Selective Search and WAND

Yubin Kim, Jamie Callan
J. Shane Culpepper, Alistair Moffat

Carnegie Mellon University

RMIT UNIVERSITY

THE UNIVERSITY OF MELBOURNE

Background: Selective search

- Traditional distributed search
  1. Randomly divide large collection into small “shards”
  2. Assign shards into multiple machines
  3. Search all shards in parallel
- Selective search
  1. Cluster large collection into small topical “shards”
  2. Assign shards into multiple machines
  3. Use resource selection to decide which shards to search
  4. Search selected shards in parallel
- For each query, only a few shards are searched
- Can result in up to 90% savings in computing cost

Background: WAND

- Dynamic pruning of postings list to reduce scoring operations
- Each postings list stores a maximum score indicating the upper bound
- Keeps track of a minimum score threshold required for a document to appear in the top-k, updated as documents are scored
- If a document cannot exceed the minimum score threshold based on the max score of the postings list, early exit

Background

- Selective search and WAND both reduce costs by avoiding scoring large chunks of the index

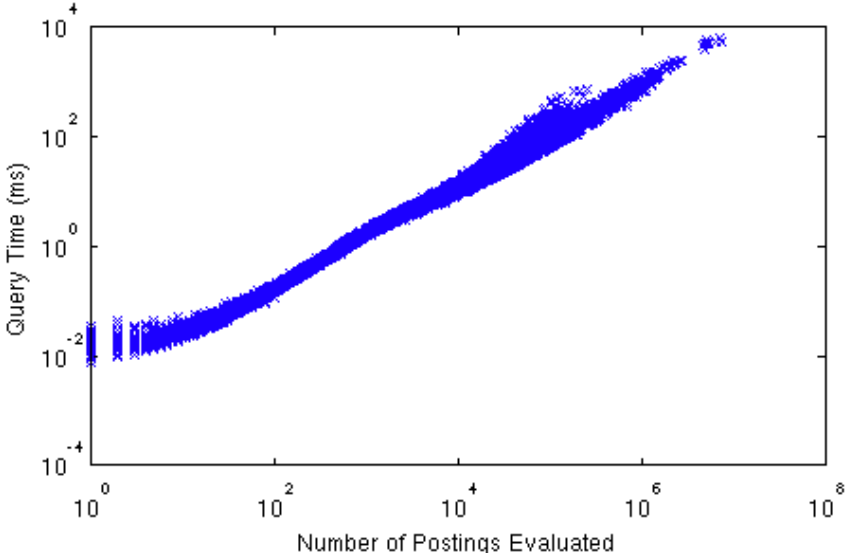
Do selective search and WAND do the same thing?

- Do they skip the same areas of the index?
- Are there additive gains?

Experimental method: Cost metric

- Number of postings evaluated
- This has a high correlation with actual query processing time
- Used to calculate w/b

  w = # postings evaluated by WAND
  b = total # postings



- Micro-averaging: w and b are summed over queries and then ratio calculated
- Macro-averaging: w/b calculated for each query and averaged across queries

Lower = more savings = better!

Experimental method

- First 1000 unique queries from AOL query log and TREC Million Query track
  - Single term queries removed – WAND doesn't affect 1 term queries
  - Final total of 713 from AOL and 756 from Million Query Track
- ClueWeb09 Category B
  - 50 million web documents
  - Divided into 100 shards
- Index scored using BM25

Experiment 1: Random vs Topic shards

- Topic shard vs Random shard
  - WAND run on all shards
  - No resource selection, for now
- WAND has similar performance on both types of shards.

	Topic shards	Random shards
AOL micro-avg	0.35	0.34
MQT micro-avg	0.36	0.36
AOL macro-avg	0.51	0.52
MQT macro-avg	0.60	0.63

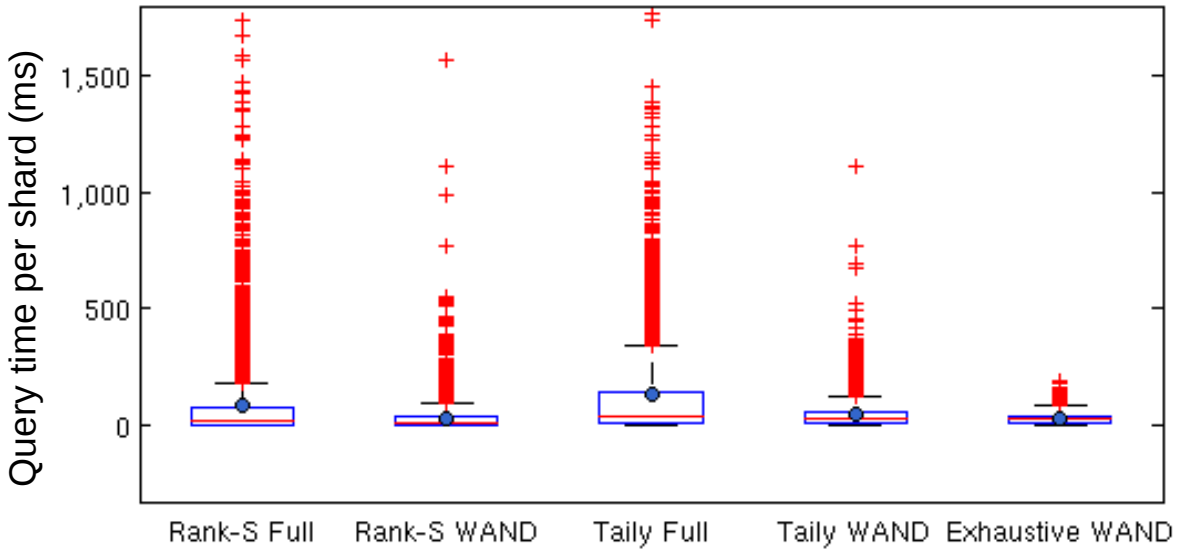
Experiment 2: Selected vs Non-selected Topic Shards

- Selected topic shards vs Non-selected topic shards
- WAND is best with shards dense in relevant docs, which is exactly what resource selection delivers
- Selected shards see greater improvement from WAND, better-than-additive savings

	Selected	Non-selected
Taily AOL	0.32	0.35
Taily MQT	0.23	0.37
Rank-S AOL	0.27	0.36
Rank-S MQT	0.24	0.37

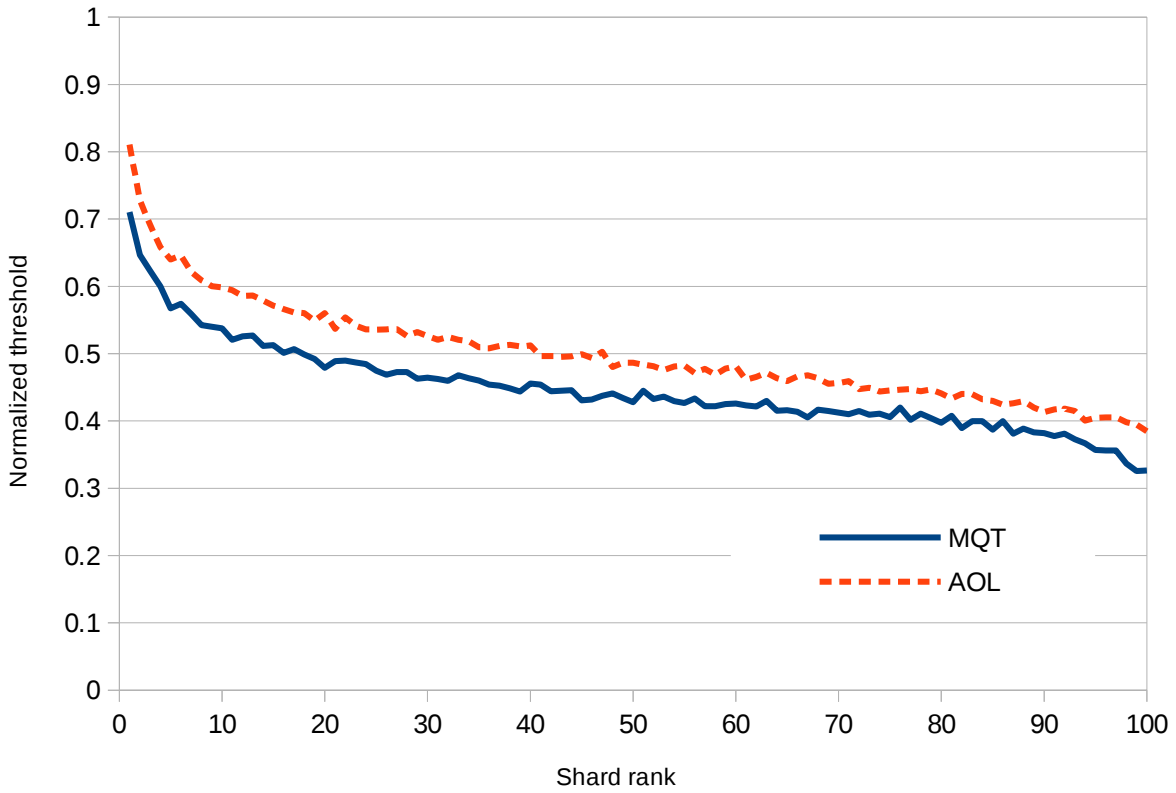
Experiment 3: Distribution of query response times

- WAND reduces the variance of query costs; affects the slowest shards most
  - Compare e.g. Rank-S Full vs. Rank-S WAND
- Note: these are per shard costs
  - Selective search: 3~5 shards
  - Exhaustive search searches 100



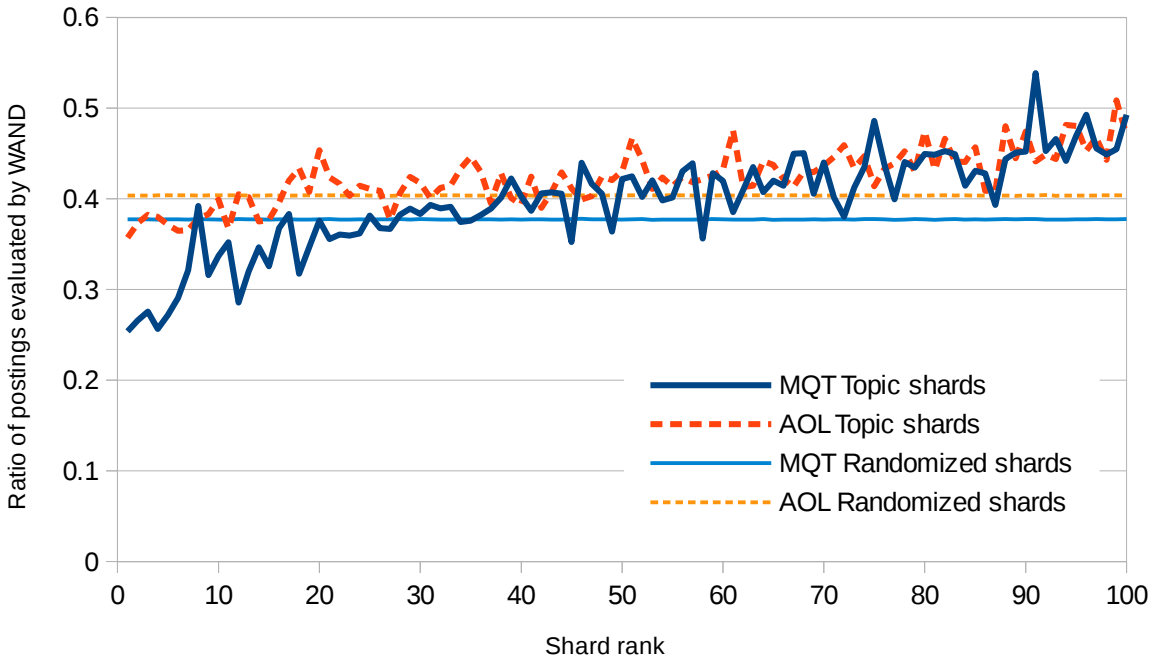
Experiment 4: Why does it work?

- Graph of the final minimum score thresholds (i.e. the score of the 1000th document) for shards in order ranked by Taily
- High-ranking shards have higher scoring documents



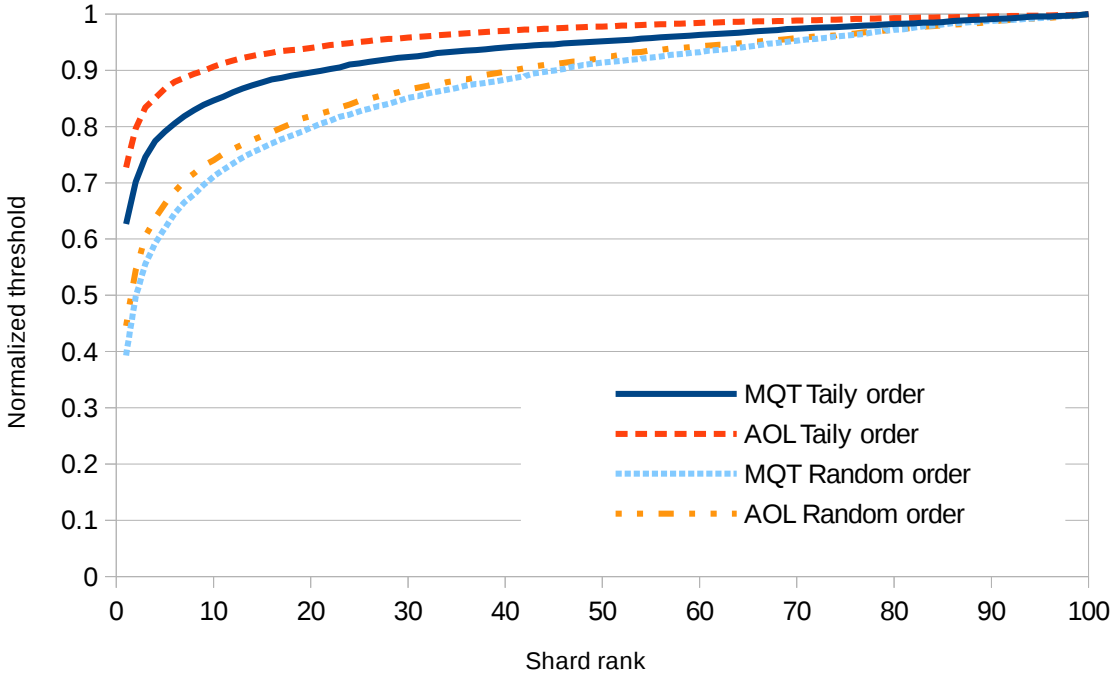
Experiment 4: Why does it work?

- w/b calculated for each shard in order of Taily scores
- Higher-ranking shards benefit more from WAND
- Good resource selection can improve efficiency as well as effectiveness



Experiment 5: Sharing Minimum Score Thresholds

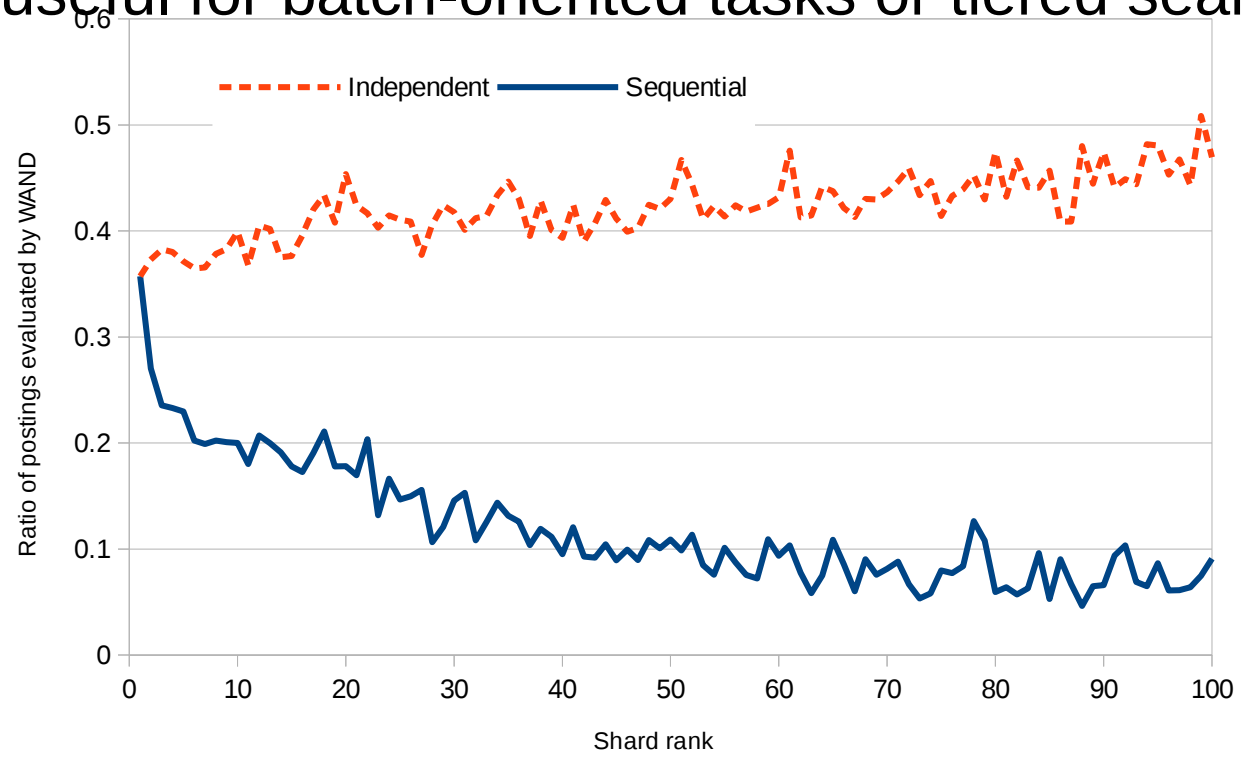
- So far, experiments were conducted assuming each shard is processed independently; the score thresholds were not shared



- Alternatively, the shards can be searched in order of ranking and the minimum score thresholds preserved and passed on
- Higher starting thresholds should generate additional savings

Experiment 5: Sharing Minimum Score Thresholds

- w/b comparing independent shard search and sequential shard search with shared thresholds
- Shared thresholds require far less total computation – at the cost of latency
  - Possibly useful for batch-oriented tasks or tiered search



Conclusions

- Selective search and WAND produce better-than-additive gains!
- WAND produces greater savings on selected shards than random or non-selected shards
- Resource selection identifies the shards where WAND optimization will be most effective
- By passing the thresholds in a sequential shard search, can significantly reduce total costs at the cost of latency
  - A hybrid between two approaches? e.g. tiered seach

Questions?

Yubin Kim, Jamie Callan
J. Shane Culpepper, Alistair Moffat

Carnegie Mellon University

RMIT UNIVERSITY

THE UNIVERSITY OF MELBOURNE