Query Reformulation Using Anchor Text

Van Dang and W. Bruce Croft
Query Reformulation

(MSN query log)

cheap **airfare**  
hunting **deaths**

(TREC query)

cheap **flights**  
hunting **deaths accidents**

Retrieval System
Related work

- Relevance feedback
  - Well-known, not in the scope of this paper
- Recent reformulation techniques rely on query logs
  - [Jones et al., 06], [Wang and Zhai, 08]
  - These techniques have proven effective for real web queries
    - Many of these queries are badly formulated (“cheap airfare”)
  - What if queries are good? (e.g. “hunting deaths”)
    - Can these techniques still make them better?

Do these methods work with good queries?
Related Work

- Recent reformulation techniques rely on query logs
  - [Jones, 06], [Wang and Zhai, 08]
- And so do many other tasks
  - Spelling correction: [Cucerzan et al., 04], [Ahmad et al., 05]
  - Stemming: [Peng et al., 07]
- Query logs might not be available to research community
  - Any alternatives?
- <anchor text, url> is just like <query, clicked doc>.

Can we use anchor text to simulate a query log?
Introduction

Do these methods work with good queries?

- Using TREC collections to evaluate the most recent log-based reformulation technique [Wang and Zhai, 08] on three tasks
  - Query Substitution
  - Query Expansion
  - Query Stemming

Can we use anchor text to simulate a query log?

- Uses anchor text in place of a query log
The Anchor Log

- Extract <anchor, url> pairs from the Gov-2 collection to create the anchor log.

<table>
<thead>
<tr>
<th></th>
<th>MSN Log</th>
<th>Anchor Log</th>
</tr>
</thead>
<tbody>
<tr>
<td># Total Queries</td>
<td>14 million</td>
<td>526 million</td>
</tr>
<tr>
<td># Unique Queries</td>
<td>6 million</td>
<td>20 million</td>
</tr>
<tr>
<td>Avg. Query Length</td>
<td>2.68</td>
<td>2.62</td>
</tr>
</tbody>
</table>

- The anchor log is very noisy
  - “click here”, “print version”, … don’t represent the linked page
A context of a word is the unigram preceding it

**Context distribution**

\[
P(c_i \mid w) = \frac{\text{count}_w(c_i)}{\sum_{c_j \in C(w)} \text{count}_w(c_j)}
\]

The probability that the term \(c_i\) appears in \(w\)'s context

**The translation model**

\[
t(s \mid w) = \frac{e^{-D(P(\cdot \mid w) \parallel P(\cdot \mid s))}}{Z}
\]

The KL divergence between the context distributions of \(w\) and \(s\)

**The substitution model**

- \(Q = q_1, \ldots, q_{i-2}, q_{i-1}, q_i, q_{i+1}, q_{i+2}, \ldots\) and \(\text{candidate} = s\)

\[
P(w_i \rightarrow s) = t(s \mid w_i) \times P(q_{i-2}q_{i-1} - q_{i+1}q_{i+2} \mid s)
\]

How fit the new term is to the context of the current query
**Substitution: An example**

**Query Log**

- cheap airfare
  - 0.01

- inexpensive airfare
  - 0.001

**Substitution model**

\[
P(w_i \rightarrow s) = t(s \mid w_i) \times P(q_{i-2}q_{i-1} - q_{i+1}q_{i+2} \mid s)
\]

**Translation model**

- cheap $\rightarrow$ inexpensive 0.02
- airfare $\rightarrow$ flight 0.10
- airfare $\rightarrow$ ticket 0.12

- cheap ticket 0.03
- cheap flight 0.15
Query Expansion and Stemming

- Query Expansion is exactly the same as substitution
  - We add the new term and keep the original term
    substitution: “cheap airfare” → “cheap flight”
    expansion: “cheap airfare” → “cheap airfare flight”

- Stemming
  - New terms are restricted to Porter-stemmed root terms
    “drive direction” → “drive driving direction”
Experimental Setup

- **Evaluation**
  - Conducted on three TREC collections:
    - Robust-04 (news)
    - WT10G (web)
    - Gov-2 (web)

<table>
<thead>
<tr>
<th>Collection</th>
<th># Documents</th>
<th># Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Robust-04</td>
<td>0.5 M</td>
<td>250</td>
</tr>
<tr>
<td>WT10G</td>
<td>1.5 M</td>
<td>100</td>
</tr>
<tr>
<td>Gov-2</td>
<td>25 M</td>
<td>150</td>
</tr>
</tbody>
</table>

- **Title queries vs. Description queries**
## Evaluation of Reformulated Query

<table>
<thead>
<tr>
<th>Original Queries</th>
<th>MSN-Log Substitution</th>
<th>Anchor-Log Substitution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Query 1</td>
<td>Substitution 1</td>
<td>Substitution 1</td>
</tr>
<tr>
<td></td>
<td>Substitution 2</td>
<td>Substitution 2</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Substitution m</td>
<td>Substitution m</td>
</tr>
<tr>
<td>Query n</td>
<td>Substitution 1</td>
<td>Substitution 1</td>
</tr>
<tr>
<td></td>
<td>Substitution 2</td>
<td>Substitution 2</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td></td>
<td>Substitution m</td>
<td>Substitution m</td>
</tr>
<tr>
<td>P@5</td>
<td>P@5</td>
<td>P@5</td>
</tr>
</tbody>
</table>
Substitution vs. Expansion (Title Q.)

The Anchor log is comparable to the MSN Log.
“Chance” vs. “Risk”

- Substitution works for web queries [Wang and Zhai, 08]
  - Does not work here
  - Expansion is much better
  - Why?

- Both Substitution and Expansion
  - Introduce a new term to the query
    - “chance”: it brings more relevant documents
    - “risk”: it brings more non-relevant documents
“Chance” vs. “Risk”

- **Results**
  - Among 99 queries that were reformulated

<table>
<thead>
<tr>
<th></th>
<th># Queries</th>
<th>P@5 change</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Substitution</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Substitution helps</td>
<td>34</td>
<td>+110.94%</td>
</tr>
<tr>
<td>Expansion helps</td>
<td>32</td>
<td>+88.72%</td>
</tr>
<tr>
<td><strong>Expansion</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Substitution hurts</td>
<td>32</td>
<td>-55.29%</td>
</tr>
<tr>
<td>Expansion hurts</td>
<td>14</td>
<td>-53.85%</td>
</tr>
</tbody>
</table>

**Substitution**
- Helps substantially
- Hurts drastically
- Does NOT help in general

**Expansion**
- Helps more than it hurts, thus better
“Chance” vs. “Risk”

- Translation model does NOT provide « synonyms »
  - \{women, men, children\}
  - \{diamond, gold, necklace, watches\}

- It is undesirable to
  - “diamond smuggling” $\rightarrow$ “watches smuggling”

- TREC queries have good quality
  - Complete substitution is too risky
Substitution vs. Expansion (Desc Q.)

The Anchor log is comparable to the MSN Log.
Substitution good for Long Query?

- Substitute \( w \) for \( s = \) drop \( w \) + add \( s \)
  - \( Q_{\text{org}} \): original query
  - \( Q_{\text{drop}} \): drop the target word
  - \( Q_{\text{add}} \): add the substitution candidate

<table>
<thead>
<tr>
<th>MSN Log</th>
<th>WT10G</th>
<th>Robust04</th>
<th>Gov-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_short</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT10G</td>
<td>0.3291</td>
<td>0.2734</td>
<td>0.3468</td>
</tr>
<tr>
<td>Robust04</td>
<td>0.4786</td>
<td>0.4009</td>
<td>0.4937</td>
</tr>
<tr>
<td>Gov-2</td>
<td>0.5632</td>
<td>0.4529</td>
<td>0.5515</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Long Q</th>
<th>WT10G</th>
<th>Robust04</th>
<th>Gov-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q_short</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WT10G</td>
<td>0.3158</td>
<td>0.3074</td>
<td>0.3768</td>
</tr>
<tr>
<td>Robust04</td>
<td>0.4764</td>
<td>0.5138</td>
<td>0.5976</td>
</tr>
<tr>
<td>Gov-2</td>
<td>0.5238</td>
<td>0.5578</td>
<td>0.6612</td>
</tr>
</tbody>
</table>

- Dropping hurts
- Dropping helps [Kumaran et al., 09]

Similar improvement

- It is the dropping that helps
Stemming

- We compare using P@10 queries
  - Unstemmed
  - Krovetz
  - Log-based (MSN vs. Anchor Log)

The Anchor log is comparable to the MSN Log.
Conclusions

- Anchor text gives comparable performance to MSN log on
  - Substitution
  - Expansion
  - Stemming

- Expansion is more reliable than substitution

- Substitution helps with long (desc) queries
  - It is the dropping that helps

- Log-based stemming is promising