Relevance Debugging and Explaining at LinkedIn

May 20th, 2019

Daniel Qiu
Software Engineer

Yucheng Qian
Sr. Software Engineer
Why relevance debugging and explaining

**Modeling**  
Improve the machine learning model

**Value**  
Bring value to our members by providing relevant experience

**Trust**  
Build trust with our members
What Could Go Wrong

- **Bad feature quality**
- **Feature online offline inconsistency**
- **Model A/B test issues**

- **Service is down**
- **Stale data**
Challenges

- Complex Infrastructure
- Hard to Reproduce
- Time Consuming
Call Graph

Understand the call graph and see what goes wrong in the service.

- service returned status code 500
  - service: mail-carriers-broker
  - requestId: -1974802163
  - server: ft1-app[004] qty.linkedin.com
  - instance: i000
  - version: 0.3.223
Targeted Query

Debug a specific service, data center, cluster, or host
Timing Data

The statistics about the time spent on each phase of a search.

### Timing Data

<table>
<thead>
<tr>
<th>Phase</th>
<th>Start Time</th>
<th>End Time</th>
<th>Total Time</th>
<th>Resent?</th>
<th>Partitions</th>
<th>Min</th>
<th>Max</th>
<th>p50</th>
<th>p90</th>
</tr>
</thead>
<tbody>
<tr>
<td>search_phase_one</td>
<td>7</td>
<td>266</td>
<td>259</td>
<td>false</td>
<td>16</td>
<td>205</td>
<td>253</td>
<td>223.0</td>
<td>245.5</td>
</tr>
<tr>
<td>facet_discovery</td>
<td>13</td>
<td>240</td>
<td>227</td>
<td>true</td>
<td>16</td>
<td>135</td>
<td>232</td>
<td>164.0</td>
<td>186.0</td>
</tr>
<tr>
<td>facet_count</td>
<td>262</td>
<td>1041</td>
<td>779</td>
<td>true</td>
<td>16</td>
<td>523</td>
<td>785</td>
<td>617.0</td>
<td>700.0</td>
</tr>
<tr>
<td>search_phase_two</td>
<td>266</td>
<td>274</td>
<td>8</td>
<td>false</td>
<td>15</td>
<td>5</td>
<td>9</td>
<td>8.0</td>
<td>9.0</td>
</tr>
</tbody>
</table>
Features

Inspect the features for the model

<table>
<thead>
<tr>
<th>Group</th>
<th>Feature</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPR</td>
<td>activity_recent_click</td>
<td>968</td>
</tr>
<tr>
<td>SPR</td>
<td>some_text</td>
<td>1</td>
</tr>
<tr>
<td>SPR</td>
<td>another_text</td>
<td>6.8762646</td>
</tr>
<tr>
<td>SPR</td>
<td>third_text</td>
<td>null</td>
</tr>
<tr>
<td>SPR</td>
<td>fourth_text</td>
<td>null</td>
</tr>
<tr>
<td>SPR</td>
<td>binary_activity_recent_click</td>
<td>1</td>
</tr>
<tr>
<td>SPR</td>
<td>more_text</td>
<td>null</td>
</tr>
<tr>
<td>SPR</td>
<td>log_activity_recent_click</td>
<td>6.8762646</td>
</tr>
<tr>
<td>SPR</td>
<td>yet_text</td>
<td>0</td>
</tr>
<tr>
<td>SPR</td>
<td>this_text</td>
<td>0</td>
</tr>
</tbody>
</table>
How to Conduct Mobile UX Research (and What to Do with It)
The number of people browsing and shopping on mobile devices continues to grow. During the past 10 years, smartphone and......

<table>
<thead>
<tr>
<th>Position</th>
<th>#1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference</td>
<td>urn:li:article:</td>
</tr>
<tr>
<td>SPR Score</td>
<td>0.0035542063</td>
</tr>
<tr>
<td>Relevance Model</td>
<td></td>
</tr>
<tr>
<td>Source Type</td>
<td>ORGANIC</td>
</tr>
<tr>
<td>FPR Model</td>
<td></td>
</tr>
</tbody>
</table>
Entities Returned

See how entities are reranked and filtered between services.
Advanced Use Cases

Perturbation

Comparison

Replay
System Perturbation
Turning knobs and see what happens.

1. Inject
   Injected as part of the request

2. Relay
   Passed to downstream service

3. Overwrite
   Overwrite the system behavior
Perturbation Example

- Override A/B test settings
- Model selection
- Feature override
Side-by-Side Comparison

Compare Model
Compare results of 2 different queries/models

Compare Items
Compare features and scores of 2 different items, from the same query or different queries
Compare 2 queries
### Compare same item in 2 queries

<table>
<thead>
<tr>
<th>Feature</th>
<th>Item 1</th>
<th>Item 2</th>
<th>% Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>responsePenalty /</td>
<td>4.0601455e-7</td>
<td>0.009018197</td>
<td>2221061.19</td>
</tr>
<tr>
<td>response</td>
<td>5.2125584e-9</td>
<td>0.000011580406</td>
<td>222063.57</td>
</tr>
<tr>
<td>score_response_viral</td>
<td>5.2125584e-9</td>
<td>0.000011580406</td>
<td>222063.57</td>
</tr>
<tr>
<td>diffHoursSinceLvFiveAndAgeInHour /</td>
<td>-3.0348454</td>
<td>-50.475624</td>
<td>1563.2</td>
</tr>
</tbody>
</table>
Replay

Scoring and ranking could be time sensitive

• Relevance services emit features and scores using Kafka.
• Consume Kafka event using Samza and store data in the database
• Query database for historical data when debugging
Feed Replay UI
Protecting members’ privacy

- GDPR compliant
- Controlled and audited access
- Limited data retention time
- Replay for internal employees only
Customer AI teams

• Search
• Feed
• Comments
• People you may know
• Job you may be interested in
• Notification
Q&A