Selective Search: A distributed search architecture reduces computational costs

Resource Selection: Selects shards that are likely to have relevant documents to the query
- **Term-based**: store a language model for each shard, using term statistics (term frequency in shard, etc.)
- **Sample-based**: run the query in a small sample of the collection. More accurate, but slower
- **Supervised**: train a classifier for each shard. Expensive when hundreds of shards (Jnt)

Motivation:
- Most resource selection algorithms are heuristic
- The few learned resource selection algorithms are expensive to apply at scale (hundreds of index shards)

Learning-To-Rank Resources:
- An efficient approach to learn resource selection: A single model applied to all shards. Pairwise learning-to-rank with new features
- Automatically generate training labels

Training Labels

Two Definitions of Ground Truth
1. Relevance-based
   - The number of relevant documents a shard contains
   - Training data require queries with relevance judgments. Expensive
2. Overlap-based
   - The number of documents in a shard that were ranked highly by exhaustive search
   - No manual judgement required
   - Can be automatically generated

Features

1. Query-Independent Information
   - Shard Popularity
2. Term Based Statistics
   - Tally: score, inverse rank (1/r), binned rank (r/10)
   - Champion List Features: \( \sum_{\text{term } t \in \text{query}} \# \text{of documents the shard contributes to the term } t\)'s top-k document
   - Shard Query Likelihood: \( p(\text{term} | \text{shard}) \)
   - Query Term Statistics: min-shardTF, min-shardTF * IDF, max-shardTF, max-shardTF * IDF
   - Bigram Log Frequencies: estimates term co-occurrence \( \sum_{\text{bigram } b \in \text{shard}} \log(\text{frequency of bigram } b \text{ in shard}) \)

Experiments

Dataset
- CW09-B: 123 shards, 200 test queries
- Gov2: 199 shards, 150 test queries
- Select top 6% of total shards

Proposed Methods
- L2R-TREC: relevance-based, 200 or 150 queries, 10-fold cross-validation
- L2R-AOL: overlap-based, 1000 AOL queries
- L2R-MQT: overlap-based, 1000 MQT queries

Baselines
- Term-based: Tally
- Sample-based: ReDDE, RankS, Jnt
- Supervised: Jnt

Exhaustive Search (Exh): Searching all shards

FAST vs. SLOW

<table>
<thead>
<tr>
<th>Method</th>
<th>P@10</th>
<th>NDCG@30</th>
<th>MAP@1000</th>
<th>Average Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cw99</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redde</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tally</td>
<td>0.363*</td>
<td>0.275*</td>
<td>0.187</td>
<td>156,180</td>
</tr>
<tr>
<td>Jnt</td>
<td>0.366</td>
<td>0.260</td>
<td>0.175</td>
<td>470</td>
</tr>
<tr>
<td>ALL</td>
<td>0.375*</td>
<td>0.286*</td>
<td>0.202</td>
<td>158,529</td>
</tr>
<tr>
<td>FAST</td>
<td>0.373*</td>
<td>0.285*</td>
<td>0.201*</td>
<td>2,349</td>
</tr>
<tr>
<td>Gov2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Redde</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tally</td>
<td>0.579*</td>
<td>0.445*</td>
<td>0.289</td>
<td>105,080</td>
</tr>
<tr>
<td>Jnt</td>
<td>0.588*</td>
<td>0.465*</td>
<td>0.292</td>
<td>313,875</td>
</tr>
<tr>
<td>ALL</td>
<td>0.593*</td>
<td>0.474*</td>
<td>0.309*</td>
<td>108,306</td>
</tr>
<tr>
<td>FAST</td>
<td>0.587*</td>
<td>0.471*</td>
<td>0.310*</td>
<td>3,226</td>
</tr>
</tbody>
</table>

FAST feature set:
- Query independent feature and term based statistics
ALL feature set:
- Slower. Sample-document features are slow

FAST is
- ... as accurate as exhaustive search
- and ALL
- ... but 100+ times faster than ALL

Non-inferior To Exhaustive
- All Baselines: 10% gap from exhaustive
- L2R: Searching for 6% shards is statistically non-inferior to searching all shards exhaustively, even for the recall-oriented MAP@1000

Manual Label Not Necessary
- L2R-AOL and L2R-MQT are not worse than L2R-TREC in most cases
- Overlap-based training is as good as relevance-based
- Does not require manual label

Conclusions
- Training data can be generated automatically using a slower system that searches all index shards.
- Comparable to exhaustive search down to rank 1,000. Make it possible to apply a document re-ranker.
- The slower sample-document features provide only a small gain. No longer need to make a choice between accuracy and query latency.

SIGIR 2017, Tokyo, Japan