Dynamic Shard Cutoff Prediction for Selective Search

Hafeezul Rahman Mohammad
Keyang Xu
Jamie Callan

J. Shane Culpepper

Carnegie Mellon University
Language Technologies Institute

RMIT University
Selective search is a recent distributed search architecture

- During indexing, split the corpus into small, topical index shards
Introduction: Selective Search

Selective search is a recent distributed search architecture

- During indexing, split the corpus into small, topical index shards

- Use resource selection to pick shards for query q
  1. Rank the index shards

ClueWeb09

\[ S_1 \quad S_3 \quad \cdots \quad S_23 \quad S_{19} \quad S_{47} \quad S_{37} \quad \vdots \]
Selective search is a recent distributed search architecture

• During indexing, split the corpus into small, topical index shards

• Use resource selection to pick shards for query q
  1. Rank the index shards
  2. Decide how many shards to search
Introduction: Selective Search

Selective search is a recent distributed search architecture:

- During indexing, split the corpus into small, topical index shards.
- Use resource selection to pick shards for query $q$.
  1. Rank the index shards.
  2. Decide how many shards to search.
  3. Search the (few) selected shards.

Usually evaluated using an early precision metric:
- $P@10$, $NDCG@30$
Introduction:
Motivation

The number of shards selected impacts performance

• Selecting too few: Hurts document retrieval accuracy
• Selecting too many: Costly and inefficient

Previous shard selection algorithms include:

• ReDDE, L2RR: Static cutoff
• Taily, Rank-S: Tightly linked with shard ranking
• ShRkC: Independent of shard ranker
Introduction: Motivation

Prior studies focus on early precision in selective search

• Multi-stage ranking pipelines are now common
• As an early stage retrieval step, recall should be a priority
• Later rankers in the pipeline will re-rank these documents
Predicting Shard Ranking Cutoffs

**Problem:** Given query $q$, predict the shard cutoff $k$

**Solution:** Treat this as a regression problem
- Easy to tune for early precision or high recall

**Key elements to be addressed**
- Features
- Learning algorithms
- Training data

**Talks are short this year, so this talk skips many details**
- See the paper for details
Predicting Shard Ranking Cutoffs: Features

147 (query, corpus) features
• Typical query-difficulty features
• Eg., Variance of similarity scores

42 shard distribution features
• Characterize the different score distribution across shards
• Eg., Entropy of similarity scores across shards
Predicting Shard Ranking Cutoffs: Learning Algorithms

**Algorithms**

- Quantile Regression (QR)
  - Often better for predicting skewed distributions
  - Modification of RF that estimates conditional median
  - Parameterized by $\tau$

- Random Forest (RF) regressor
  - Less effective, so not covered in the talk
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?

1. Create an exhaustive search ranking ($r_{d,e}$)

Search all shards $q \rightarrow S_1 \quad S_2 \quad S_3 \quad S_4 \quad \ldots$
What is the ‘right’ number of shards $k$ to search for query $q$?

1. Create an exhaustive search ranking $(r_{d,e})$

Search all shards

Document rankings are returned

$\begin{align*}
S_1 & : d_{14} \\
S_2 & : d_1 \\
S_3 & : d_2 \\
S_4 & : d_{41} \\
\vdots & : \vdots \\
\vdots & : \vdots
\end{align*}$
What is the ‘right’ number of shards $k$ to search for query $q$?

1. Create an exhaustive search ranking $(r_{d,e})$

Search all shards $q \rightarrow S_1 \rightarrow S_2 \rightarrow S_3 \rightarrow S_4 \ldots$

Document rankings are returned

$\begin{array}{c}
d_{14} \\
d_{23} \\
\vdots \\
d_{32} \\
d_{41} \\
\end{array}$

Merge rankings to produce a final ranked list $(r_{d,e})$
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards: $q \rightarrow S_{12} \rightarrow S_{8} \rightarrow S_{31} \rightarrow S_{72} \rightarrow \cdots$
What is the ‘right’ number of shards $k$ to search for query $q$?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards

Same document rankings as Step 1
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards

$\begin{array}{c}
1 \\
S_{12} \\
\downarrow \\
d_{21} \\
\vdots
\end{array} \quad \begin{array}{c}
2 \\
S_{8} \\
\downarrow \\
d_{1} \\
\vdots
\end{array} \quad \begin{array}{c}
3 \\
S_{31} \\
\downarrow \\
d_{2} \\
\vdots
\end{array} \quad \begin{array}{c}
4 \\
S_{72} \\
\downarrow \\
d_{63} \\
\vdots
\end{array} \ldots$

Same document rankings as Step 1

Iterate over potential cutoffs
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards $q \rightarrow S_{12} \rightarrow d_{21} \rightarrow \cdots$

Same document rankings as Step 1

Merge $k=1$ rankings to produce a final ranked list ($r_{d,k}$)
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards k to search for query q?
2. Find a cutoff that produces a similar selective search ranking

Rank the shards

```
S_{12} | S_8 | S_{31} | S_{72} | ... \\
\downarrow | \downarrow | \downarrow | \downarrow | \downarrow \\
{d_{21}} | {d_1} | {d_2} | {d_{63}} | ... \\
\vdots   | \vdots   | \vdots   | \vdots   | ...
```

Same document rankings as Step 1

Merge k=1 rankings to produce a final ranked list \((r_{d,k})\)

If Close Enough \((r_{d,k}, r_{d,e})\)
Stop & report cutoff = 1
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards $q \rightarrow S_{12}, S_8, S_{31}, S_{72}, \ldots$

Same document rankings as Step 1

Merge $k=2$ rankings to produce a final ranked list $(r_{d,k})$

If $\text{Close Enough} (r_{d,k}, r_{d,e})$
Stop & report cutoff $= 2$
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards \( k \) to search for query \( q \)?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards

Same document rankings as Step 1

Merge \( k=3 \) rankings to produce a final ranked list \( (r_{d,k}) \)

If Close Enough \( (r_{d,k}, r_{d,e}) \)

Stop & report cutoff = 3
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the ‘right’ number of shards $k$ to search for query $q$?

2. Find a cutoff that produces a similar selective search ranking

Rank the shards $q \rightarrow$

Same document rankings as Step 1

Continue until a good cutoff is found or $k=16$ (cap for outlier queries)
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

Vary the definition of ‘close enough’ to satisfy different goals

Early Precision
Overlap in top 100 documents

High Recall
Overlap in top 1,000 documents

\[ d_{211} \]
\[ d_{29} \]
\[ \vdots \]
\[ d_{107} \]
\[ d_{231} \]
\[ \vdots \]
\[ d_{87} \]
\[ \vdots \]
\[ r_{d,k} \]
\[ d_{29} \]
\[ d_{1} \]
\[ \vdots \]
\[ d_{201} \]
\[ \vdots \]
\[ d_{76} \]
\[ \vdots \]
\[ r_{d,e} \]
Experimental Methodology

Datasets: ClueWeb09-B (Gov2 shown in paper)

Metrics

- Early-precision: $P@5$, $NDCG@10$, $Overlap@100$
- High-recall: $MAP@1000$, $RBP\ (p=0.95)$, $Overlap@5000$
- Efficiency: $C_{RES}$ (total cost), $C_{LAT}$ (latency)
- Agreement: Pearson (PCC), Mean Absolute Error (MAE)

Baselines

- Shard ranking: Taily, Rank-S, ReDDE, L2RR
- Shard cutoff: Taily, Rank-S, ShRkC
**Experiment 1:**
Cutoff Prediction Comparisons

**RQ1:** How accurate are existing shard cutoff predictions?

### ClueWeb09-B

<table>
<thead>
<tr>
<th></th>
<th>Early-Precision</th>
<th>High-Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rank-S</td>
<td>Taily</td>
</tr>
<tr>
<td>MAE</td>
<td>1.31</td>
<td>1.34</td>
</tr>
<tr>
<td>PCC</td>
<td>0.37</td>
<td>0.34</td>
</tr>
</tbody>
</table>

**Lower MAE & higher PCC:** Better at predicting k

**The Learned predictor is best under both scenarios**
**RQ3:** Are ranker-independent cutoff predictions effective?

**ClueWeb09-B**

<table>
<thead>
<tr>
<th></th>
<th>Early-Precision</th>
<th>High-Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rank-S Taily ShRkC QR Rank-S Taily ShRkC QR</td>
<td></td>
</tr>
<tr>
<td><strong>MAE</strong></td>
<td>1.31 1.34 2.99  1.14</td>
<td>2.91 2.84 4.85  1.94</td>
</tr>
<tr>
<td><strong>PCC</strong></td>
<td>0.37 0.34 0.26  0.44</td>
<td>0.38 0.39 0.28  0.64</td>
</tr>
</tbody>
</table>

**Lower MAE & higher PCC:** Better at predicting k

**Ranker-independent cutoff predictions can be effective**

- QR is, but ShRkC is not
Experiment 1: Cutoff Prediction Comparisons

Shard cutoff biases

- Closer to the ‘Label’ curve is desired
- Taily tends to under predict
Experiment 1: Cutoff Prediction Comparisons

Shard cutoff biases

- Closer to the ‘Label’ curve is desired
- Taily tends to under predict
- Rank-S and ShRkC tend to over predict
Experiment 1: Cutoff Prediction Comparisons

Shard cutoff biases

• Closer to the ‘Label’ curve is desired
• Taily tends to under predict
• Rank-S and ShRkC tend to over predict
• QR is the most accurate
Experiment 2: Shard Ranking Comparisons

**RQ2:** How accurate are existing shard rankings?

- Examine **shard ranking & cutoff prediction** separately
  - Usually these problems are conflated

- In this experiment, each ranker uses a fixed number of shards
  - Given by ‘Label’ (the gold standard)
### Experiment 2: Shard Ranking Comparisons

<table>
<thead>
<tr>
<th>Ranking</th>
<th>MAP</th>
<th>RBP,0.95</th>
<th>O@5000</th>
<th>$C_{RES}$</th>
<th>$C_{LAT}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taily</td>
<td>.180</td>
<td>.261 (.339)</td>
<td>.599</td>
<td>.811</td>
<td>.187</td>
</tr>
<tr>
<td>Rank-S</td>
<td>.181</td>
<td>.279 (.349)</td>
<td>.612</td>
<td>.811</td>
<td>.190</td>
</tr>
<tr>
<td>ReDDE</td>
<td>.182</td>
<td>.281 (.345)</td>
<td>.618</td>
<td>.853</td>
<td>.198</td>
</tr>
<tr>
<td>L2RR</td>
<td>.196</td>
<td>.293 (.304)</td>
<td>.626</td>
<td>.896</td>
<td>.199</td>
</tr>
<tr>
<td>$r_s,e$</td>
<td>.202</td>
<td>.301 (.286)</td>
<td>.709</td>
<td>.850</td>
<td>.195</td>
</tr>
</tbody>
</table>

- L2RR is the most accurate shard ranker
- Rankers tend to select smaller (Taily) or larger (L2RR) shards
  - All rankers searched the same number of shards
### Experiment 2: Shard Ranking Comparisons

<table>
<thead>
<tr>
<th>Ranking</th>
<th>P@5</th>
<th>NDCG@10</th>
<th>O@100</th>
<th>C&lt;sub&gt;RES&lt;/sub&gt;</th>
<th>C&lt;sub&gt;LAT&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Taily</td>
<td>.370</td>
<td>.214</td>
<td>.623</td>
<td>.508</td>
<td>.180</td>
</tr>
<tr>
<td>Rank-S</td>
<td>.375</td>
<td>.229</td>
<td>.673</td>
<td>.517</td>
<td>.178</td>
</tr>
<tr>
<td>ReDDE</td>
<td>.386</td>
<td>.229</td>
<td>.708</td>
<td>.551</td>
<td>.190</td>
</tr>
<tr>
<td>L2RR</td>
<td>.389</td>
<td>.234</td>
<td>.734</td>
<td>.560</td>
<td>.189</td>
</tr>
<tr>
<td>r&lt;sub&gt;s,e&lt;/sub&gt;</td>
<td>.409</td>
<td>.247</td>
<td>.818</td>
<td>.534</td>
<td>.187</td>
</tr>
<tr>
<td>Exhaustive</td>
<td>.390</td>
<td>.240</td>
<td>-</td>
<td>5.24</td>
<td>.330</td>
</tr>
</tbody>
</table>

- L2RR is the most accurate shard ranker
- Rankers tend to select smaller (Taily) or larger (L2RR) shards
  - All rankers searched the same number of shards
**Experiment 3: Precision vs Recall**

**RQ4:** How do the competing goals of precision and recall affect efficiency-effectiveness tradeoff?

Up is more accurate  
Left is more efficient  
Goal is to be close to $r_{s,e}$

QR’s $\tau$ enables tuning efficiency vs effectiveness tradeoff

- $\tau = 0.45$ works well
Experiment 3: Precision vs Recall

RQ4: How do the competing goals of precision and recall affect efficiency-effectiveness tradeoff?

Up is more accurate
Left is more efficient
Goal is to be close to $r_{s,e}$

QR’s $\tau$ enables tuning efficiency vs effectiveness tradeoff
• $\tau = 0.45$ works well
Experiment 3: Precision vs Recall

RQ4: How do the competing goals of precision and recall affect efficiency-effectiveness tradeoff?

Up is more accurate
Left is more efficient
Goal is to be close to $r_{s,e}$

QR’s $\tau$ enables tuning efficiency vs effectiveness tradeoff

- $\tau = 0.45$ works well
Experiment 4: Training Labels Comparisons

RQ5: Should the shard cutoff prediction be trained for a specific resource selection algorithm?

• Any shard ranking can generate training data for the QR predictor – E.g., Exhaustive search (previous experiments), Taily, L2RR, ..

Conclusion

• Training with rankings based on exhaustive search produces more aggressive cutoffs
• Aggressive cutoffs work well with strong rankers (L2RR)
• Weaker rankers (Taily) benefit from ranker-specific training
• See the paper for details
Conclusions

Shard ranking & cutoff prediction should be studied separately
• Distinct problems, separate sources of error

Cutoff prediction can be done well by quantile regression
• Query difficulty and shard distribution features
• Tune for early-precision or high-recall requirements as needed
• Use with any shard ranker

Selective search can achieve high-recall
• 70% agreement with exhaustive search rankings at depth 5000 can be attained with 16-18% of the computational effort
Thank you!

Questions?