THUIR@NTCIR-12 IMine

Zeyang Liu, Ye Chen, Rongjie Cai, Jiaxin Mao, Chao Wang, Cheng Luo, Xin Li, Yiqun Liu, Min Zhang, Huanbo Luan, Shaoping Ma

Tsinghua University

Jul 7th, 2016
NTCIR: Imine-2

- **Search Intent and Task Mining**
- **Goal:** explore and evaluate the technologies of understanding user intents behind the query and satisfying different user intents.

- **Subtasks:**
  - **Query Understanding(C):** generate a *diversified* ranked list of subtopics with corresponding *vertical intents*
  - **Vertical Incorporating(C):** return a *diversified* ranked list of not more than 100 results, including organic documents and *virtual vertical results.*
Query Understanding Framework

Candidate Ranking (Learning to Rank)

Candidate Subtopic Clustering (K-means)

Candidate Queries

Candidate Vertical Intents (Logistic Regression)

Vertical Diversifying

Training Dataset

Seed Queries

Commercial Search Engine Result Pages

Baidu Baike
Hudong Baike
Query facets
Wikipedia
SogouT

Q-Runs
Candidate Mining

• Disambigous items
  • Wikipedia, Baidu Baike and Hudong Baike

• Query Facets
  • Query completions and query suggestions

• Query Recommendations
  • Crawled from Sogou and Baidu

• Query reformulations
  • Extracted from SogouT dataset
Candidate Mining

• Query reformulations
  • SogouT dataset
  • Session detection [Catledge et al., 1995]
  • Find out sessions containing the task query
  • Queries in such sessions are regarded as query reformulations
Subtopic Candidate Clustering

• Background: Candidates from different resources are duplicated
• Goal: Find diversified candidates
• Cluster candidates with K-means algorithm
• Query vector representation
  • Word embedding trained based on SogouT dataset
  • Long query candidate: average word embedding of its’ words
• Cluster subtopic candidates into n clusters
  • n = 5 or 10
Subtopic Candidate Clustering

- Subtopic candidate evaluation
  - Clusters: $c_1, c_2, \cdots, c_n$
  - Inner distance
    $$S_{\text{inner}}(q) = \frac{\sum_{q_k \in c_i, q_k \neq q} \text{dist}(q, q_k)}{|c_i| - 1}$$
  - Outer distance
    $$S_{\text{outer}}(q) = \min\left\{\frac{\sum_{q_k \in c_j} \text{dist}(q, q_k)}{|c_j|} \right\} (j = 1, 2, \cdots, n, j \neq i)$$
  - Candidate score
    $$S(q) = \frac{S_{\text{outer}}(q) - S_{\text{inner}}(q)}{\max\{S_{\text{outer}}(q), S_{\text{inner}}(q)\}}$$
Candidate Ranking

• Background: Candidates are noisy

• Goal: Find high quality subtopic candidates

• Rank candidates with Learning To Rank algorithm

• Features:
  • Text similarity: length difference, Jaccard similarity, edit distance...
  • Word embedding: average, medium, top 3 average of cosine similarities
  • Search Result Similarity: number of shared results...

• Metric to optimize: NDCG@10

• Training set: Ranked Subtopics from NTCIR-11 Imine
# Candidate Ranking

<table>
<thead>
<tr>
<th>Method</th>
<th>No Normalization</th>
<th>Normalized by Sum</th>
<th>Normalized by Z-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>MART</td>
<td>0.5133</td>
<td>0.5426</td>
<td>0.5043</td>
</tr>
<tr>
<td>RankNet</td>
<td>0.5365</td>
<td>0.5328</td>
<td>0.6221</td>
</tr>
<tr>
<td><strong>RankBoost</strong></td>
<td><strong>0.6618</strong></td>
<td>0.6560</td>
<td>0.5467</td>
</tr>
<tr>
<td>AdaRank</td>
<td>0.5428</td>
<td>0.5560</td>
<td>0.5468</td>
</tr>
<tr>
<td>Coordinate Ascent</td>
<td>0.6157</td>
<td>0.5839</td>
<td>0.6352</td>
</tr>
<tr>
<td>LambdaRank</td>
<td>0.5349</td>
<td>0.4943</td>
<td>0.5796</td>
</tr>
<tr>
<td>LambdaMART</td>
<td>0.5306</td>
<td>0.5386</td>
<td>0.5199</td>
</tr>
<tr>
<td>ListNet</td>
<td>0.5543</td>
<td>0.5829</td>
<td>0.5783</td>
</tr>
</tbody>
</table>
Query Understanding Framework

Candidate Queries

Baidu Baike
Hudong Baike
Query facets
Wikipedia
SogouT

Candidate Subtopic Clustering (K-means)

Candidate Ranking (Learning to Rank)

Vertical Diversifying
Candidate Vertical Intents (Logistic Regression)

Q-Runs

Training Dataset
Seed Queries

Commercial Search Engine Result Pages
Vertical Predicting

• Training Query

• 1212 Seed URLs generated from an Open Directory Project (ODP)

• News, videos, shopping, communities and games

<table>
<thead>
<tr>
<th>Category</th>
<th>Subcategory</th>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Video</td>
<td>Movie</td>
<td><a href="http://movie.youku.com/">http://movie.youku.com/</a></td>
<td>an online video site</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://www.iqiyi.com/dianying/">http://www.iqiyi.com/dianying/</a></td>
<td>an online video site</td>
</tr>
<tr>
<td></td>
<td>Television</td>
<td><a href="http://cctv.cntv.cn/">http://cctv.cntv.cn/</a></td>
<td>the official website of CCTV</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://www.brtn.cn/">http://www.brtn.cn/</a></td>
<td>the official website of Beijing Television Station</td>
</tr>
<tr>
<td></td>
<td>News of Movie</td>
<td><a href="http://ent.sina.com.cn/film/">http://ent.sina.com.cn/film/</a></td>
<td>a news website about movies and celebrities</td>
</tr>
<tr>
<td></td>
<td></td>
<td><a href="http://ent.163.com/">http://ent.163.com/</a></td>
<td>a news website about movies and celebrities</td>
</tr>
</tbody>
</table>
Vertical Predicting

• Random walk
  • Based on a click-through bipartite graph
  • Expand candidate queries based on the seed urls

• Vertical Distribution
  • Vertical information collected from search engine result page
  • Vertical types extracted from CSS styles
Vertical Predicting

• Random walk
  • Based on a click-through bipartite graph
  • Expand candidate queries based on the seed urls

• Vertical Distribution
  • Vertical information collected from search engine result page
  • Vertical types extracted from CSS styles
  • Presentation score

\[ P-score = \begin{cases} 
\frac{1}{\log(1 + R_i)} & \text{if exists} \\
0 & \text{else}
\end{cases} \]
Vertical Predicting

• Model Construction
  • Logistic Regression
  • Six prediction models for each type of verticals
  • Input: Query representation based on word embedding
  • Output: presentation score of one specific type of vertical
  • Vertical type with the highest score is chosen to be the vertical intent
Vertical Diversification

• Background: strong effect of web vertical bias
  • Web verticals occupy a large portion of search results in practical
  • The prediction performance is limited
Vertical Diversification

• Background: strong effect of web vertical bias
  • Web verticals occupy a large portion of search results in practical
  • The prediction performance is limited

• Empirical rules for queries with web vertical intent
  • If the top result is a vertical result: replaced with the corresponding vertical intent
  • If there are two verticals in the top 3 positions: replaced with the highest ranked vertical
<table>
<thead>
<tr>
<th>RUNNAME</th>
<th>SYSTEM DESC.</th>
<th>D#-nDCG</th>
<th>V-score</th>
<th>QU-score</th>
</tr>
</thead>
<tbody>
<tr>
<td>THUIR-QU-1A</td>
<td>Cluster all subtopic candidates into 10 clusters and select the candidate with the highest S(q) from each cluster.</td>
<td>0.5204</td>
<td>0.5579</td>
<td>0.5392</td>
</tr>
<tr>
<td>THUIR-QU-2A</td>
<td>Cluster all subtopic candidates into 5 clusters and select two candidates with the highest two S(q) from each cluster.</td>
<td>0.5550</td>
<td>0.5506</td>
<td>0.5528</td>
</tr>
<tr>
<td>THUIR-QU-1B</td>
<td>Rerank the 10 subtopics generated by THUIR-QU-1A with learning to rank algorithm.</td>
<td>0.5368</td>
<td>0.5763</td>
<td><strong>0.5565</strong></td>
</tr>
<tr>
<td>THUIR-QU-2B</td>
<td>Rerank the 10 subtopics generated by THUIR-QU-2A with learning to rank algorithm.</td>
<td>0.5436</td>
<td>0.5686</td>
<td><strong>0.5561</strong></td>
</tr>
<tr>
<td>THUIR-QU-3A</td>
<td>Cluster all subtopic candidates into 5 clusters and select the candidate with the highest ten S(q).</td>
<td>0.4973</td>
<td>0.5942</td>
<td>0.5458</td>
</tr>
</tbody>
</table>
Vertical Incorporating

• Retrieval Models for Organic Results:

  • Probabilistic model based on BM25 and our previous proposed word pair model

  • Relevance score for subtopic

    • \( R(q, D) = W_{BM25} + \alpha \cdot W_{wp} \)

    • \( W_{BM25} = \sum_{i=1}^{m} \log\frac{n - n(q_i) + 0.5}{n(q_i) + 0.5} \cdot \frac{f(q_i,D)\cdot(k_1+1)}{f(q_i,D)+k_1\cdot(1-b+b\cdot\frac{|D|}{avgdl})} \)

    • \( W_{wp} = \sum_{i=1}^{m} \log\frac{n - n(q_iq_{i+1}) + 0.5}{n(q_iq_{i+1}) + 0.5} \cdot \frac{f(q_iq_{i+1},D)\cdot(k_1+1)}{f(q_iq_{i+1},D)+k_1\cdot(1-b+b\cdot\frac{|D|}{avgdl})} \)

  • Relevance score for query

    • \( R(Q, D) = \sum_{i=1}^{10} R(q_i, D) \times S(q_i) \)
Vertical Incorporating

• Vertical Result Ranking
  • Vertical importance based on subtopic candidate score
    \[ I(v) = \alpha \cdot S-score(v) \]
  • \( v \): a type of vertical intent
  • \( S-score(v) \): the score of the subtopic which contains this vertical intent
  • \( \alpha \): importance weight, different for different types of verticals
  • Combination of \( R(Q, D) \) and \( I(v) \)
Vertical Incorporating

• Vertical Result Ranking
  • Vertical importance based on subtopic candidate score

\[ I(v) = \alpha \cdot S\text{-score}(v) \]

• \( v \): a type of vertical intent
• \( S \text{-score}(v) \): the score of the subtopic which contains this vertical intent
• \( \alpha \): importance weight, different for different types of verticals
• Combination of \( R(Q, D) \) and \( I(v) \)
### Vertical Incorporating Results

<table>
<thead>
<tr>
<th>RUNNAME</th>
<th>D#-nDCG (unclear topics)</th>
<th>nDCG (clear topics)</th>
<th>D#-nDCG+nDCG (all topics)</th>
</tr>
</thead>
<tbody>
<tr>
<td>THUIR-QU-1A</td>
<td>0.6677</td>
<td>0.5756</td>
<td><strong>0.6594</strong></td>
</tr>
<tr>
<td>THUIR-QU-2A</td>
<td>0.6664</td>
<td>0.5652</td>
<td><strong>0.6573</strong></td>
</tr>
<tr>
<td>THUIR-QU-1B</td>
<td>0.6594</td>
<td>0.5416</td>
<td><strong>0.6488</strong></td>
</tr>
<tr>
<td>THUIR-QU-2B</td>
<td>0.6632</td>
<td>0.5442</td>
<td><strong>0.6525</strong></td>
</tr>
<tr>
<td>THUIR-QU-3A</td>
<td>0.6429</td>
<td>0.5506</td>
<td><strong>0.6346</strong></td>
</tr>
</tbody>
</table>
Thank you!

chenye617@gmail.com
www.thuir.cn