Effective Structured Query Formulation for Session Search

Dongyi Guan, Grace Hui Yang, Nazli Goharian

Speaker: Grace Hui Yang

Department of Computer Science
Georgetown University

Nov 9, 2012@TREC

Introduction

• A session contains
  • Interactions
    - Previous Queries \( q_1, q_2, \ldots, q_{n-1} \)
    - Previous Search Results
    - Click Information
  • Current Query \( q_n \)

• A retrieval task
  • So, we basically use Lemur – a strong baseline

• The problem becomes
  • how to make good use of Lemur
  • ... and how to improve over Lemur
Outline

• Our Approach
  – Structured Query Formulation
  – Query Expansion
  – Duplicated Queries
  – Document Re-ranking by Dwell Time

• Experiments

Structured Query Formulation

• We emphasize on formulating effective Lemur queries
  – then submit the queries to Lemur

• An observation:
  – In a query, several words often bundle together as a phrase to express a coherent meaning; we call them nuggets
  – Nuggets are substrings in $q_n$, similar to phrases but not necessarily as semantically coherent as phrases
  – Using nuggets to form structured query could be more effective than using plain free text query
How to Identify **Nuggets**

- More Observations:
  - a valid nugget (within a query) appears frequently in the top returned snippets for the query
  - Even if the words in a nugget do not appear continuously in the snippets, they appear close.

  ...preservation uk spinal cord injury care in egypt cky cord
  noats guitar viewsonic power cord malfunction spinal cord
  stimulator...injury spinal cord dell...extension cord
  nylon...cords paralysis vocal vegas...

  A sample snippet for TREC 2012 session 53 query servering spinal cord paralysis, where “spinal cord” are grouped as a nugget #1(spinal cord)

---

How to Identify **Nuggets**

1. Send $q_n$ to Lemur, get initial retrieval results
2. look for possible nuggets in the top $k$ snippets
   - High frequency adjacent words
   - Other words which frequently co-occur within a certain proximity
### Strict Method

\[ q = w_1 w_2 \cdots w_i \]
\[
\text{count}(w_i w_{i+1}; \text{Snippet}) \geq \theta \implies w_i w_{i+1} \text{ are connected}
\]

\[ \text{nugget} = \#I(w_i w_{i+1} \cdots w_j) \quad w_i w_{i+1} \cdots w_j \text{ are connected} \]

### Example

**Query**

servering spinal cord paralysis

**Snippet**

...preservation uk spinal cord injury care in egypt cky cord noats guitar viewsonic power cord malfunction spinal cord stimulator...injury spinal cord dell...extension cord nylon...cord paralysis vocal vegas...

**Structured Query**

servering #1(spinal cord) paralysis

### Relaxed Method

Define centroid of a word \( w_i \) is

\[
\bar{x}(w_i) = \frac{1}{k} \sum_{j=1}^{k} x_j(w_i; S_i)
\]

Predict the nugget window size \( \text{nugget} = \begin{cases} 
\#I(w_i w_{i+1}) & |\bar{x}(w_i) - \bar{x}(w_{i+1})| \leq 5 \\
\#2(w_i w_{i+1}) & 5 < |\bar{x}(w_i) - \bar{x}(w_{i+1})| \leq 10 \\
\phi & |\bar{x}(w_i) - \bar{x}(w_{i+1})| > 10
\end{cases}
\]

### Example

**Query**

marsupial cartoon character

**Structured Query**

marsupial #2(cartoon character)

**Snippet**

...about a cartoon character. For the carnivorous marsupial, see Tasmanian...animated cartoon character in the...series of cartoons...the character after...between the marsupial...encyclopedia cartoon character} Jump...propelled the character to new... animated cartoon character...Tunes series of cartoons. The character appeared in...

\[
\bar{x}(\text{marsupial}) = 35\\
\bar{x}(\text{cartoon}) = 54\\
\bar{x}(\text{character}) = 60
\]

\[ \begin{cases} \quad \text{marsupial} \\ #2(\text{cartoon character}) \end{cases} \]

**RL1 Query**
Query Expansion with Previous Queries

1. Extract nuggets and words from every query \( q_1, q_2, \ldots, q_n \) in a session
2. Combine them and weigh them by per-query weight \( \lambda_k \)

\#weight:
\[
\begin{align*}
\lambda_1 &= \text{combine}(\text{nugget}_{11} \text{nugget}_{12} \cdots \text{nugget}_{1m} \ w_{11} w_{12} \cdots w_{1r}) \\
\lambda_2 &= \text{combine}(\text{nugget}_{21} \text{nugget}_{22} \cdots \text{nugget}_{2m} \ w_{21} w_{22} \cdots w_{2r}) \\
&\vdots \\
\lambda_n &= \text{combine}(\text{nugget}_{n1} \text{nugget}_{n2} \cdots \text{nugget}_{nm} \ w_{n1} w_{n2} \cdots w_{nr}) 
\end{align*}
\]

Query Expansion with previous queries

Weighting Schemes

• Uniform
  All queries are assigned the same weight. \( \lambda_k = 1 \)

• Previous vs. current
  All previous queries share the same weight while the current query uses a complementary and higher weight
  \[
  \lambda_k = \begin{cases} 
  \lambda_p & k = 1,2,\ldots,n-1 \\
  1 - \lambda_p & k = n 
\end{cases} 
\]
  \( \lambda_p = 0.4 \)

• Distance-based
  The weights are distributed based on how far a query’s position in the session is from the current query
  \[
  \lambda_k = \begin{cases} 
  \frac{\lambda_p}{n-k} & k = 1,2,\ldots,n-1 \\
  1 - \lambda_p & k = n 
\end{cases} 
\]
  \( \lambda_p = 0.4 \)
Query Expansion with search results

Anchor Log

• Collected by harvestlink in the Lemur toolkit
• Extract the top 5 frequent anchor text in the previous results
• Weights are proportional to normalized frequency of anchor text

\[
\text{weight}(\lambda_1 \#\text{combine}(n\text{ugget}_t_1, n\text{ugget}_t_2, \ldots, n\text{ugget}_t_m, w_{i1}, w_{i2}, \ldots, w_{i_r}))
\]

\[
\text{weight}(\lambda_2 \#\text{combine}(n\text{ugget}_t_1, n\text{ugget}_t_2, \ldots, n\text{ugget}_t_m, w_{i1}, w_{i2}, \ldots, w_{i_r}))
\]

\[
\vdots
\]

\[
\text{weight}(\lambda_n \#\text{combine}(n\text{ugget}_t_1, n\text{ugget}_t_2, \ldots, n\text{ugget}_t_m, w_{i1}, w_{i2}, \ldots, w_{i_r}))
\]

\[
\beta \omega \_1 \# \text{combine}(e_1) \beta \omega \_2 \# \text{combine}(e_2) \ldots \beta \omega \_n \# \text{combine}(e_n)
\]

Example (TREC 2012 session 53)

#weight(1.0 #1(spinal cord) 0.6 consequences 0.4 paralysis 1.0 servering 0.38 #combine(type of paralysis) 0.0048 #combine(quadriplegia paraplegia) 0.0048 paraplegia 0.0048 #combine(spinal cord injury) 0.0024 #combine(quadriplegic tetraplegic) )

Duplicated Queries

• Suggest user’s intention

1. pocono mountains pennsylvania
2. pocono mountains pennsylvania hotels
3. pocono mountains pennsylvania things to do
4. pocono mountains pennsylvania hotels
5. pocono mountains camelbeach
6. pocono mountains camelbeach hotel
7. pocono mountains chateau resort
8. pocono mountains chateau resort attractions
9. pocono mountains chateau resort getting to
10. chateau resort getting to
11. pocono mountains chateau resort directions

Example: TREC 2012 session 6
Duplicated Queries

Assumptions

• If there is a previous query that is the same as the current query $q_n$, we only use the current query to generate the structured session query
  - The user came back to a previous query, which perhaps indicates that other previous queries are not very satisfying

• If several previous queries (other than $q_n$) are duplicated, we remove them when formulating the structured session query
  - The user changed the query after checking it twice, which indicates the results of this query is not satisfying

<table>
<thead>
<tr>
<th>Queries in a Session</th>
<th>Without removing duplicates</th>
<th>Removing duplicates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Structured query</td>
<td>nDCG@10</td>
</tr>
<tr>
<td>shoulder joint pain</td>
<td>#weight(1.4 joint 0.4 nhs 1.4 pain 1.4 shoulder) 0.036 #1(shoulder pain) 0.036</td>
<td>0.5538</td>
</tr>
<tr>
<td>shoulder joint pain nhs</td>
<td>#1(frozen shoulder) 0.01 #1(shoulder pain causes) 0.006 #1(bursitis) 0.006 #1(painful shoulder conditions)</td>
<td></td>
</tr>
<tr>
<td>shoulder joint pain</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example

TREC 2011 session 22

Used in RL3/RL4
Document Re-ranking

Dwell time: the elapsed time that a user stays in the page

\[ \Delta t = t_{end} - t_{start} \]

Clicked documents: \( \{c_1, c_2, \cdots, c_k\} \)

Associated dwell time: \( \{\Delta t_1, \Delta t_2, \cdots, \Delta t_k\} \)

Re-ranking the returned documents \( \{d_j\} \) by:

\[ s(d_j) = \sum_{i=1}^{k} \text{Sim}(d_j, c_i) \cdot \Delta t_i \]

Using raw dwell time to strongly bias towards SAT (satisfying) clicks

Submitted Runs

<table>
<thead>
<tr>
<th>run</th>
<th>RL1</th>
<th>RL2</th>
<th>RL3</th>
<th>RL4</th>
</tr>
</thead>
<tbody>
<tr>
<td>guphrase1</td>
<td>strict method ( \mu = 4000 ), ( k = 10 )</td>
<td>strict method query expansion ( \mu = 4500 ), ( k = 5 )</td>
<td>strict method query expansion anchor text remove duplicates ( \mu = 4500 ), ( k = 5 )</td>
<td>strict method query expansion anchor text remove duplicates re-ranking by time ( \mu = 4500 ), ( k = 5 )</td>
</tr>
<tr>
<td>guphrase2</td>
<td>strict method ( \mu = 3500 ), ( k = 10 )</td>
<td>strict method query expansion ( \mu = 5000 ), ( k = 5 )</td>
<td>strict method query expansion anchor text remove duplicates ( \mu = 5000 ), ( k = 5 )</td>
<td>strict method query expansion anchor text remove duplicates re-ranking by time ( \mu = 5000 ), ( k = 5 )</td>
</tr>
<tr>
<td>gurelaxphr</td>
<td>relaxed method ( \mu = 4000 ), ( k = 20 )</td>
<td>relaxed method query expansion ( \mu = 4500 ), ( k = 20 )</td>
<td>relaxed method query expansion anchor text remove duplicates ( \mu = 4500 ), ( k = 20 )</td>
<td>strict method query expansion anchor text remove duplicates re-ranking by time ( \mu = 4500 ), ( k = 5 )</td>
</tr>
</tbody>
</table>
Evaluation Results (2012)

nDCG@10 for TREC 2012 runs

<table>
<thead>
<tr>
<th>run</th>
<th>original</th>
<th>guphrase1</th>
<th>guphrase2</th>
<th>gurelaxphr</th>
<th>Mean of the median</th>
</tr>
</thead>
<tbody>
<tr>
<td>RL1</td>
<td>0.2474</td>
<td>0.2298</td>
<td>0.2265</td>
<td>0.2334</td>
<td>0.1746</td>
</tr>
<tr>
<td>RL2</td>
<td>0.2932</td>
<td>0.2839</td>
<td>0.2832</td>
<td>0.1901</td>
<td></td>
</tr>
<tr>
<td>RL3</td>
<td>0.3021</td>
<td>0.2995</td>
<td>0.3033</td>
<td>0.216</td>
<td></td>
</tr>
<tr>
<td>RL4</td>
<td>0.3021</td>
<td>0.2995</td>
<td>0.29</td>
<td>0.2261</td>
<td></td>
</tr>
</tbody>
</table>

- Terms from previous queries boost the accuracy significantly
- Big improvement from RL2 to RL1
- Removing duplicated queries improves the search accuracy
  - Improvement from RL3 to RL2
- Grouping terms into nuggets is not effective to 2012 queries
  - Might overfit on 2011 queries

Evaluation Results (2011)

nDCG@10 for TREC 2011 RL1 runs. A significant improvement over the baseline is indicated with a † at p<0.05 level and a ‡ at p<0.005 level

<table>
<thead>
<tr>
<th>Metric</th>
<th>original query</th>
<th>strict</th>
<th>relaxed</th>
<th>2011 Best</th>
<th>2011 Median</th>
</tr>
</thead>
<tbody>
<tr>
<td>nDCG@10</td>
<td>0.3378</td>
<td>0.3834</td>
<td>0.3979</td>
<td>0.3789</td>
<td>0.3232</td>
</tr>
<tr>
<td>%chg</td>
<td></td>
<td>+13.50%†</td>
<td>+17.79%‡</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Structured queries built on nuggets improve the accuracy significantly
- Relaxed method even boosts higher
  - allows larger window size, may be more suitable
Evaluation Results (2011)

nDCG@10 for TREC 2011 RL2 runs. A significant improvement over the baseline is indicated with a † at p<0.05 level and a ‡ at p<0.005 level.

<table>
<thead>
<tr>
<th>Metric</th>
<th>original query</th>
<th>uniform</th>
<th>previous vs. current</th>
<th>distance-based</th>
<th>2011 Best</th>
<th>2011 Media</th>
</tr>
</thead>
<tbody>
<tr>
<td>nDCG@10</td>
<td>0.3378</td>
<td>0.4475</td>
<td>0.4626</td>
<td>0.4431</td>
<td>0.4281</td>
<td>0.3215</td>
</tr>
<tr>
<td>%chg</td>
<td>32.47%†</td>
<td>36.94%†</td>
<td>31.17%‡</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

• Terms from previous queries significantly boost the search accuracy

*previous vs. current* outperforms other schemes

- The intention of user is complicated
- Cannot assume that the early queries are less important

---

Evaluation Results (2011)

nDCG@10 for TREC 2011 RL3 and RL4 runs. A significant improvement over the baseline is indicated with a † at p<0.05 level and a ‡ at p<0.005 level.

<table>
<thead>
<tr>
<th>Method</th>
<th>anchor text</th>
<th>nDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>all documents</td>
<td>clicked documents (RL4 only)</td>
</tr>
<tr>
<td></td>
<td>nDCG@10</td>
<td>%chg</td>
</tr>
<tr>
<td>all queries</td>
<td>0.4695</td>
<td>38.99%†</td>
</tr>
<tr>
<td>remove duplicated queries</td>
<td>0.4836</td>
<td>43.16%†</td>
</tr>
<tr>
<td>re-rank by dwell time (RL4 only)</td>
<td>0.4435</td>
<td>31.29%‡</td>
</tr>
</tbody>
</table>

• Removing duplicated queries improves the accuracy
• Re-ranking does not perform well
  - Ranking by raw dwell time might be rough
Conclusions

- Construct structured Lemur queries for session search
- What works:
  - Using previous queries
  - Eliminating duplicates
- What we believe that works
  - Using nuggets to form structured query
    - we did achieve good performance gain over 2011 data
    - ... thus keep investigating

Thank You

Grace Hui Yang
Department of Computer Science
Georgetown University
huiyang@cs.georgetown.edu

Nov 9, 2012