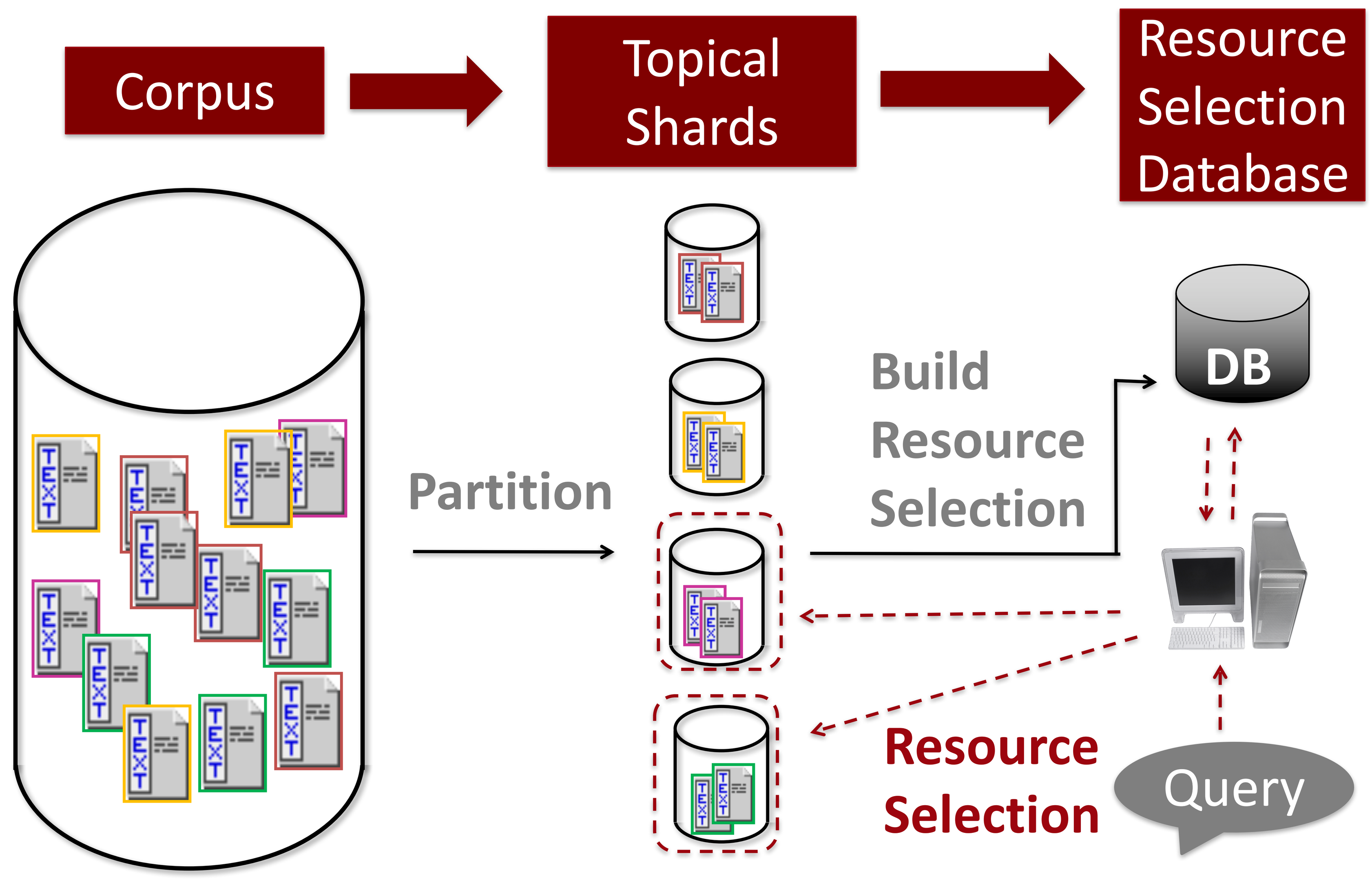
Learning To Rank Resources

Zhuyun Dai, Yubin Kim, Jamie Callan
Language Technologies Institute, Carnegie Mellon University

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

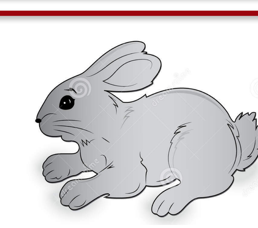
2. Overlap-based

- The number of documents in a shard that were ranked highly by exhaustive search
- No manual judgement required
- Can be automatically generated

Features

FAST

1. Query-Independent Information



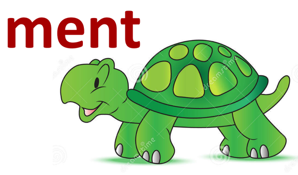
- **Shard Popularity**

2. Term Based Statistics

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

SLOW

3. Sample-Document Features



- **Ranks and ReDDE**: score, inverse rank, binned rank
- **Average Distance to Shard Centroid:** the distance between the top-k documents retrieved from the CSI to their respective shards' centroids

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

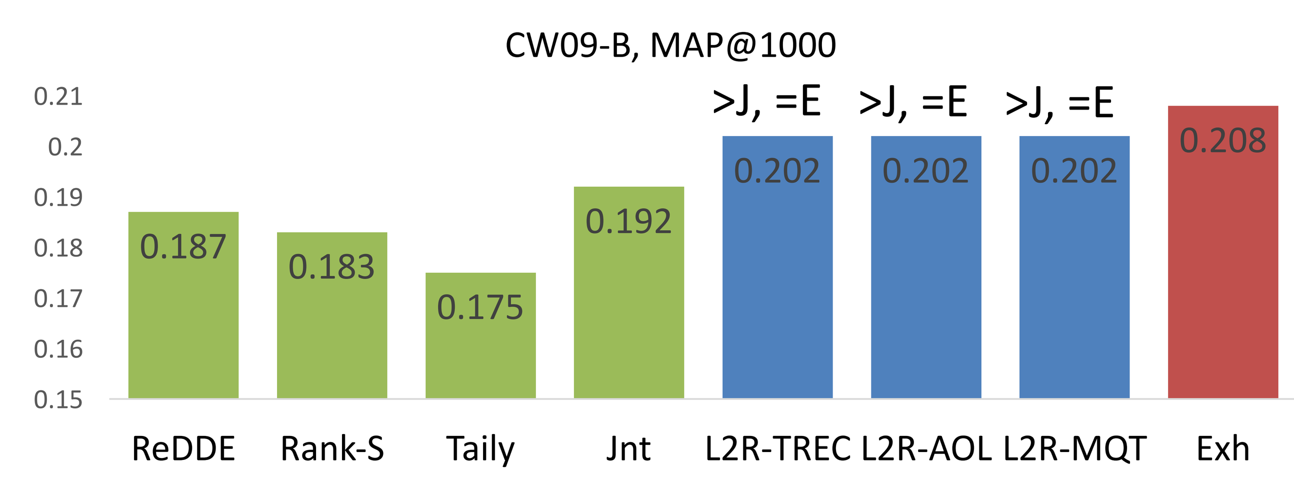
Baselines:

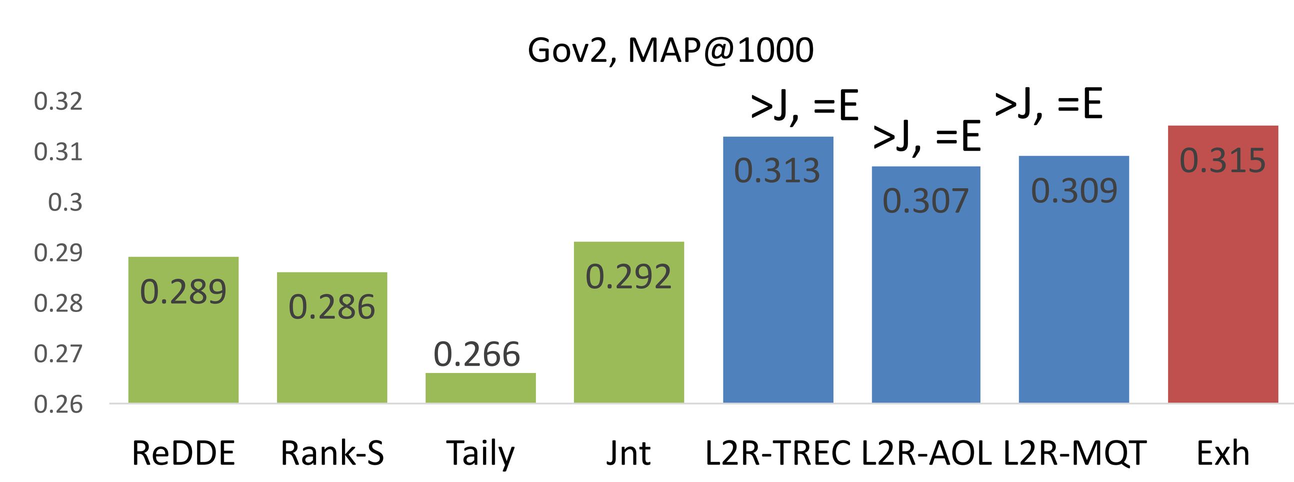
- Term-based: Taily
- Sample-based: ReDDE, Rank-S
- Supervised: Jnt

Model

- RankSVM
- Linear kernel

Exhaustive Search (Exh): Searching all shards





>J: Statistically better than Jnt baseline
=E: Statistically equivalent to Exh

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

FAST v.s. SLOW

	Method	P @10	NDCG @30	MAP @1000	Average Cost
Cw09 -B	Redde	0.363*	0.275*	0.187	156,180
	Taily	0.346	0.260	0.175	470
	Jnt	0.367*	0.277*	0.192	468,710
	ALL	0.375*	0.286*	0.202*	158,529
	FAST	0.373*	0.285*	0.201*	2,349
Gov2	Redde	0.579*	0.445*	0.289	105,080
	Taily	0.518	0.403	0.256	758
	Jnt	0.588*	0.465*	0.292	315,875
	ALL	0.593*	0.474*	0.309*	108,306
	FAST	0.587*	0.471*	0.310*	3,226

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

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