Query Auto-Completion for Rare Prefixes

Bhaskar Mitra
Microsoft
bmitra@microsoft.com

Nick Craswell
Microsoft
nickcr@microsoft.com

Step 1
Mine popular suffixes from historical query logs.

<table>
<thead>
<tr>
<th>Top suffixes</th>
<th>Top 2-word suffixes</th>
<th>Top 3-word suffixes</th>
</tr>
</thead>
<tbody>
<tr>
<td>com</td>
<td>for sale</td>
<td>federal credit union</td>
</tr>
<tr>
<td>org</td>
<td>yahoo com</td>
<td>new york city</td>
</tr>
<tr>
<td>net</td>
<td>myspace com</td>
<td>in new york</td>
</tr>
<tr>
<td>gov</td>
<td>google com</td>
<td>or no deal</td>
</tr>
<tr>
<td>pictures</td>
<td>new york</td>
<td>dicoey channel com</td>
</tr>
<tr>
<td>lyrics</td>
<td>real estate</td>
<td>my space com</td>
</tr>
<tr>
<td>edis</td>
<td>of america</td>
<td>in new jersey</td>
</tr>
<tr>
<td>sale</td>
<td>high school</td>
<td>homes for sale</td>
</tr>
<tr>
<td>games</td>
<td>new jersey</td>
<td>department of corrections</td>
</tr>
<tr>
<td>florida</td>
<td>space com</td>
<td>chamber of commerce</td>
</tr>
<tr>
<td>us</td>
<td>s com</td>
<td>in las vegas</td>
</tr>
</tbody>
</table>

Step 2
Extract the end-term from an input prefix.

- cheapest flight fro
  End-term: “fro”
- cheapest flight from
  End-term: “from”
- cheapest flight from i
  End-term: “from”
- cheapest flight from ri
  End-term: “n”

Step 3
Identify suffixes that begin with the end-term and append them to the prefix to generate synthetic suggestions. Then rank the synthetic suggestions alongside the full-query candidates using a LambdaMART model with n-gram and CLSM based features.

- cheapest flights from seattle to
- cheapest flights from seattle to dc
- cheapest flights from seattle to washington dc
- cheapest flights from seattle to bermuda
- cheapest flights from seattle to bahamas
- cheapest flights from seattle to aruba
- cheapest flights from seattle to punta cana

Convolutional Latent Semantic Model
We train a novel CLSM based language model on a query prefix-suffix pairs corpus. The training pairs are sampled from the search query logs by splitting each query at every possible word boundary.

The trained CLSM model is then used for generating features for the LambdaMART based ranking model.

Summary
Generating synthetic candidates for query auto-completion by appending popular query suffixes to the input prefix and ranking them alongside traditional full-query candidates using n-gram statistics and a novel deep neural network model. Unlike next word prediction, this approach always suggests complete queries.

Results
Overall, the LambdaMART model with both the n-gram and the CLSM features achieves the best MRR. However, compared to the CLSM feature the n-gram features show higher improvements.

Most of the improvements in MRR are from the rare and the unseen prefixes as expected.