Dynamic Shard Cutoff Prediction for Selective Search

Hafeezul Rahman Mohammad
Keyang Xu
Jamie Callan

Carnegie Mellon University
Language Technologies Institute

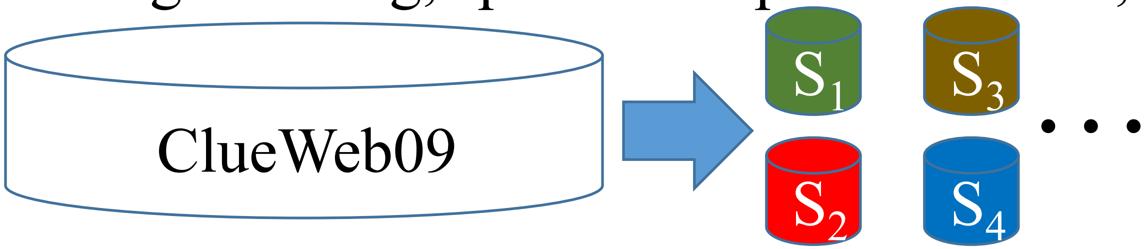
J. Shane Culpepper

RMIT UNIVERSITY

Introduction:
Selective Search

Selective search is a recent distributed search architecture

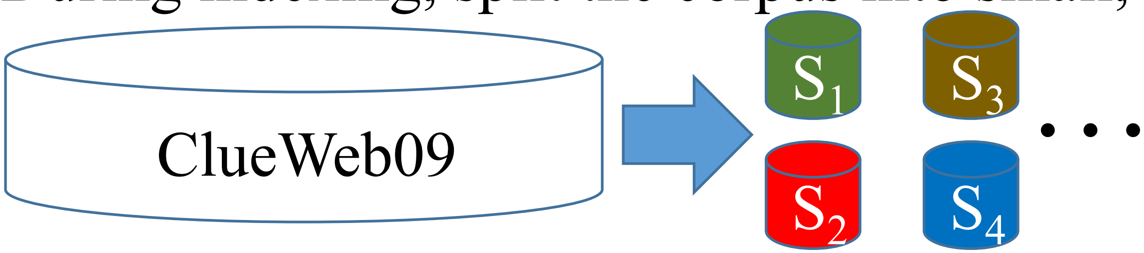
- During indexing, split the corpus into small, topical index shards



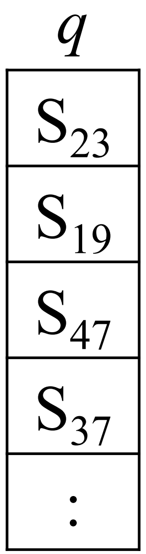
Introduction: Selective Search

Selective search is a recent distributed search architecture

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



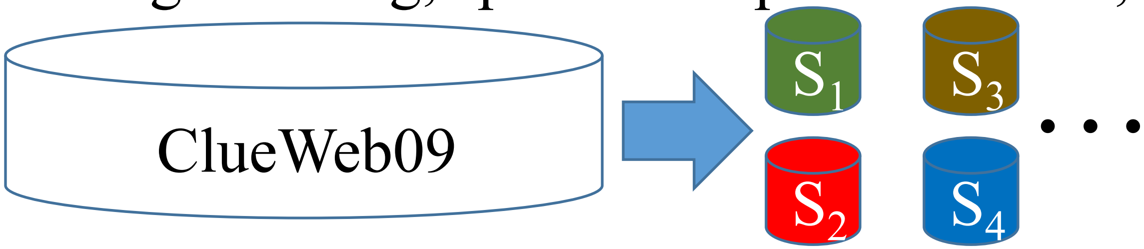
- Use resource selection to pick shards for query q
  1. Rank the index shards



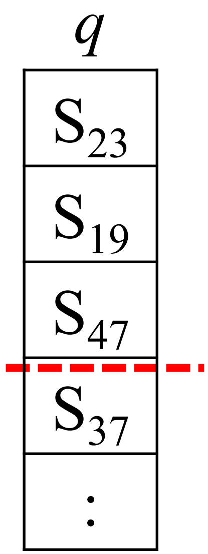
Introduction: Selective Search

Selective search is a recent distributed search architecture

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



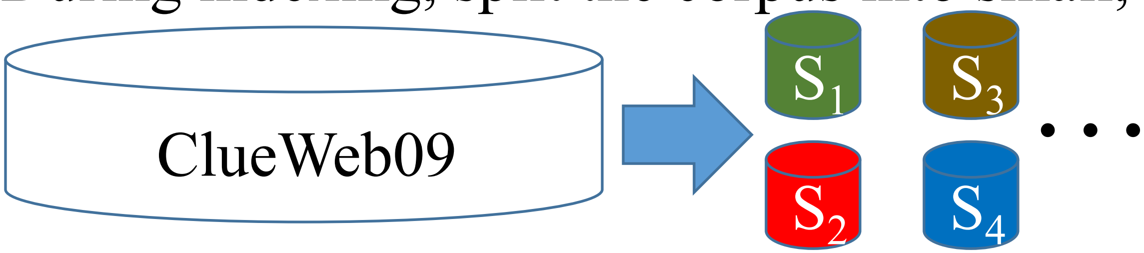
- Use resource selection to pick shards for query q
  1. Rank the index shards
  2. Decide how many shards to search



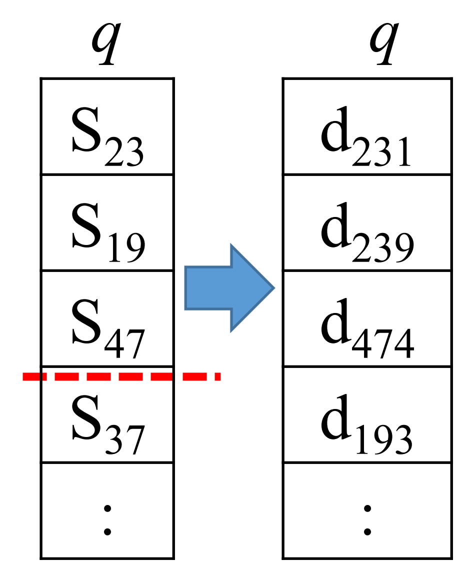
Introduction:
Selective Search

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

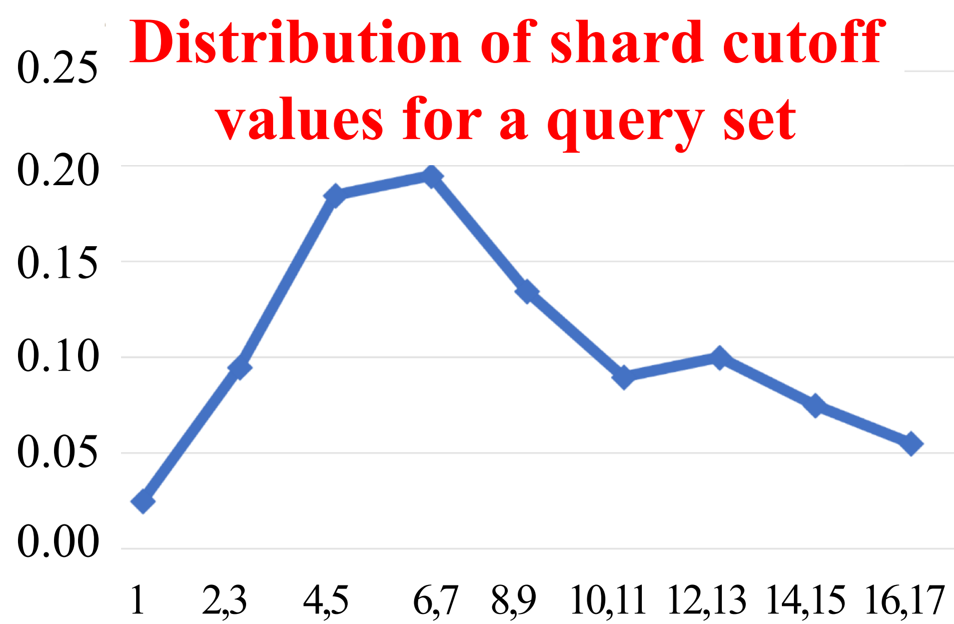
42 shard distribution features

- 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 τ
- 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 ($r_{d,e}$)

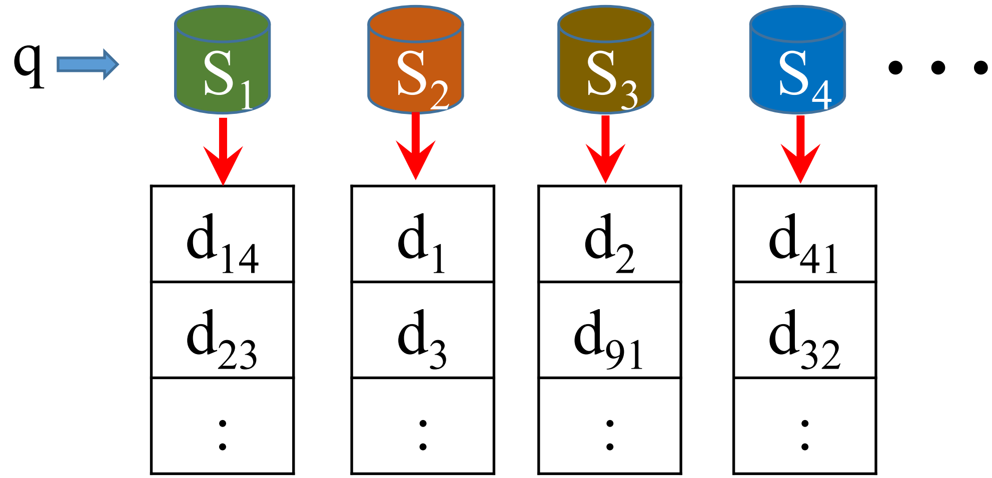
Search <u>all</u> shards

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 <u>all</u> shards



Document rankings are returned

Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

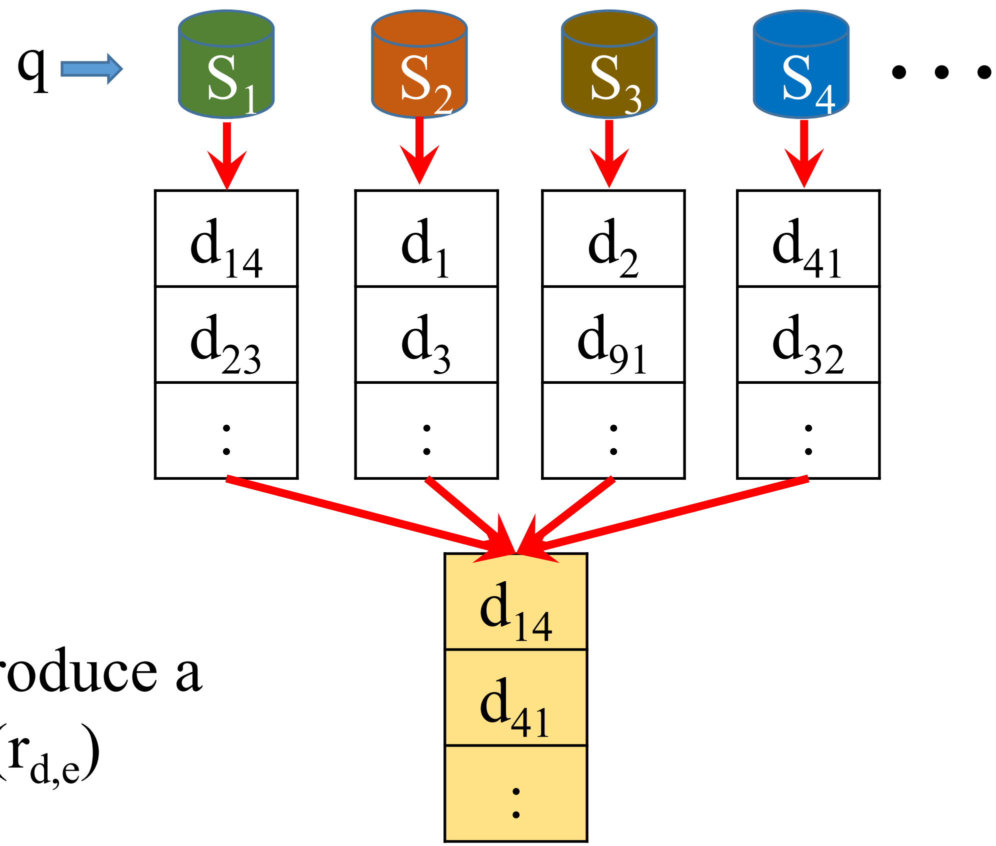
What is the 'right' number of shards k to search for query q?

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

Search <u>all</u> shards

Document rankings are returned

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

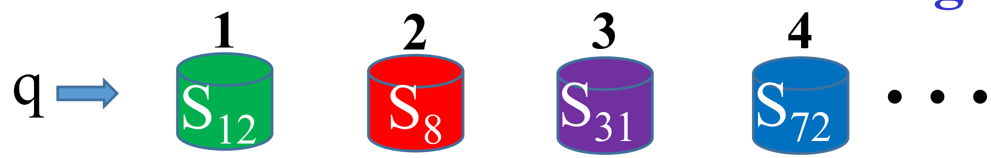


Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

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



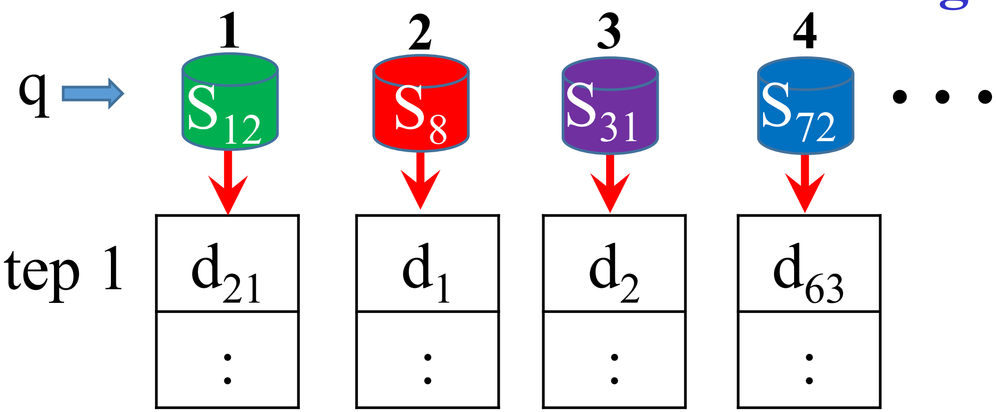
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

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



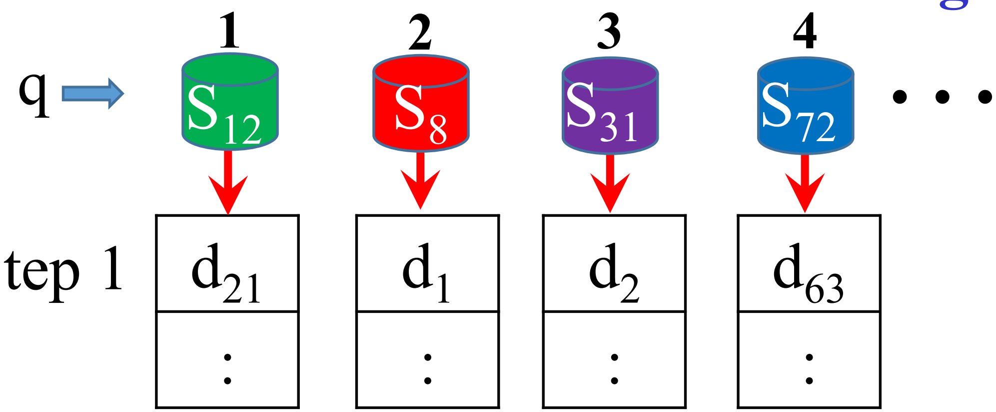
Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

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



Iterate over potential cutoffs

Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

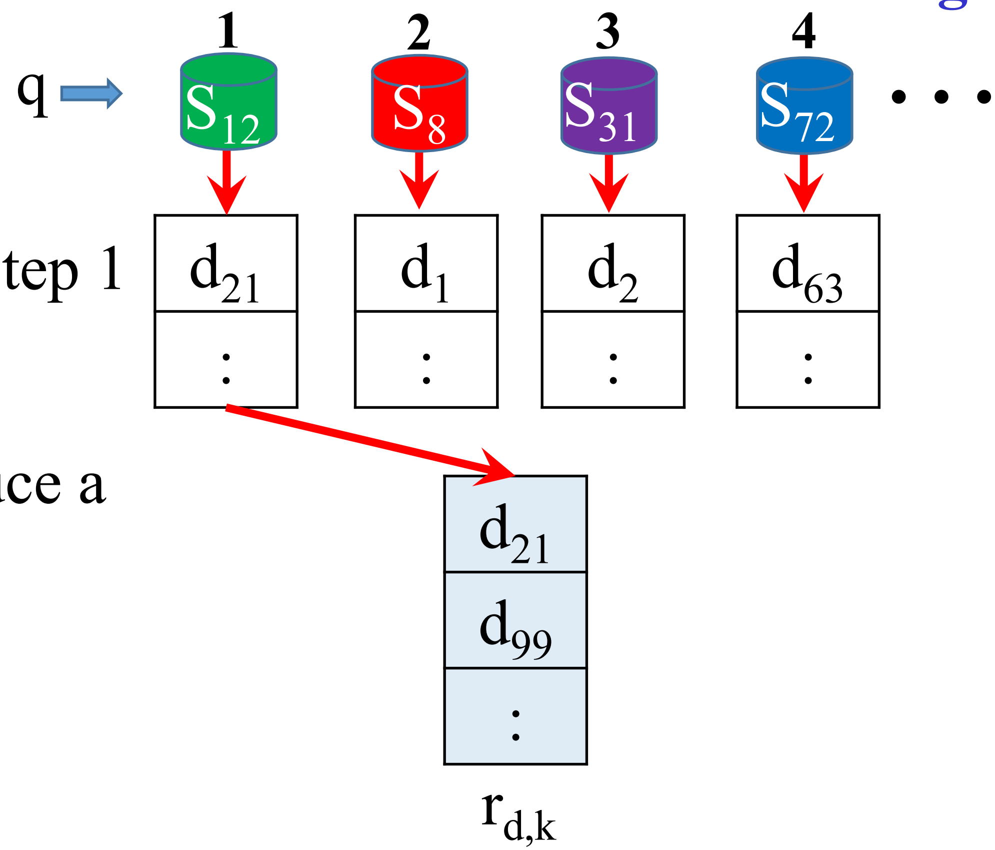
What is the 'right' number of shards k to search for query q?

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

Rank the shards

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

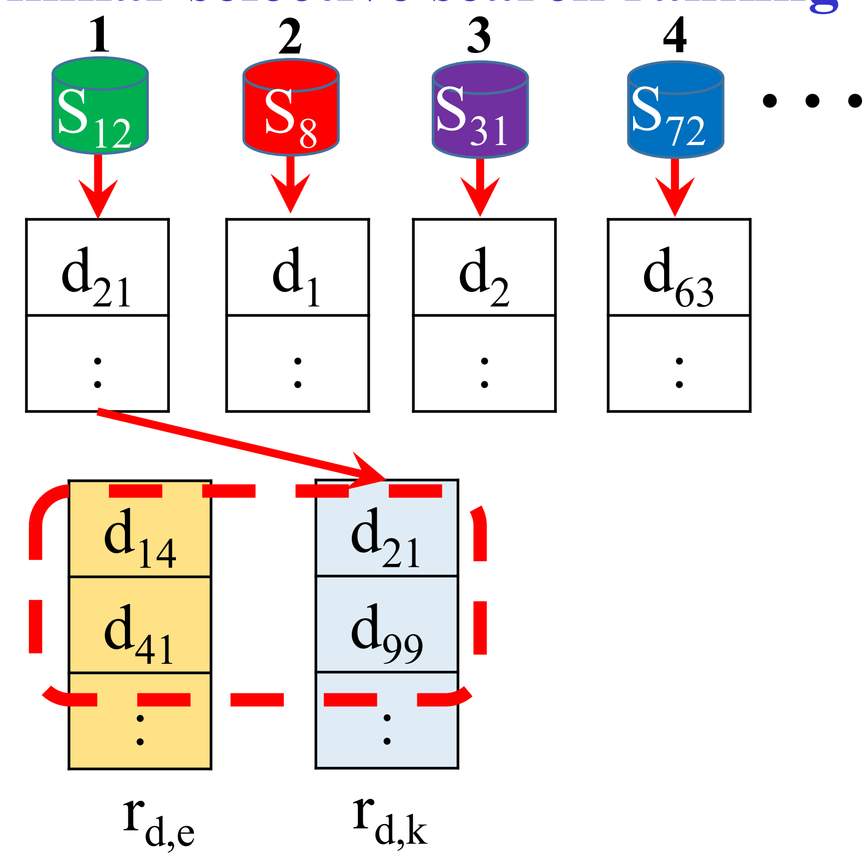
Rank the shards

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

q



Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

What is the 'right' number of shards k to search for query q?

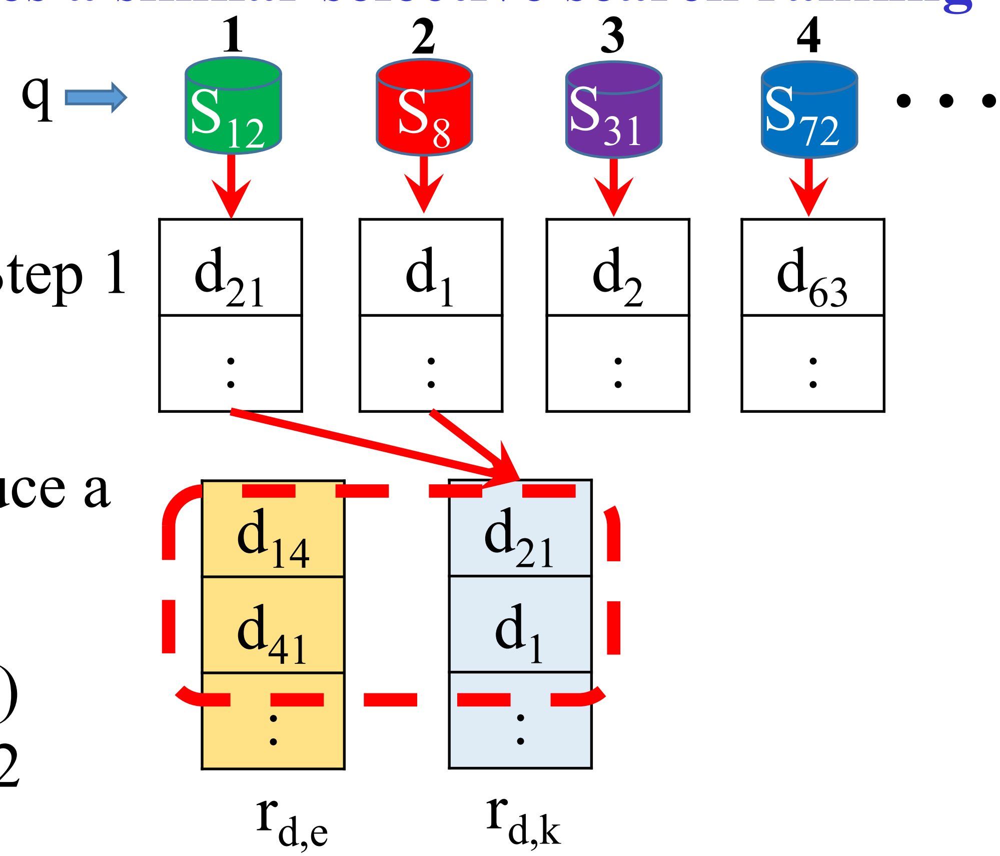
2. Find a cutoff that produces a similar selective search ranking

Rank the shards

Same document rankings as Step 1

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

If 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?

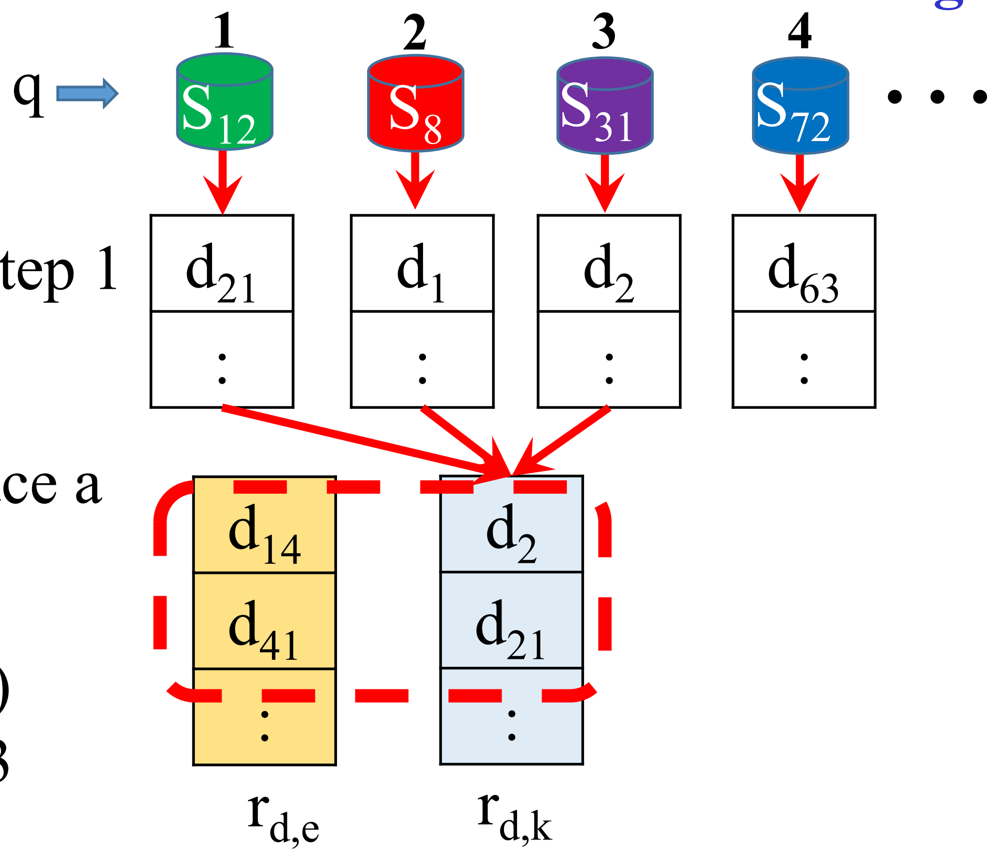
2. Find a cutoff that produces a similar selective search ranking

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)

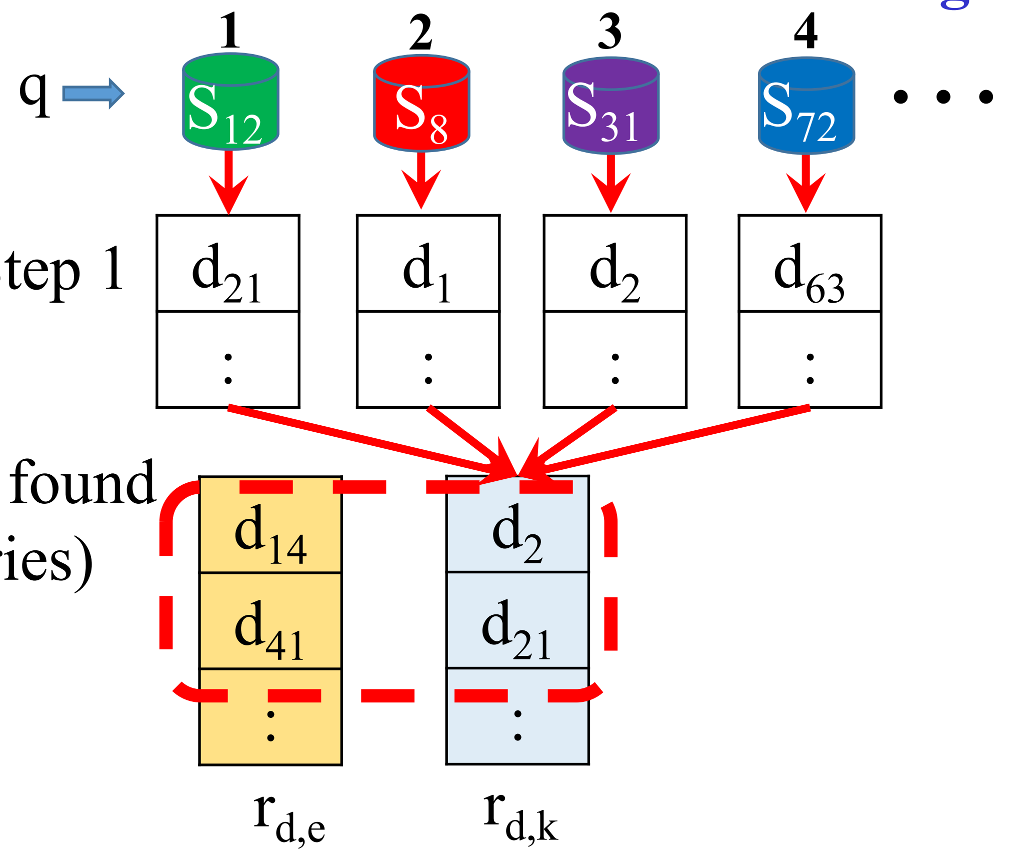
What is the 'right' number of shards k to search for query q?

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

Rank the shards

Same document rankings as Step 1

Continue until a good cutoff is found or k=16 (cap for outlier queries)

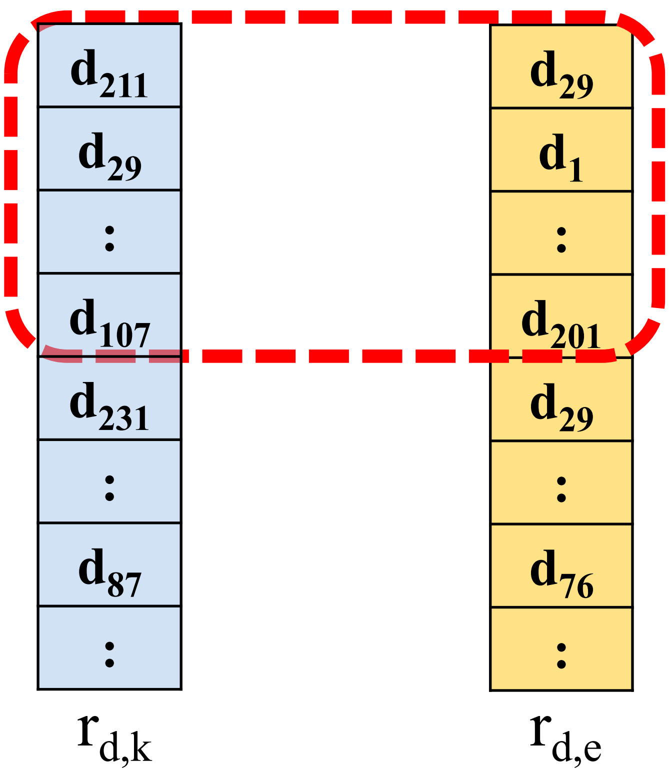


Predicting Shard Ranking Cutoffs: Training Data (Gold Standard)

Vary the definition of 'close enough' to satisfy different goals

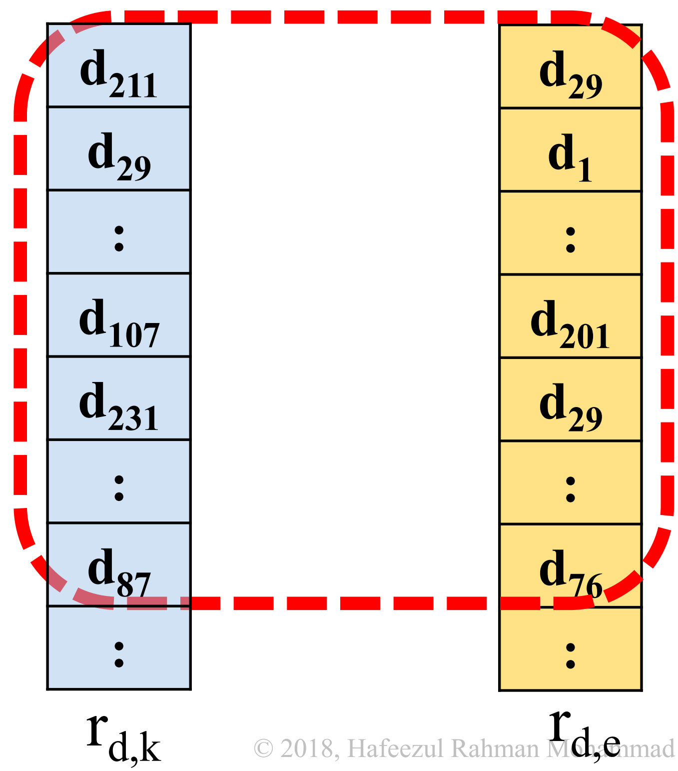
Early Precision

Overlap in top 100 documents



High Recall

Overlap in top 1,000 documents



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

	Early-Precision				High-Recall			
	Rank-S	Taily	ShRkC	QR	Rank-S	Taily	ShRkC	QR
MAE	1.31	1.34	2.99	**1.14**	2.91	2.84	4.85	**1.94**
PCC	0.37	0.34	0.26	**0.44**	0.38	0.39	0.28	**0.64**

Lower MAE & higher PCC: Better at predicting k

The Learned predictor is best under both scenarios

Experiment 1: Cutoff Prediction Comparisons

RQ3: Are ranker-independent cutoff predictions effective?

ClueWeb09-B

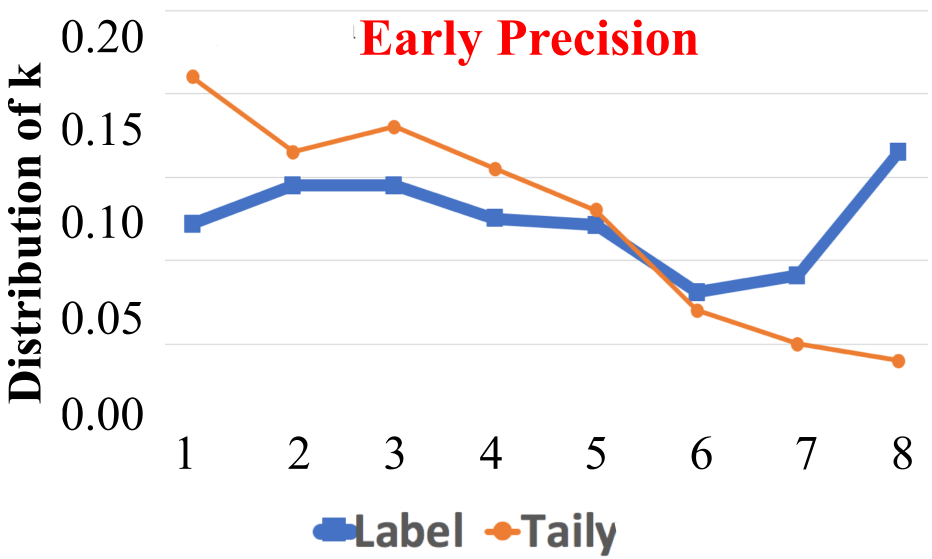
	Early-Precision				High-Recall			
	Rank-S	Taily	ShRkC	QR	Rank-S	Taily	ShRkC	QR
MAE	1.31	1.34	2.99	**1.14**	2.91	2.84	4.85	**1.94**
PCC	0.37	0.34	0.26	**0.44**	0.38	0.39	0.28	**0.64**

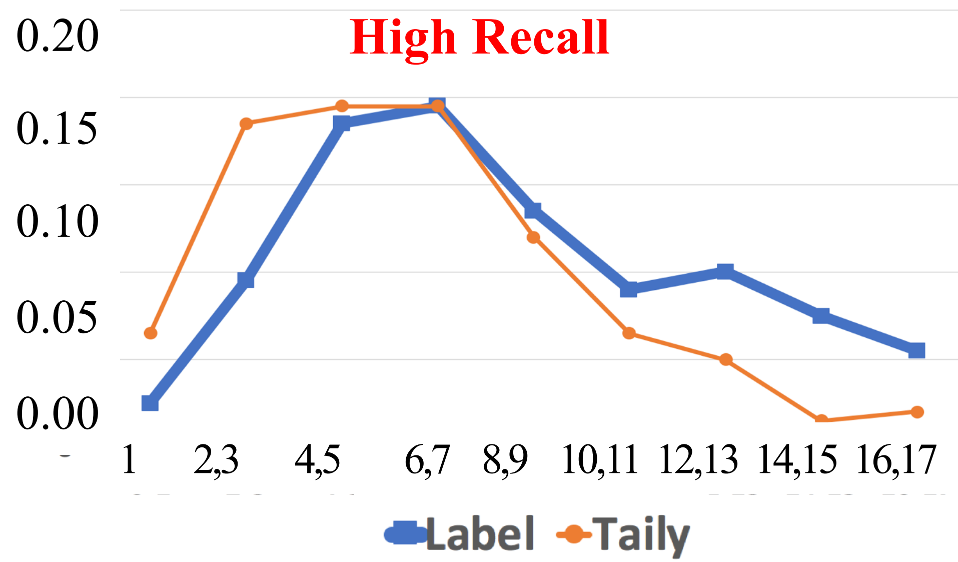
Lower MAE & higher PCC: Better at predicting k

Ranker-independent cutoff predictions can be effective

- QR is, but ShRkC is not

Experiment 1: Cutoff Prediction Comparisons

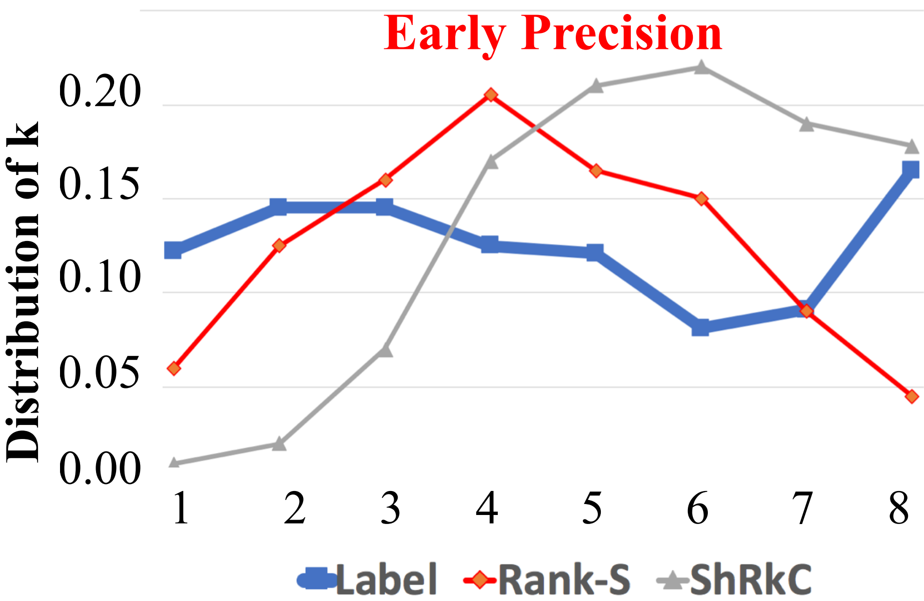


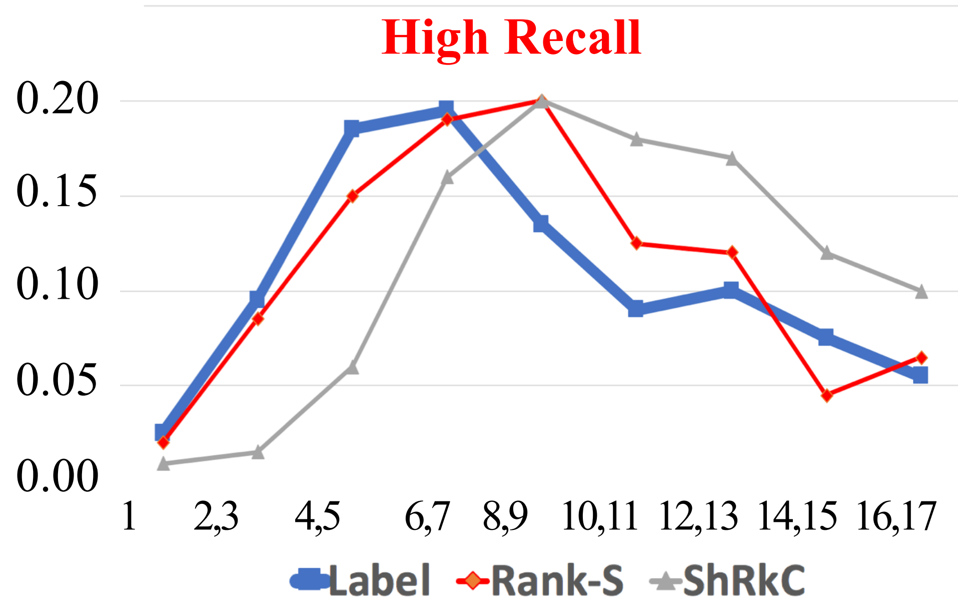


Shard cutoff biases

- Closer to the 'Label' curve is desired
- Taily tends to under predict

Experiment 1:
Cutoff Prediction Comparisons

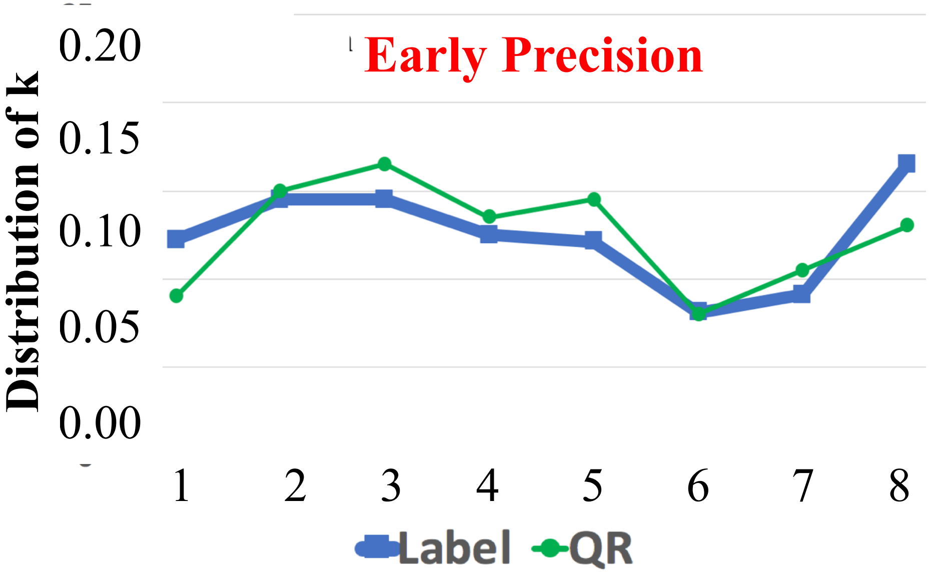


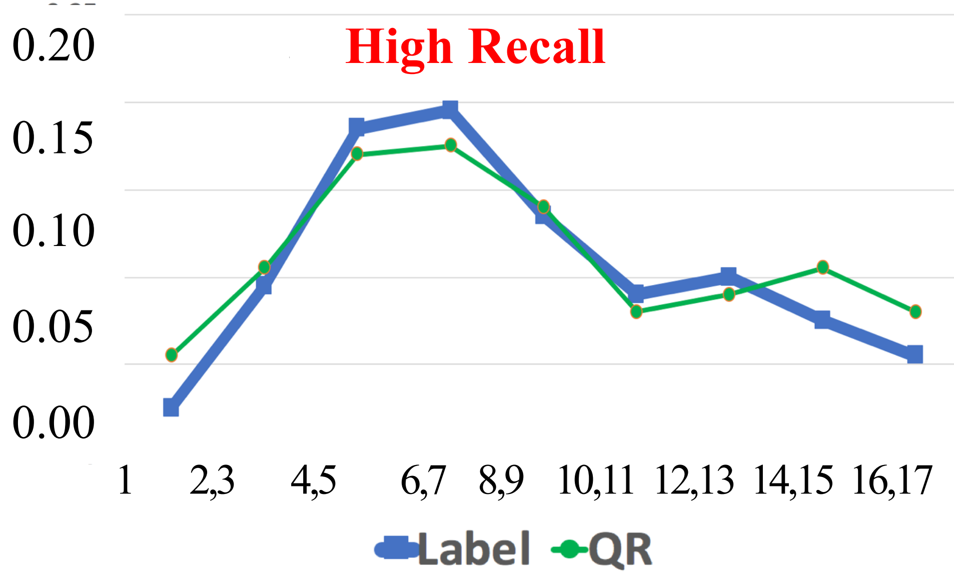


Shard cutoff biases

- 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

	High-Recall Oriented				
	Accuracy			Efficiency	
Ranking	MAP	RBP,0.95	O@5000	C_{RES}	C_{LAT}
Taily	.180	.261 (.339)	.599	.811	.187
Rank-S	.181	.279 (.349)	.612	.811	.190
ReDDE	.182	.281 (.345)	.618	.853	.198
L2RR	.196	.293 (.304)	.626	.896	.199
$r_{s,e}$	.202	.301 (.286)	.709	.850	.195
Exhaustive	.202	.292 (.309)	-	5.24	.330

Smaller shards

Larger shards

- L2RR is the most accurate shard ranker
- Rankers tend to select smaller (Taily) or larger (L2RR) shards
  - All rankers searched the same <u>number</u> of shards

Experiment 2: Shard Ranking Comparisons

	Early-Precision Oriented				
	Accuracy			Efficiency	
Ranking	P@5	NDCG@10	O@100	C_{RES}	C_{LAT}
Taily	.370	.214	.623	.508	.180
Rank-S	.375	.229	.673	.517	.178
ReDDE	.386	.229	.708	.551	.190
L2RR	.389	.234	.734	.560	.189
$r_{s,e}$	.409	.247	.818	.534	.187
Exhaustive	.390	.240	-	5.24	.330

Smaller shards (Taily, Rank-S)

Larger shards (ReDDE, L2RR, $r_{s,e}$)

- L2RR is the most accurate shard ranker
- Rankers tend to select smaller (Taily) or larger (L2RR) shards
  - All rankers searched the same <u>number</u> 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

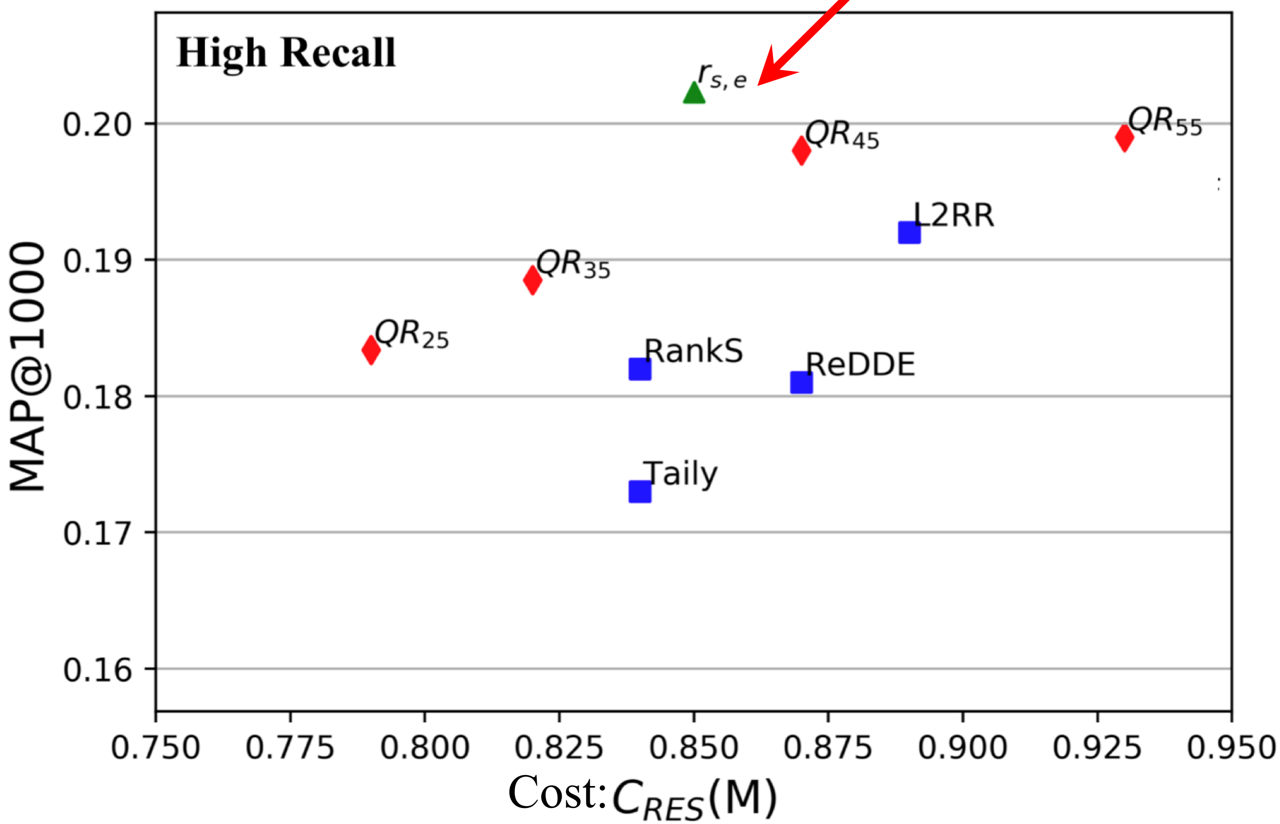
Left is more efficient

Goal is to be close to $r_{s,e}$

QR's τ enables tuning efficiency vs effectiveness tradeoff

- $\tau = 0.45$ works well

Optimal result



Experiment 3: Precision vs Recall

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

Up is more accurate

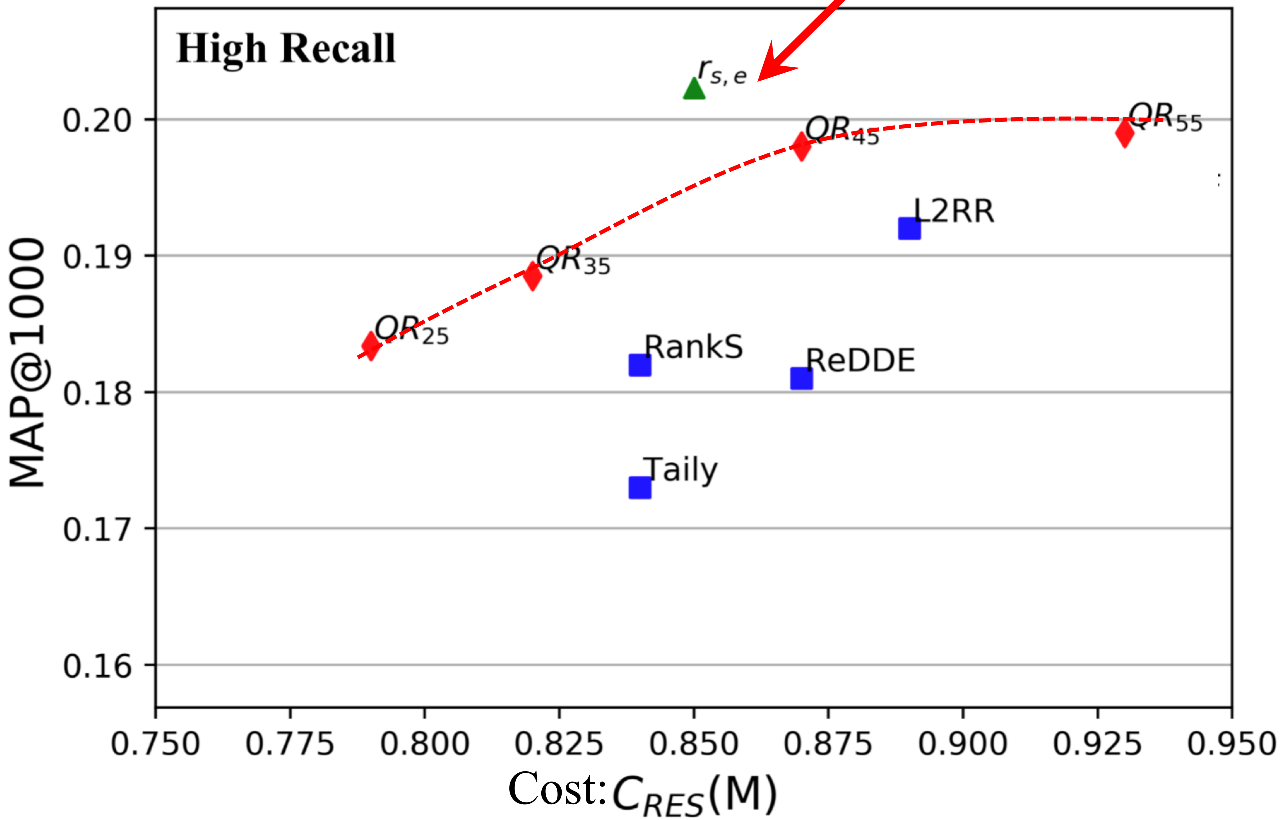
Left is more efficient

Goal is to be close to $r_{s,e}$

QR's τ enables tuning efficiency vs effectiveness tradeoff

- $\tau = 0.45$ works well

Optimal result



Experiment 3: Precision vs Recall

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

Optimal result

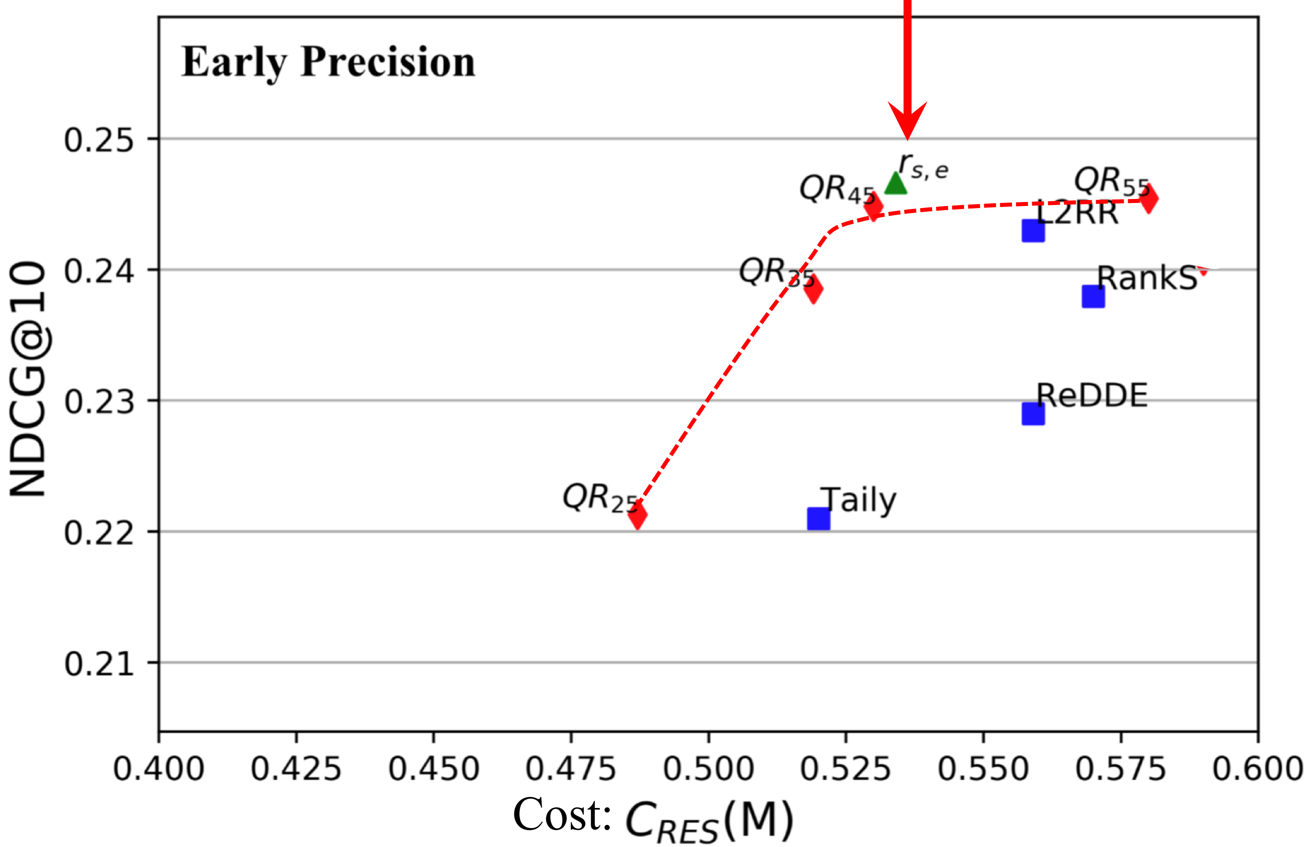
Up is more accurate

Left is more efficient

Goal is to be close to $r_{s,e}$

QR's τ 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?