

# LambdaMerge: Merging the Results of Query Reformulations

Daniel Sheldon  
Oregon State University  
sheldon@eecs.oregonstate.edu

Milad Shokouhi, Martin Szummer, and Nick Craswell  
Microsoft Research  
{milads,szummer,nickcr}@microsoft.com

## Merging Query Reformulations

### Query reformulation algorithms

- Improve retrieval by alleviating Q-D mismatch  
**bill gates bio** → **bill gates biography**
- Or can make things worse e.g. **bio** → **biog**
- Or even drift off-topic e.g. **bill** → **melinda**

### Approach: Run multiple queries, merge results

- Post-retrieval it is easier to detect quality/drift
- Multiple queries give diverse relevance evidence

### Contribution: New merging methods

1. CombRW: Weighted CombSUM (unsupervised)
2. LambdaMerge: Supervised merging
  - Trained to maximize target such as NDCG
  - Incorporating quality and drift features
  - Robust to bad reformulations

## Experiments

**Test collections:** Bing data and GOV2

**Reformulations:** Click graph random walk [2]

### Single-query methods:

- ORG: Original query
- RW1: Most likely alternative query from RW
- RAPP-L: Predict best query (lin. regression) [3]
- RAPP( $\Omega$ ): Choose query via NDCG@5 (oracle)

### Merging methods:

- CombSUM: Sum scores
- CombRW: CombSUM with random walk weight
- LambdaMerge: Using these features

Table 1: List of features.

| Query-document features      | Score, Rank, NormScore <sub>[0,1]</sub> , NormScore <sub>N(0,1)</sub> , IsTopN |
|------------------------------|--|
| Gating features (difficulty) | ListMean, ListStd, ListSkew, Clarity, RewriteLen, RAPP                         |
| Gating features (drift)      | IsRewrite, RewriteRank, RewriteScore, Overlap@N                                |

### Results:

|                  | NDCG@5 | NDCG@10 |
|------------------|--------|---------|
| ORG              | 0.538  | 0.524   |
| RW1              | 0.422  | 0.387   |
| CombSUM          | 0.510  | 0.486   |
| CombRW           | 0.542  | 0.516   |
| RAPP-L           | 0.534  | 0.524   |
| $\lambda$ -Merge | 0.555  | 0.539   |
| RAPP( $\Omega$ ) | 0.556  | 0.530   |

Bing

|                  | P@5   | NDCG@100 |
|------------------|-------|----------|
| ORG              | 0.548 | 0.431    |
| CombRW           | 0.584 | 0.438    |
| $\lambda$ -Merge | 0.596 | 0.447    |
| RAPP( $\Omega$ ) | 0.592 | 0.457    |

GOV2

## LambdaMerge

### Issue $k$ formulations to search engine:

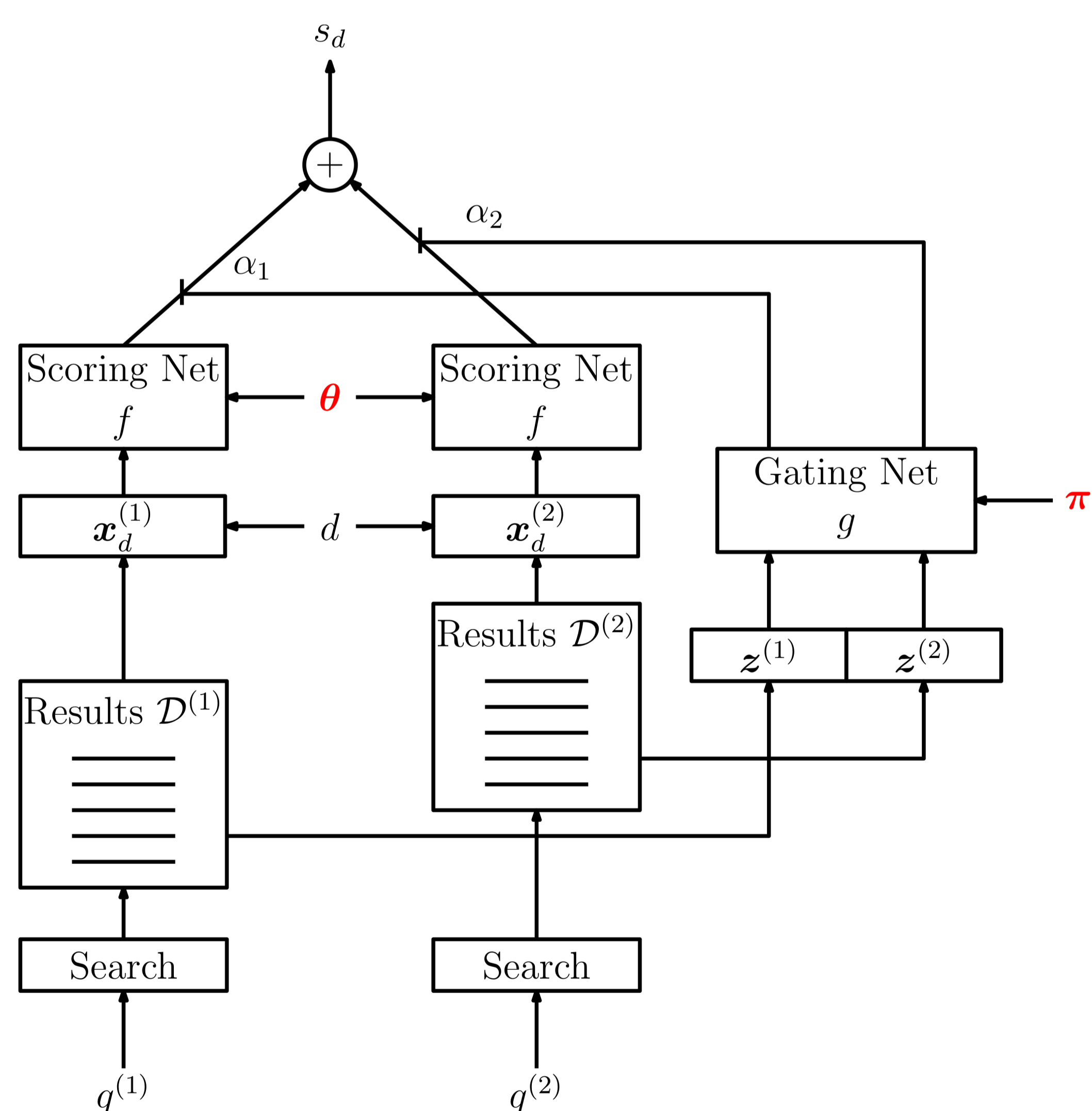
- Original query  $q^{(1)}$  plus reformulations  $q^{(2)}, \dots, q^{(k)}$
- Get top- $N$  lists  $D^{(1)}, \dots, D^{(k)}$

### Generate features:

- Query-document features  $\mathbf{x}_d^{(k)}$ : relevance of document  $d$  specific to  $D^{(k)}$
- Gating features  $\mathbf{z}^{(k)}$ : drift + overall quality of  $D^{(k)}$

**Scoring net** assigns score  $f(\mathbf{x}_d^{(k)}; \theta)$  to each formulation-document pair

**Gating net** assigns weights  $\alpha_1, \dots, \alpha_k$  to formulations

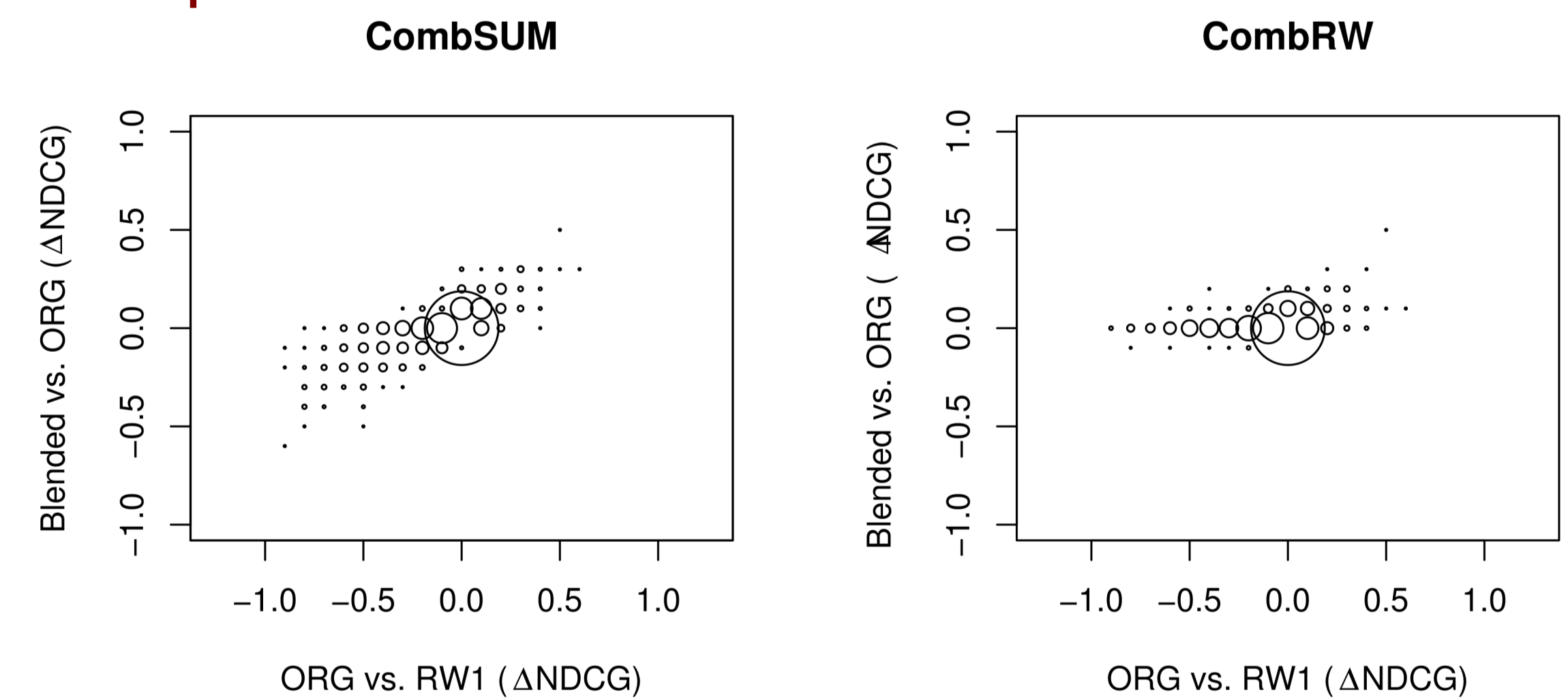


$$\text{Overall document score: } s_d = \sum_k \alpha_k \cdot f(\mathbf{x}_d^{(k)}; \theta)$$

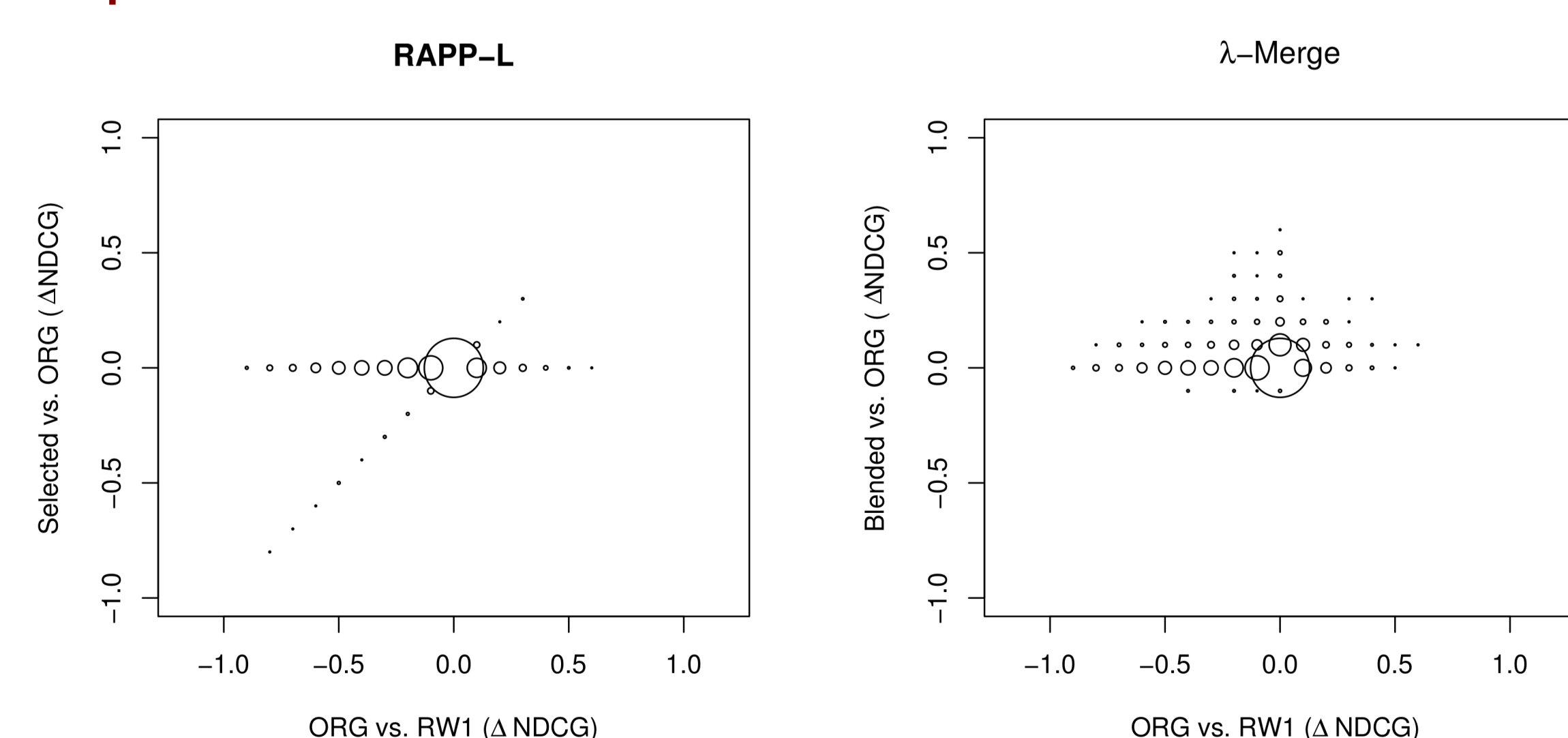
Parameters  $\theta$  (scoring) and  $\pi$  (gating) trained by backprop with **LambdaRank** gradients to optimize **NDCG** [1]

## Robustness Analysis

### Unsupervised methods:



### Supervised methods:



[1] Burges, Ragno, and Le. **Learning to rank with nonsmooth cost functions**. NIPS 2006

[2] Craswell and Szummer. **Random Walks on the Click Graph**. SIGIR 2007

[3] Balasubramanian, Kumaran, and Carvalho. **Predicting query performance on the web**. SIGIR 2010 poster