Query Session Detection as a Cascade

Matthias Hagen    Benno Stein    Tino Rüb

Bauhaus-Universität Weimar
matthias.hagen@uni-weimar.de

SIR 2011
Dublin, Ireland
April 18, 2011
It’s quiz time!
It’s quiz time!

What is the user searching?

paris hilton
Without context . . .

paris hilton

source: [http://upload.wikimedia.org/wikipedia/commons/2/26/Paris_Hilton_3_Crop.jpg]
What if you knew the previous queries?

paris hotels
paris marriott
paris hyatt
paris hilton
What if you knew the previous queries?

paris hotels
paris marriott
paris hyatt
paris hilton

[Map of Paris with Hilton hotel location]

Sources:
- [http://maps.google.com/]
- [http://upload.wikimedia.org/wikipedia/en/e/eb/Hiltonbrandlogo.jpg]
Query sessions: same information need

The benefits

- Improved understanding of user intent
- Improved retrieval performance via session knowledge
Query sessions: same information need

The benefits
- Improved understanding of user intent
- Improved retrieval performance via session knowledge

The “minor” issue
Users do not announce when querying for a new information need.
## A typical query log

<table>
<thead>
<tr>
<th>User</th>
<th>Query</th>
<th>Click domain + Click rank</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>773</td>
<td>istanbul</td>
<td>en.wikipedia.org</td>
<td>2011-04-16 20:34:17</td>
</tr>
<tr>
<td>773</td>
<td>istanbul archeology</td>
<td><a href="http://www.kulturturizm.tr">www.kulturturizm.tr</a></td>
<td>2011-04-17 12:02:54</td>
</tr>
<tr>
<td>773</td>
<td>constantinople</td>
<td><a href="http://www.roman-empire.net">www.roman-empire.net</a></td>
<td>2011-04-17 19:00:40</td>
</tr>
<tr>
<td>773</td>
<td>constantinople</td>
<td><a href="http://www.roman-empire.net">www.roman-empire.net</a></td>
<td>2011-04-17 19:01:02</td>
</tr>
<tr>
<td>773</td>
<td>hurling</td>
<td>en.wikipedia.org</td>
<td>2011-04-17 19:03:05</td>
</tr>
<tr>
<td>773</td>
<td>hurling</td>
<td>en.wikipedia.org</td>
<td>2011-04-17 19:03:05</td>
</tr>
<tr>
<td>773</td>
<td>liam mccarthy cup</td>
<td>starbets.ie</td>
<td>2011-04-18 12:42:48</td>
</tr>
</tbody>
</table>
How to determine the break points?

<table>
<thead>
<tr>
<th>User</th>
<th>Query</th>
<th>Click domain + Click rank</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>773</td>
<td>istanbul</td>
<td>en.wikipedia.org</td>
<td>2011-04-16 20:34:17</td>
</tr>
<tr>
<td>773</td>
<td>istanbul archeology</td>
<td><a href="http://www.kulturturizm.tr">www.kulturturizm.tr</a></td>
<td>2011-04-17 12:03:15</td>
</tr>
<tr>
<td>773</td>
<td>constantinople</td>
<td></td>
<td>2011-04-17 19:01:02</td>
</tr>
<tr>
<td>773</td>
<td>constantinople</td>
<td><a href="http://www.roman-empire.net">www.roman-empire.net</a></td>
<td>2011-04-17 19:01:02</td>
</tr>
<tr>
<td>773</td>
<td>hurling</td>
<td></td>
<td>2011-04-17 19:03:01</td>
</tr>
<tr>
<td>773</td>
<td>hurling</td>
<td>en.wikipedia.org</td>
<td>2011-04-17 19:03:05</td>
</tr>
<tr>
<td>773</td>
<td>liam mccarthy cup</td>
<td></td>
<td>2011-04-17 23:33:04</td>
</tr>
<tr>
<td>773</td>
<td>liam mccarthy cup</td>
<td>starbets.ie</td>
<td>2011-04-18 12:42:48</td>
</tr>
</tbody>
</table>
The key is ... Automatic query session detection
### Automatic query session detection

**Usual “technique”**

Check for consecutive queries whether same/new information need.

<table>
<thead>
<tr>
<th>ID</th>
<th>Query</th>
<th>Date</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>773</td>
<td>istanbul</td>
<td>2011-04-16 20:34:17</td>
<td>✓ same</td>
</tr>
<tr>
<td>773</td>
<td>istanbul archeology</td>
<td>2011-04-17 18:24:07</td>
<td>✓ same</td>
</tr>
<tr>
<td>773</td>
<td>constantinople</td>
<td>2011-04-17 19:01:02</td>
<td></td>
</tr>
<tr>
<td>773</td>
<td>hurling</td>
<td>2011-04-17 19:03:05</td>
<td>⇣ new</td>
</tr>
</tbody>
</table>
## Typical features

<table>
<thead>
<tr>
<th>Temporal thresholds</th>
<th>5 minutes</th>
<th>[Silverstein et al., 1999]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>10–15 minutes</td>
<td>[He and Göker, 2000]</td>
</tr>
<tr>
<td></td>
<td>30 minutes</td>
<td>[Downey et al., 2007]</td>
</tr>
<tr>
<td></td>
<td>user specific</td>
<td>[Murray et al., 2006]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lexical similarity</th>
<th>$n$-gram overlap</th>
<th>[Zhang and Moffat, 2006]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Levenshtein distance</td>
<td>[Jones and Klinkner, 2008]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Semantic similarity</th>
<th>Search results</th>
<th>[Radlinski and Joachims, 2005]</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ESA</td>
<td>[Lucchese et al., 2011]</td>
</tr>
</tbody>
</table>
Previous methods

Observations

- Temporal thresholds: fast but bad accuracy
- Feature combinations: more accurate
- One of the best: Geometric method (time + lexical) [Gayo-Avello, 2009]
Previous methods

Observations
- Temporal thresholds: fast but bad accuracy
- Feature combinations: more accurate
- One of the best: Geometric method (time + lexical) \[Gayo-Avello, 2009\]

Shortcomings
- All features evaluated simultaneously $\rightarrow$ runtime
- Geometric method ignores semantics $\rightarrow$ accuracy

Examples
- Subset test suffices
  - hurling ✓ same
  - hurling gaa ✓ same

- Geometric method fails
  - hurling ✓ same
  - mccarthy cup ✓ same
We address the shortcomings in a cascade ...
... well ... a small 4-step cascade
Cascading Method

The Framework

... well ... a small 4-step cascade

Step 1: Subset tests

↘

Step 2: Geometric method

↘

Step 3: ESA similarity

↙

Step 4: Search results

Basic Idea

Increased feature cost (runtime) from step to step. Expensive features only if previous steps “unreliable.”
Simple string comparison

Criterion

Consecutive queries $q$ and $q'$ in same session if $q$ sub- or superset of $q'$. Else: Goto Step 2.

Remarks: Repetition, specialization, or generalization. Time gap = continuing a pending session.

Example

<table>
<thead>
<tr>
<th>Repetition</th>
<th>Specialization</th>
<th>Generalization</th>
</tr>
</thead>
<tbody>
<tr>
<td>hurling</td>
<td>hurling</td>
<td>hurling gaa</td>
</tr>
<tr>
<td>✓ same</td>
<td>✓ same</td>
<td>✓ same</td>
</tr>
<tr>
<td>hurling</td>
<td>hurling gaa</td>
<td>hurling</td>
</tr>
</tbody>
</table>
For consecutive queries $q$ and $q'$

$$f_{\text{temp}} = \max\{0, 1 - \frac{t}{24h}\} \quad t \text{ is time between } q \text{ and } q'$$

$$f_{\text{lex}} = \text{cosine similarity of 3- to 5-grams of } q' \text{ and } s \quad s \text{ is session of } q$$
Combination of temporal and lexical features

For consecutive queries \( q \) and \( q' \)

\[
f_{\text{temp}} = \max(0, 1 - \frac{t}{24h}) \quad t \text{ is time between } q \text{ and } q'
\]

\[
f_{\text{lex}} = \text{cosine similarity of 3- to 5-grams of } q' \text{ and } s
\]

\( s \) is session of \( q \)

Criterion (original)

Consecutive queries \( q \) and \( q' \) in same session if

\[
\sqrt{f_{\text{temp}}^2 + f_{\text{lex}}^2} \geq 1.
\]
Performs well on standard test corpus ...
... but has some problems “on the edge”

Major problems

Similar queries, time gap (upper left) → Merely a matter of opinion

Diff. queries, same semantics (lower right) → Incorporate semantics
Cascading Method

Step 2: Geometric method

... but has some problems “on the edge”

Major problems

Similar queries, time gap (upper left) → Merely a matter of opinion

Diff. queries, same semantics (lower right) → Incorporate semantics

Criterion (adapted)

Original geometric method if \( f_{\text{temp}} < 0.8 \) or \( f_{\text{lex}} > 0.4 \).
Else: Goto Step 3.
How ESA works

[Gabrilovich and Markovitch, 2007]

Preprocessing

*tf* · *idf*-weighted inverted index of Wikipedia articles → term-document matrix *M*

For consecutive queries *q* and *q'*

\[ f_{esa} = \text{cosine similarity of } M^T \cdot q' \text{ and } M^T \cdot s \]

* s is session of *q*

Criterion

Consecutive queries *q* and *q'* in same session if \( f_{esa} \geq 0.35 \).
Else: Goto Step 4.
Even more “semantics”

**Idea**
Enrich the short query strings with the results of some web search engine.

**Criterion**
Consecutive queries \( q \) and \( q' \) in same session iff

they share at least one of the top 10 search results.
Even more “semantics”

**Idea**

Enrich the short query strings with the results of some web search engine.

Criterion

Consecutive queries \( q \) and \( q' \) in same session iff 
they share at least one of the top 10 search results.

Remark

If \( q \) and \( q' \) share no top 10 result, decision should be “not sure.”
That’s the complete cascade

Step 1: Subset tests

↘

Step 2: Geometric method

↘

Step 3: ESA similarity

↙

Step 4: Search results

source: [http://www.solarshop.com/solarpix/Solar Cascade 4 Tier GreenL.jpg]
That’s the complete cascade

Step 1: Subset tests

Step 2: Geometric method

Step 3: ESA similarity

Step 4: Search results

What about accuracy and performance?
### Accuracy and runtime

#### Accuracy on Gayo-Avello’s corpus (11,000 queries, 2.7 per session)

<table>
<thead>
<tr>
<th>Method</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure ($\beta = 1.5$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometric</td>
<td>0.8673</td>
<td>0.9431</td>
<td>0.9184</td>
</tr>
<tr>
<td>Cascading</td>
<td>0.8618</td>
<td>0.9676</td>
<td>0.9328</td>
</tr>
</tbody>
</table>

#### Performance per step on Gayo-Avello’s corpus

<table>
<thead>
<tr>
<th>Step</th>
<th>affected</th>
<th>F-Measure</th>
<th>time</th>
<th>factor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 1</td>
<td>40.49%</td>
<td>0.8303</td>
<td>0.08 ms</td>
<td>1.0</td>
</tr>
<tr>
<td>Step 2</td>
<td>35.15%</td>
<td>0.9292</td>
<td>0.20 ms</td>
<td>2.5</td>
</tr>
<tr>
<td>Step 3</td>
<td>2.05%</td>
<td>0.9316</td>
<td>0.27 ms</td>
<td>3.4</td>
</tr>
<tr>
<td>Step 4</td>
<td>0.85%</td>
<td>0.9328</td>
<td>9.85 ms</td>
<td>123.1</td>
</tr>
</tbody>
</table>
Goal: high quality session test data

Our own use case
Sample sessions from the AOL log as test data.
AOL log (cleaned): 35.4 million interactions from 470,000 users.

Some figures
Step 4 involved on 22.5% → 8 million web queries
→ 300 ms per search → 1 month
Goal: high quality session test data

Our own use case
Sample sessions from the AOL log as test data.
AOL log (cleaned): 35.4 million interactions from 470,000 users.

Some figures
Step 4 involved on 22.5% → 8 million web queries
→ 300 ms per search → 1 month

Way out
• Drop Step 4 and the sessions on which it would have been invoked

Remaining sessions: F-Measure = 0.9755
Cleaned AOL log: 27 minutes
Almost the end: The take-away messages!
What we have done

Results
- Cascading method
- Cheap features first
- Beats geometric
- 3 step version: simple, fast, high quality sessions

Future Work
- Postprocessing for multi-tasking
- Postprocessing for goals/missions
What we have (not) done

**Results**
- Cascading method
- Cheap features first
- Beats geometric
- 3 step version: simple, fast, high quality sessions

**Future Work**
- Postprocessing for multi-tasking
- Postprocessing for goals/missions
Conclusion

What we have (not) done

Results
- Cascading method
- Cheap features first
- Beats geometric
- 3 step version: simple, fast, high quality sessions

Future Work
- Postprocessing for multi-tasking
- Postprocessing for goals/missions

Thank you 😊