT Finder

Individual DSATs

DSAT Segment Mining

Hybrid Ranker

Combine SSR and GR

Segment Specific Ranker (SSR)

Build a Segment Specific Ranker (SSR)

Additional Investment

DSAT Segments
System Overview

Improving Search Quality

- General Ranker (GR)
- Search Logs
- DSAT Finder
- Individual DSATs
- DSAT Segment Mining
- Hybrid Ranker
- Combine SSR and GR
- Segment Specific Ranker (SSR)
- Build a Segment Specific Ranker (SSR)
- DSAT Segments

Additional Investment
Mining DSAT Instances

- Engine Switching
  - Voluntary transition from one search engine to another search engine
  - A *search engine switching event* is a pair of consecutive queries that are issued on different search engines in a session

[White and Dumais, CIKM 2009]
Reasons for Engine Switching

- A retrospective questionnaire of 488 users

[White and Dumais, CIKM’09]
SwitchWatch

• Browser add-on
• Pop-up dialog when switch
• Record URLs, timestamps, tab focus, etc.
• Switch Reasons
  – Dissatisfaction
  – Verification/Coverage
  – Unintentional
  – Better for this type
  – Preference
  – Other
• Ignore Button

[Guo et al., SIGIR’11]
DSAT Switches Classifier

- A binary classifier is trained to identify DSAT switches from other switches

- $F_{0.5} = 85$ (All Features)
- $F_{0.5} = 81$ (Query+post-switch)

### Query Features

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query length in terms of number of characters</td>
</tr>
<tr>
<td>Query length in terms of number of words</td>
</tr>
<tr>
<td>Time in seconds between pre-switch and post switch queries</td>
</tr>
</tbody>
</table>

### Pre/Post-switch features

<table>
<thead>
<tr>
<th>Feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num. of queries in session</td>
</tr>
<tr>
<td>Num. of unique queries in session</td>
</tr>
<tr>
<td>Num. of query reformulations</td>
</tr>
<tr>
<td>Num. of clicks</td>
</tr>
<tr>
<td>Num. of clicks with dwell time &gt; 30s</td>
</tr>
<tr>
<td>Num. of clicks with dwell time &lt; 15s</td>
</tr>
<tr>
<td>Num. of clicks on URLs containing a query term in their title</td>
</tr>
<tr>
<td>Length of trail from search engine result page, defined as the number of clicks from SERP</td>
</tr>
</tbody>
</table>

**Number of transitions between every action pair** $a_i \rightarrow a_j$ **for every** $a_i \in A$ **where** $A = \{Query, SERP Click, Ad Click, Answer, etc.\}$

**Avg. dwell time for every action pair** $a_i \rightarrow a_j$

Features from [Guo et al., SIGIR’11], [Hassan, SIGIR 2012]
• Every dissatisfaction instance is described with a vector of binary attributes.

• Each attribute describes a specific characteristic of the dissatisfaction instance.

<table>
<thead>
<tr>
<th>Attribute_1</th>
<th>Attribute_2</th>
<th>......</th>
<th>Attribute_n</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
<td>......</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>......</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>......</td>
<td>True</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>False</td>
<td>......</td>
<td>......</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>......</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>......</td>
<td>True</td>
</tr>
</tbody>
</table>
Attributes

- Query Attributes
  - Query topic, query length, etc.
- SERP Attributes
  - Answers displayed, etc.
- Impression Attributes
  - Market, time, etc.
- User Attributes
  - Time using engine, search experience, etc.
• Find frequent attribute associations in dissatisfied queries
  – FP-Growth algorithm

<table>
<thead>
<tr>
<th>a</th>
<th>b</th>
<th>c</th>
<th>d</th>
<th>e</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>True</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>True</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>True</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>True</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>False</td>
<td>False</td>
<td>False</td>
</tr>
</tbody>
</table>
Frequent ≠ Useful

- Frequent attribute sets discovered from a set of dissatisfaction instances do not necessarily define a DSAT segment.

\{English Query, US Market\}
\{English Query, Short\}
Segments

- Add $SAT$ instances to the dataset
  - randomly sampling queries with Last click with dwell time $> 30$ seconds
- Measure $DSAT$ Correlation

<table>
<thead>
<tr>
<th>Attribute_1</th>
<th>Attribute_2</th>
<th>......</th>
<th>Attribute_n</th>
<th>(D)SAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>False</td>
<td>......</td>
<td>True</td>
<td>DSAT</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>......</td>
<td>False</td>
<td>DSAT</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>......</td>
<td>True</td>
<td>DSAT</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>False</td>
<td>True</td>
<td>......</td>
<td>True</td>
<td>SAT</td>
</tr>
<tr>
<td>True</td>
<td>True</td>
<td>......</td>
<td>False</td>
<td>SAT</td>
</tr>
<tr>
<td>True</td>
<td>False</td>
<td>......</td>
<td>True</td>
<td>SAT</td>
</tr>
<tr>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
</tbody>
</table>
DSAT Correlation

- For any attribute set \( A = \{a_1, \ldots, a_n\} \):

\[
DSAT \text{ Correlation}(A) = \frac{P(A \& DSAT)}{P(A)P(DSAT)}
\]

Negative Correlation with Dissatisfaction

Positive Correlation with Dissatisfaction
Validating DSAT Segments

- Compare DSAT Correlation to:
  - Avg. CTR
  - AVG. NDCG@1
  - AVG. NDCG@3
CTR vs. DSAT Correlation
NDCG@1 vs. DSAT Correlation
NDCG@3 vs. DSAT Correlation
System Overview

Improving Search Quality

- General Ranker (GR)
- Search Logs
- DSAT Finder
- Individual DSATs
- DSAT Segment Mining
- Hybrid Ranker
- Combine SSR and GR
- Segment Specific Ranker (SSR)
- Build a Segment Specific Ranker (SSR)
- Additional Investment

Microsoft Research
• It is hard to explain why a particular search result ranks highly or lowly

• It is hard to tweak the system to handle DSATs in isolated context
• General Ranker
  – Trained on a general queryset
• Segment Specific Ranker
  – Trained on general and segment Data
Results

- Both General and Segment-Specific Rankers tested using DSAT segment data

- NDCG Gain:

<table>
<thead>
<tr>
<th>NDCG@1</th>
<th>NDCG@2</th>
<th>NDCG@3</th>
<th>NDCG@4</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.52</td>
<td>0.58</td>
<td>0.86</td>
<td>0.94</td>
<td>0.77</td>
</tr>
</tbody>
</table>
Baseline

- Will we get a similar gain if we train a ranker for a group of random poor performing queries?
  - Train a ranker on a set of queries with low CTR

<table>
<thead>
<tr>
<th></th>
<th>NDCG@1</th>
<th>NDCG@2</th>
<th>NDCG@3</th>
<th>NDCG@4</th>
<th>NDCG@5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coherent DSAT Segment</td>
<td>1.52</td>
<td>0.58</td>
<td>0.86</td>
<td>0.94</td>
<td>0.77</td>
</tr>
<tr>
<td>Underperforming Queries</td>
<td>0.51</td>
<td>0.39</td>
<td>0.12</td>
<td>0.05</td>
<td>0.03</td>
</tr>
</tbody>
</table>

- Gain attributed to segment coherence
Conclusion

A Process

A Metric

Improving Search Quality
Next Steps

- More Attributes
- Segment Hierarchies
- Multiple Segments
Thanks

Ahmed Hassan
hassanam@microsoft.com