Sparse Pairwise Re-ranking with Pre-trained Transformers

ICTIR 2022




Lukas Gienapp[1] **Maik Fröbe**[2] Matthias Hagen[2] Martin Potthast[1]

[1] UNIVERSITÄT LEIPZIG

[2] MARTIN-LUTHER-UNIVERSITÄT HALLE-WITTENBERG

Problem Description

Pairwise ranking models are slow.

Problem Description

Pairwise ranking models are slow.

Can we make them faster?

Background

Evolution of feature-based learning to rank models

- Pointwise LTR $\Rightarrow$ Pairwise LTR $\Rightarrow$ Listwise LTR

From pointwise to pairwise transformers [Nogueira et. al 2020, Pradeep et. al 2021]:

- Pointwise retrieval with monoT5:
  Input: Query q, Document d
  Output: Probability that d is relevant to q

- Pairwise retrieval with duoT5:
  Input: Query q, Document d_a, Document d_b
  Output: Pairwise preference (probability that d_a is more relevant to q than d_b)

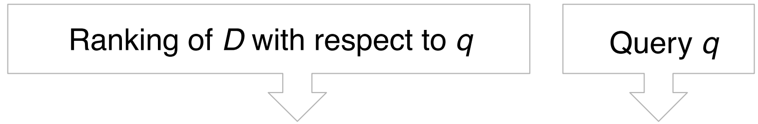
MS MARCO (Passage; DL 19/20).

Ranker	No. Inferences	nDCG@10
monoT5 (k=1000)	1000	0.50
+ duoT5 (k=50)	1000 + 2450	0.67

For k documents, duoT5 makes $k^2 - k$ pairwise comparisons.

Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview

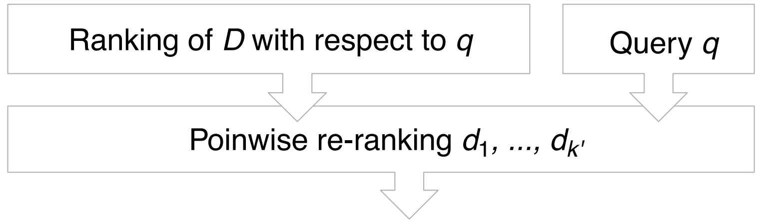


Four steps:

1. BM25 ranking (whole corpus)

Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview



Four steps:

1. BM25 ranking (whole corpus)

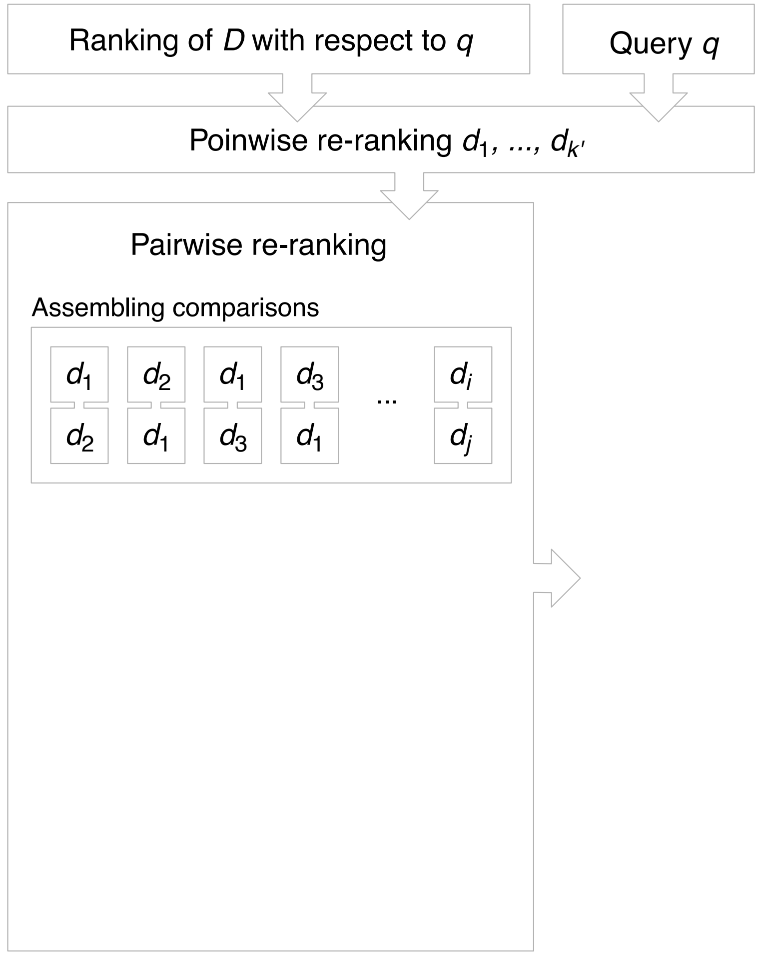
2. Pointwise re-ranking (top 1000)

Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview

Four steps:

1. BM25 ranking (whole corpus)
2. Pointwise re-ranking (top 1000)
3. Pairwise re-ranking (top 50)
   - assemble document pairs

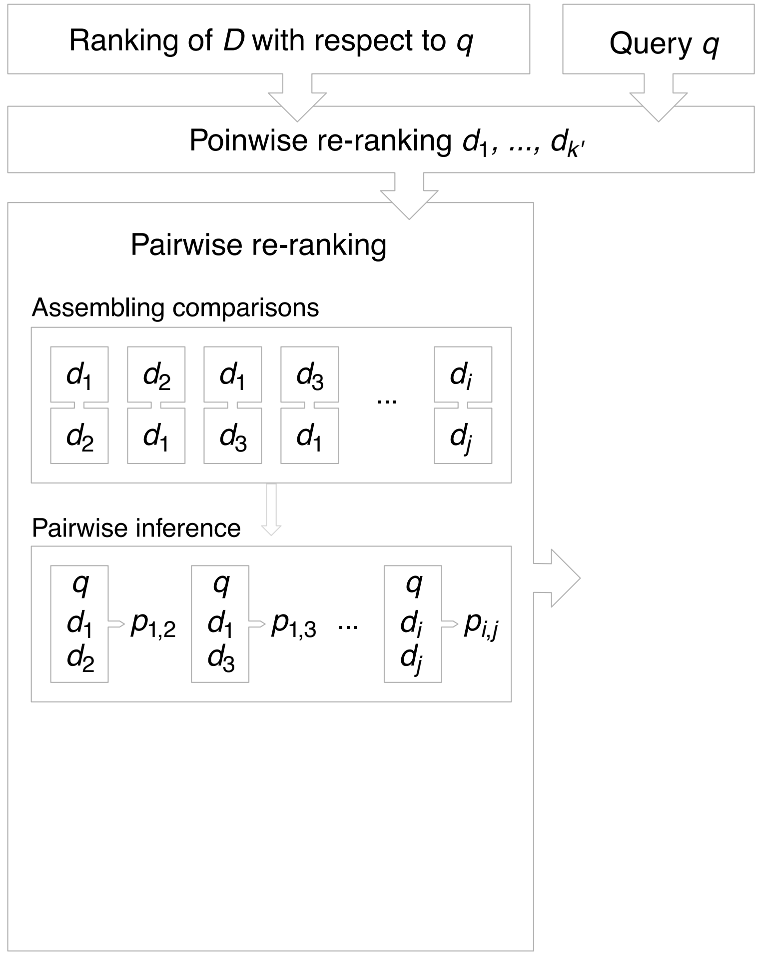


Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview

Four steps:

1. BM25 ranking (whole corpus)
2. Pointwise re-ranking (top 1000)
3. Pairwise re-ranking (top 50)
   - ❑ assemble document pairs
   - ❑ pairwise inference

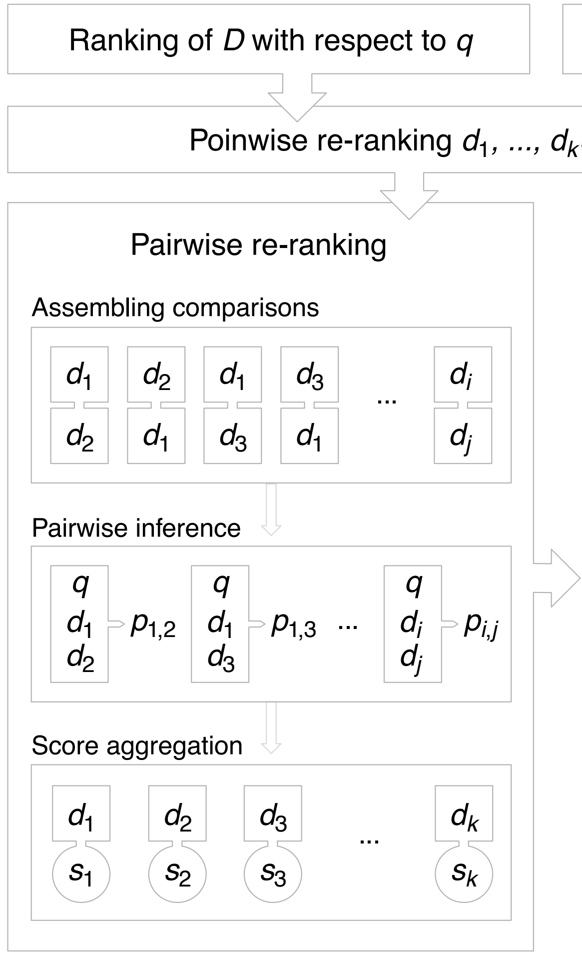


Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview

Four steps:

1. BM25 ranking (whole corpus)
2. Pointwise re-ranking (top 1000)
3. Pairwise re-ranking (top 50)
   - ❑ assemble document pairs
   - ❑ pairwise inference
   - ❑ score aggregation

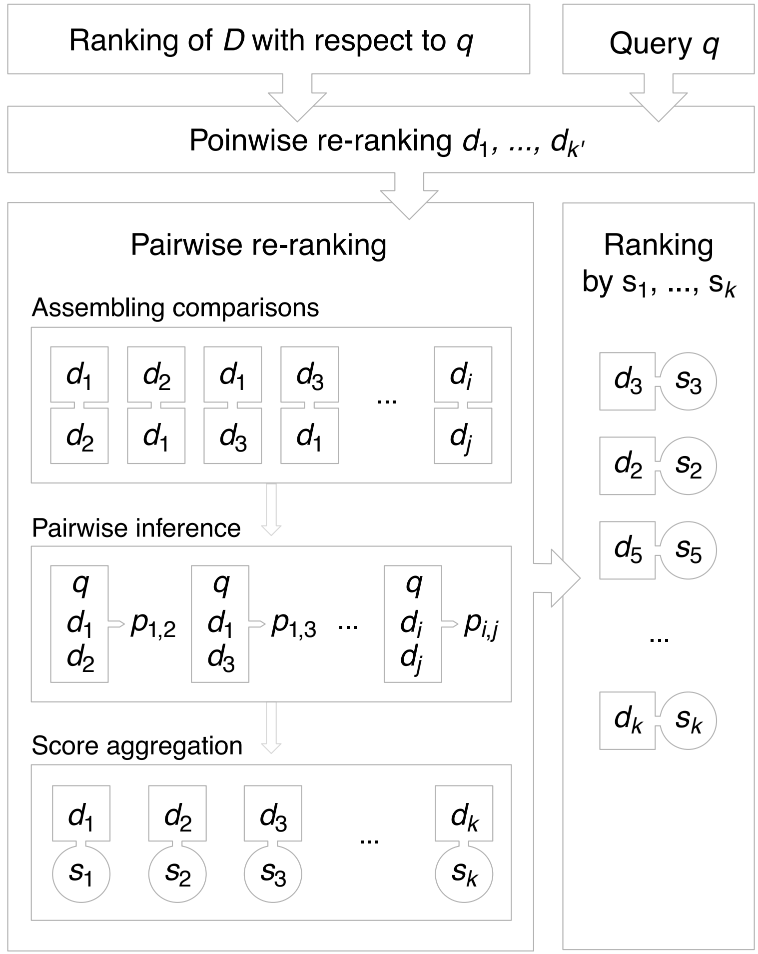


Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview

Four steps:

1. BM25 ranking (whole corpus)
2. Pointwise re-ranking (top 1000)
3. Pairwise re-ranking (top 50)
   - assemble document pairs
   - pairwise inference
   - score aggregation
4. Rank by aggregated score

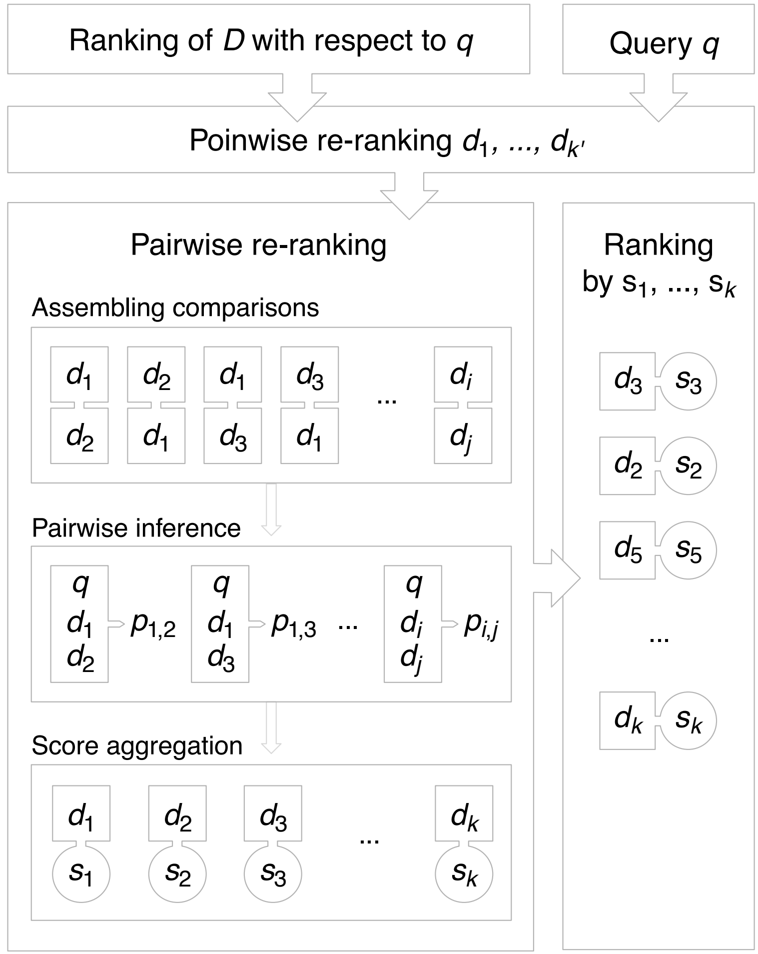


Mono-Duo Pairwise Reranking [Pradeep et. al 2021]

Pipeline Overview

Four steps:

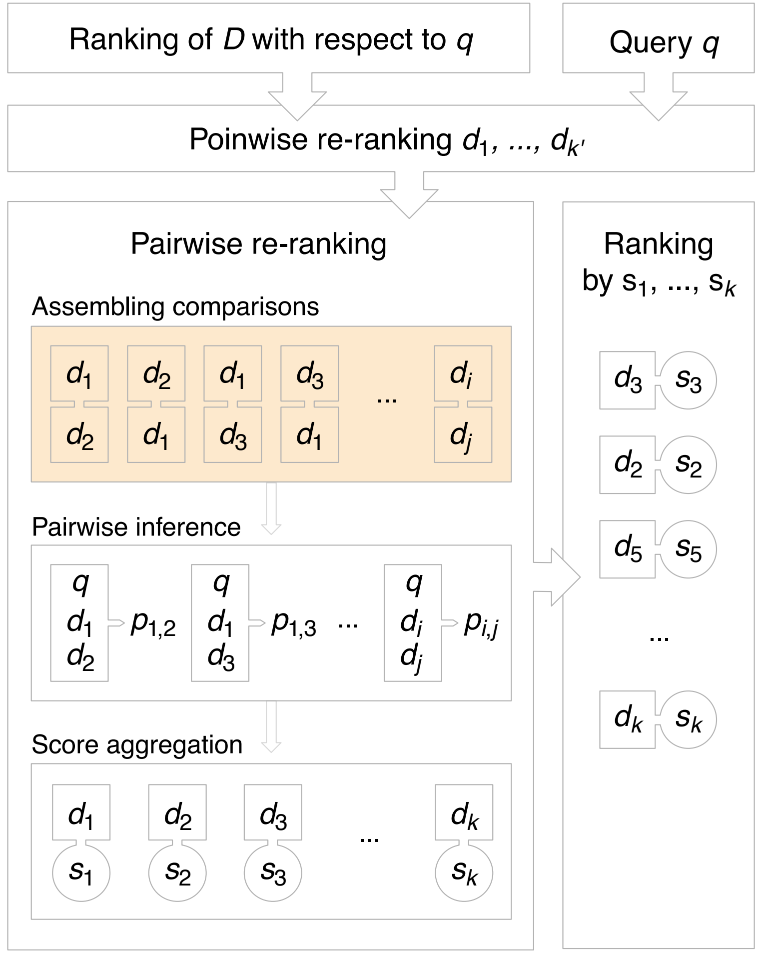
1. BM25 ranking (whole corpus)
2. Pointwise re-ranking (top 1000)
3. Pairwise re-ranking (top 50)
   - ❑ assemble document pairs
   - ❑ pairwise inference
   - ❑ score aggregation
4. Rank by aggregated score



Contributions

Key improvements in the pairwise step:

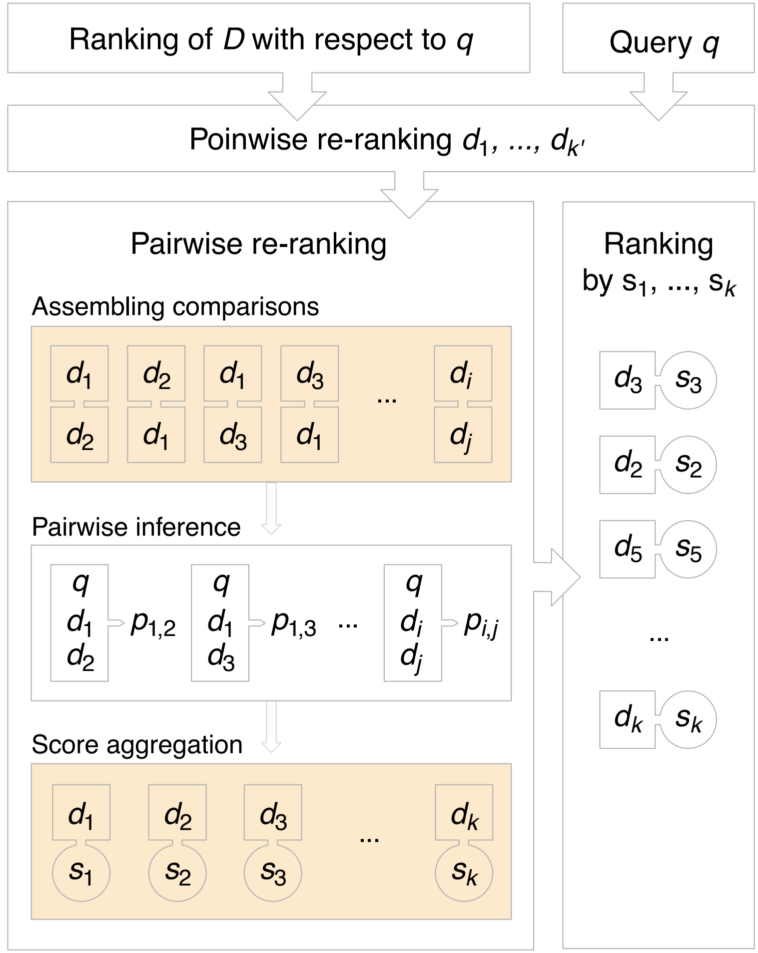
1. Efficiency
   - ❑ quadratic comparison amount when doing all doc-doc pairs is problematic
   - ❑ sparse comparison set for efficiency
   - ❑ But: requires good sampling approach



Contributions

Key improvements in the pairwise step:

1. Efficiency
   - quadratic comparison amount when doing all doc-doc pairs is problematic
   - sparse comparison set for efficiency
   - But: requires good sampling approach
2. Effectiveness
   - choice of aggregation method has direct impact on effectiveness
   - little attention in previous work
   - we investigate several aggregation methods with an without sampling



Sorting as Aggregation

Sorting: The most efficient solution we can hope for

- Kwiksort: "Quicksort" for pairwise preferences
- Complexity: $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$

Sorting as Aggregation

Sorting: The most efficient solution we can hope for

- Kwiksort: “Quicksort” for pairwise preferences
- Complexity: $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$

But: requires total order between predictions

- **consistency**: score of document pair (d_a, d_b) should be the inverse of (d_b, d_a)
- **transitivity**: predictions for three documents should be transitive

`duoT5` on MS MARCO

Property	Average Rate
Consistency	0.498
Transitivity	0.693

Average over all document pairs of 50 topics at depth 50.

Sorting as Aggregation

Sorting: The most efficient solution we can hope for

- Kwiksort: "Quicksort" for pairwise preferences
- Complexity: $\mathcal{O}(n \log n)$ instead of $\mathcal{O}(n^2)$

But: requires total order between predictions

- **consistency**: score of document pair (d_a, d_b) should be the inverse of (d_b, d_a)
- **transitivity**: predictions for three documents should be transitive

`duoT5` on MS MARCO

Property	Average Rate
Consistency	0.498
Transitivity	0.693

Average over all document pairs of 50 topics at depth 50.

MS MARCO (Passage; DL 19/20; k=50 documents).

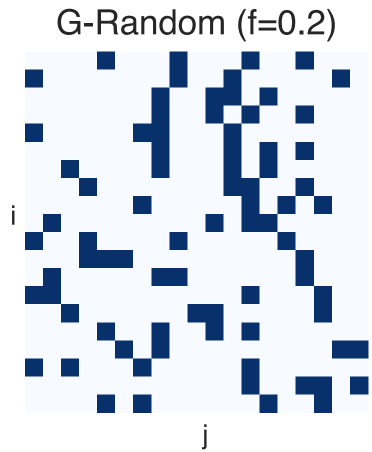
Pipeline	No. Comp.	nDCG@10
monoT5	0	0.50
+ duoT5	2450	0.67
+ duoT5 with Kwiksort	85	0.42

Pairwise model output contains too many individual errors to sort!

Sampling Methods

Random Sampling

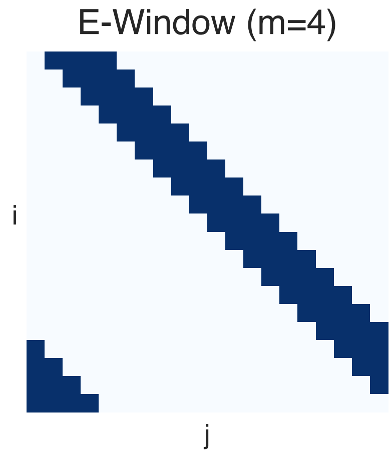
- **Motivation**: baseline method
- **Method**:
  - randomly sample a fraction f of possible comparisons
  - sampling is separate per doc.
- **Upside**: parameter-free
- **Downside**: not deterministic, pointwise ranking is not used

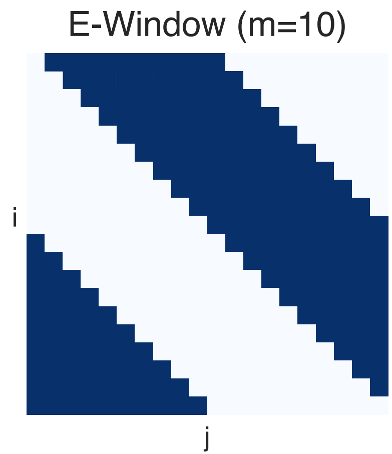


Sampling Methods

Exhaustive Window Sampling

- **Motivation**: deterministic method
- **Method**:
  - based on pointwise reranking
  - compares a doc. to its m successors
  - wraps around to compare last to first
- **Upside**: parameter-free, incorporates pointwise ranking context locally
- **Downside**: global context lost, cannot stray far from pointwise ranking

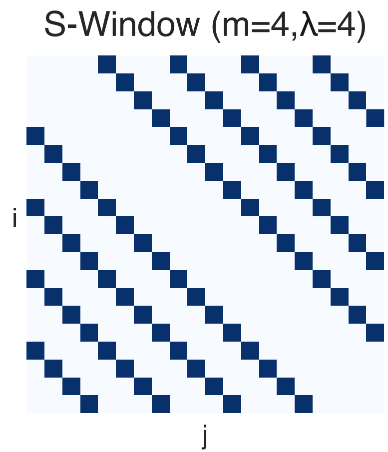


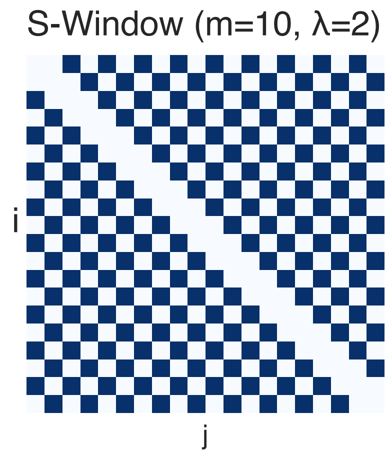


Sampling Methods

Skip Window Sampling

- **Motivation**: deterministic + global method
- **Method**:
  - like exhaustive window sampling
  - skips with steps size λ
- **Upside**: incorporates pointwise ranking context globally
- **Downside**: parametric, λ has to be tuned





Aggregation Methods

Four different aggregation methods, each from a different aggregation paradigm.

Additive Aggregation

- baseline [Pradeep et. al 2021]
- symmetric sum of preference scores

Aggregation Methods

Four different aggregation methods, each from a different aggregation paradigm.

Additive Aggregation

- baseline [Pradeep et. al 2021]
- symmetric sum of preference scores

Bradley-Terry Aggregation

- maximum-likelihood logistic regression
- optimizes to fit pairwise preferences

Aggregation Methods

Four different aggregation methods, each from a different aggregation paradigm.

Additive Aggregation

- baseline [Pradeep et. al 2021]
- symmetric sum of preference scores

Bradley-Terry Aggregation

- maximum-likelihood logistic regression
- optimizes to fit pairwise preferences

Greedy Aggregation

- similar to additive
- identify best doc., then recursively apply to remaining

Aggregation Methods

Four different aggregation methods, each from a different aggregation paradigm.

Additive Aggregation

- baseline [Pradeep et. al 2021]
- symmetric sum of preference scores

Bradley-Terry Aggregation

- maximum-likelihood logistic regression
- optimizes to fit pairwise preferences

Greedy Aggregation

- similar to additive
- identify best doc., then recursively apply to remaining

PageRank Aggregation

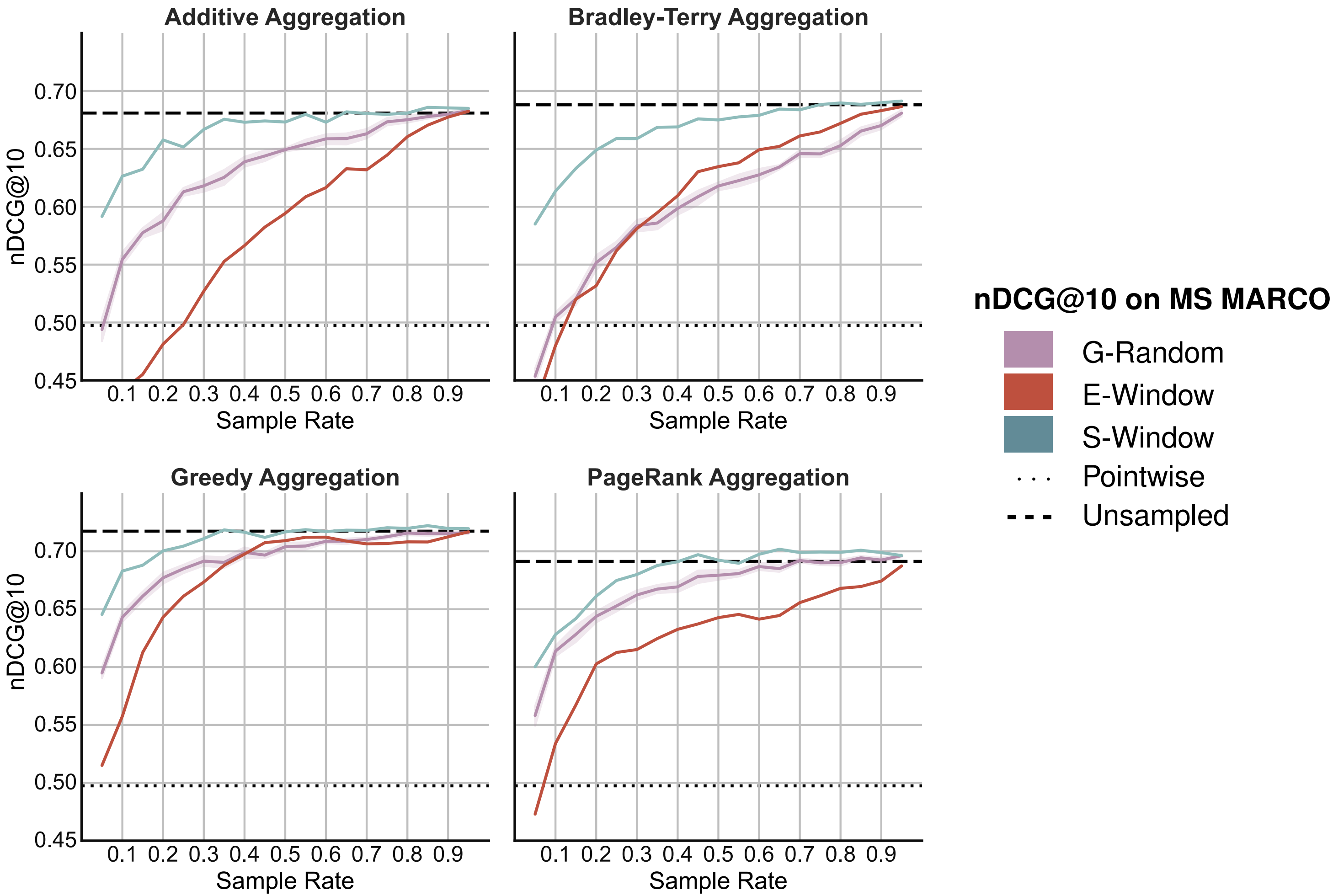
- graph-based aggregation
- docs. are nodes, comparisons are weighted edges

Evaluation

Experimental Setup

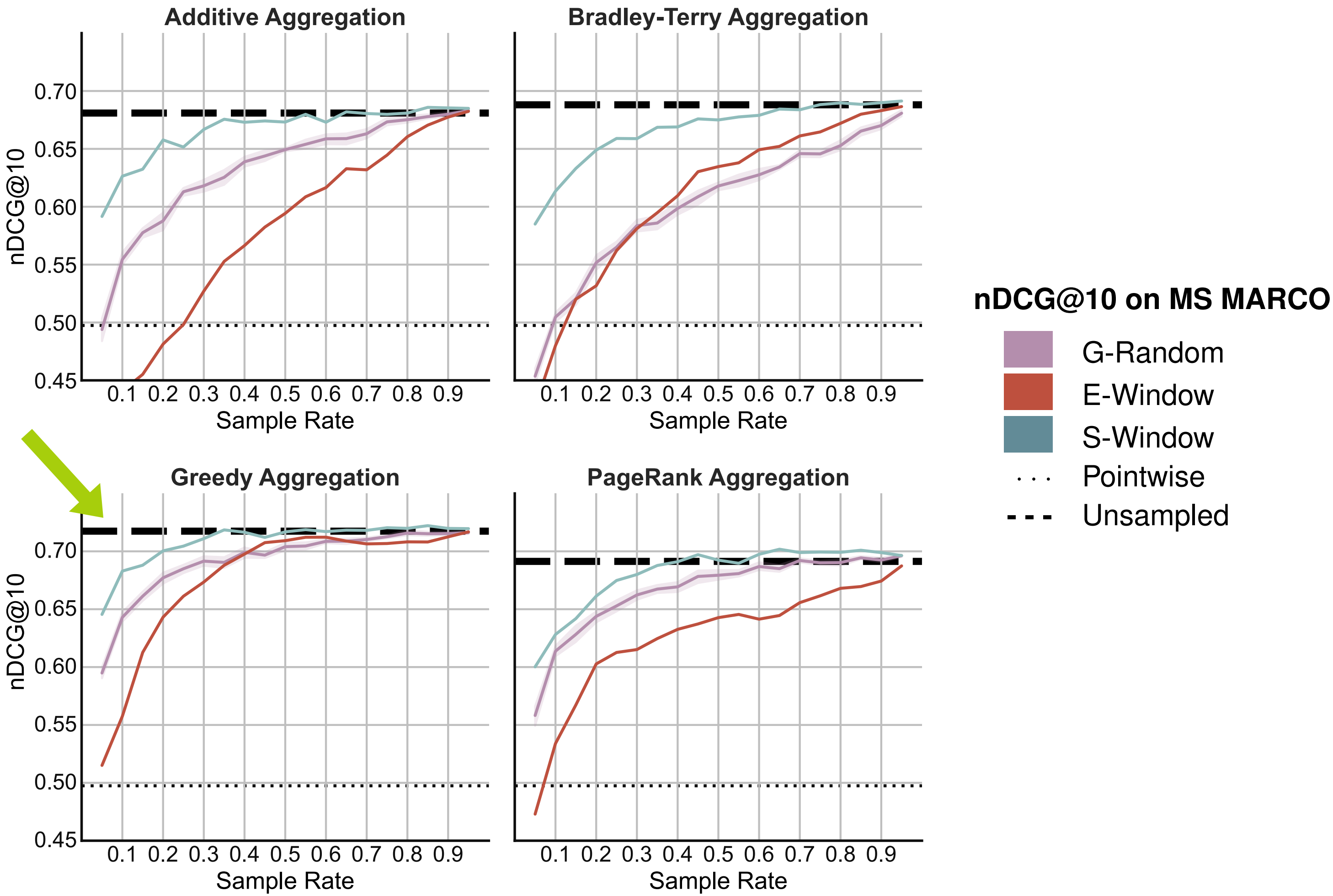
- **Collection**: MS MARCO
- **Ranking Pipeline**:
  1. BM25 with default parameters
  2. Top 1000 reranking with monoT5
  3. Top 50 reranking with duoT5
- **Measure**: nDCG@10 with qrels from TREC-DL passage ranking
- **Parameters**: grid search was carried out to find optimal λ-value for S-Window sampling

Evaluation



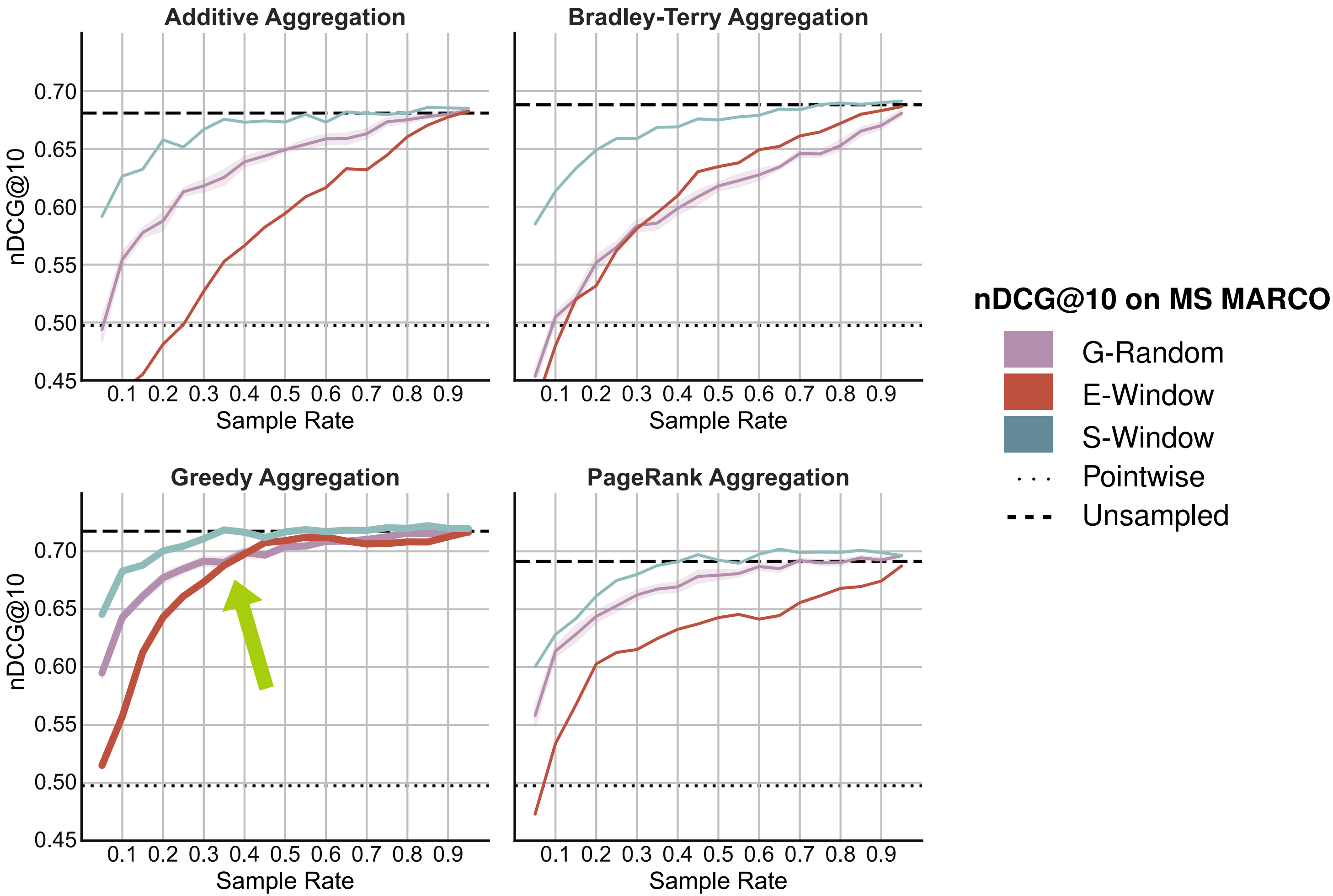
Additive Aggregation
Bradley-Terry Aggregation
Greedy Aggregation
PageRank Aggregation
nDCG@10
0.45
0.50
0.55
0.60
0.65
0.70
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9
Sample Rate
nDCG@10 on MS MARCO
G-Random
E-Window
S-Window
Pointwise
Unsampled

Evaluation



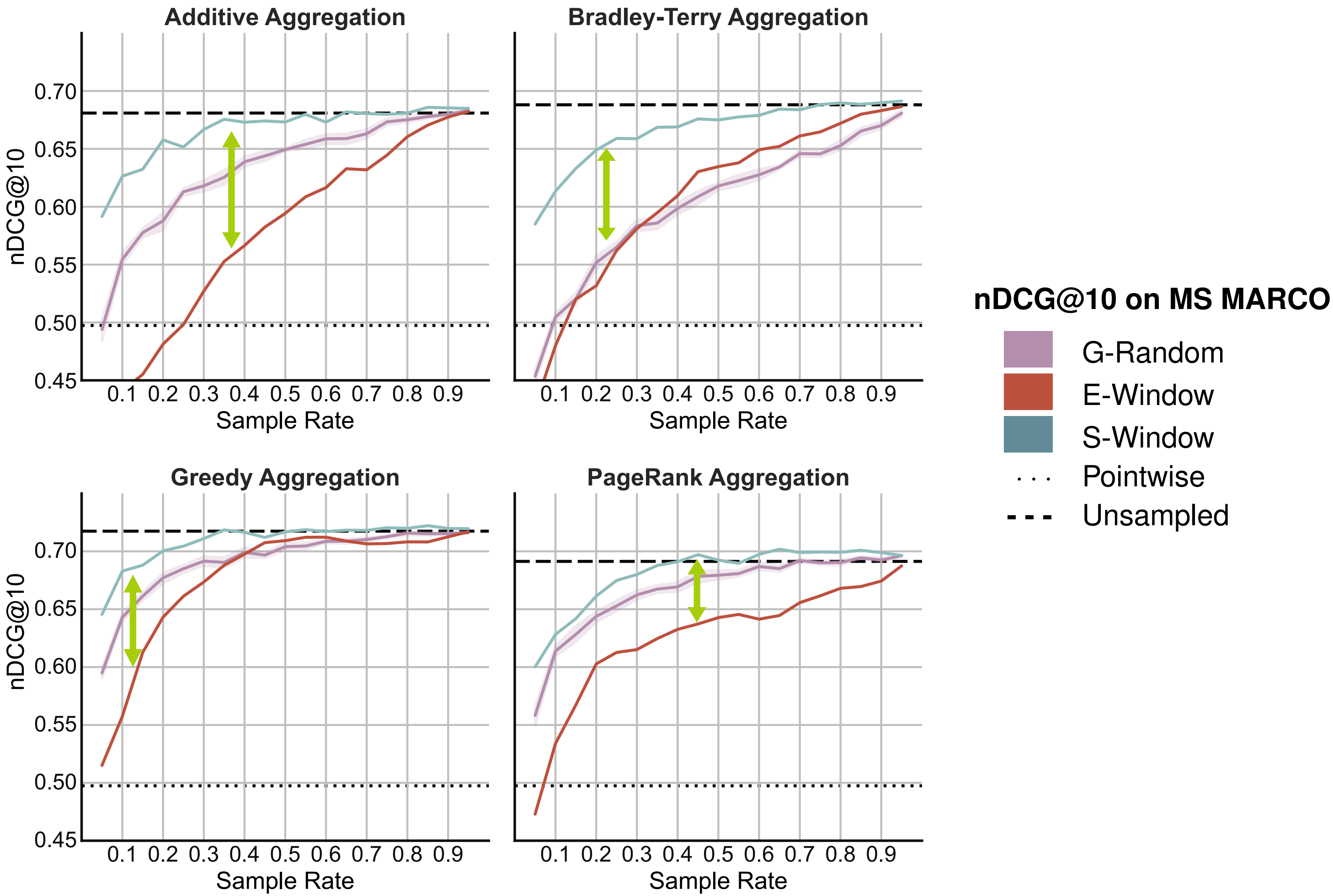
Greedy aggregation is best under no sampling.

Evaluation



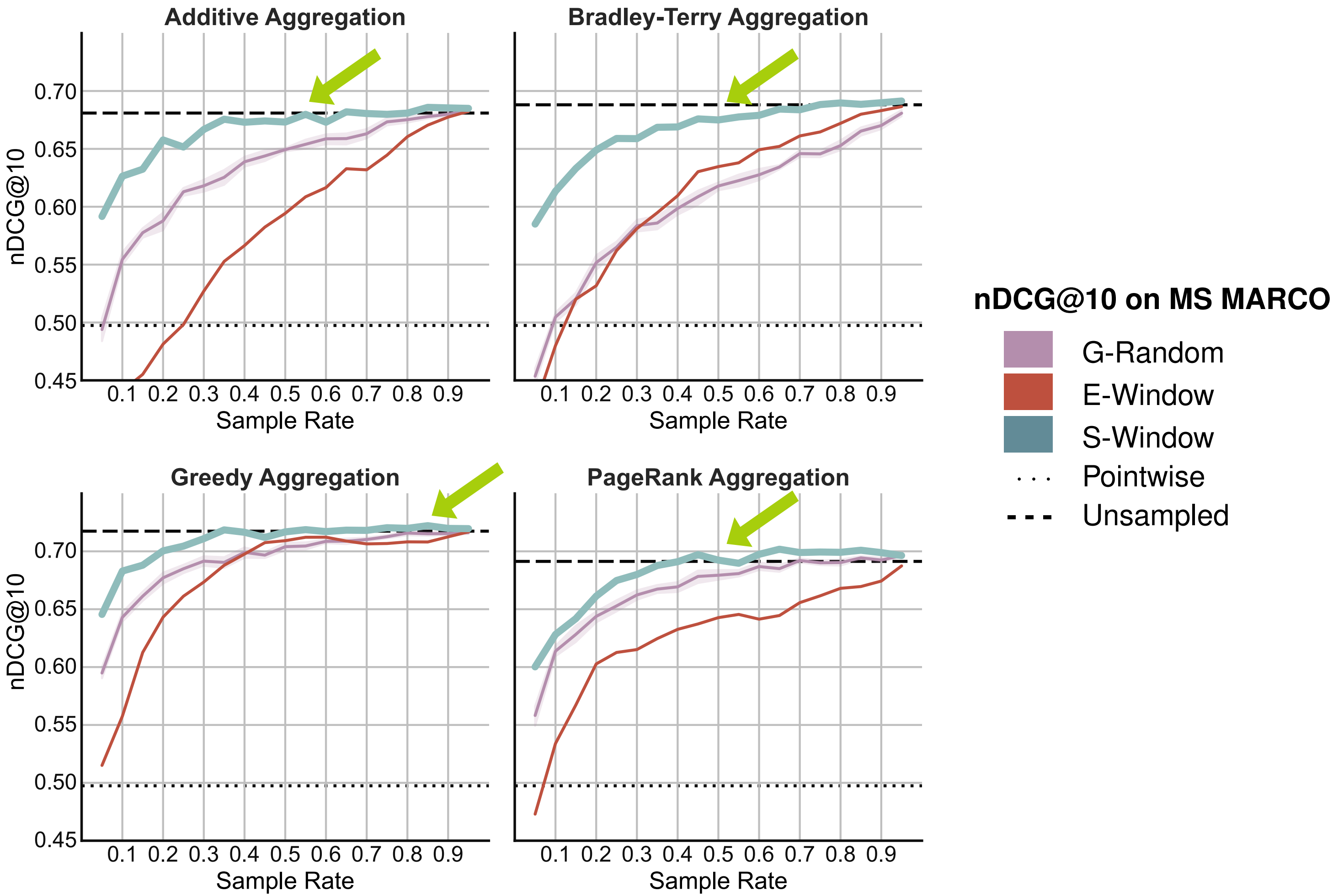
Greedy aggregation is best across all sampling methods.

Evaluation



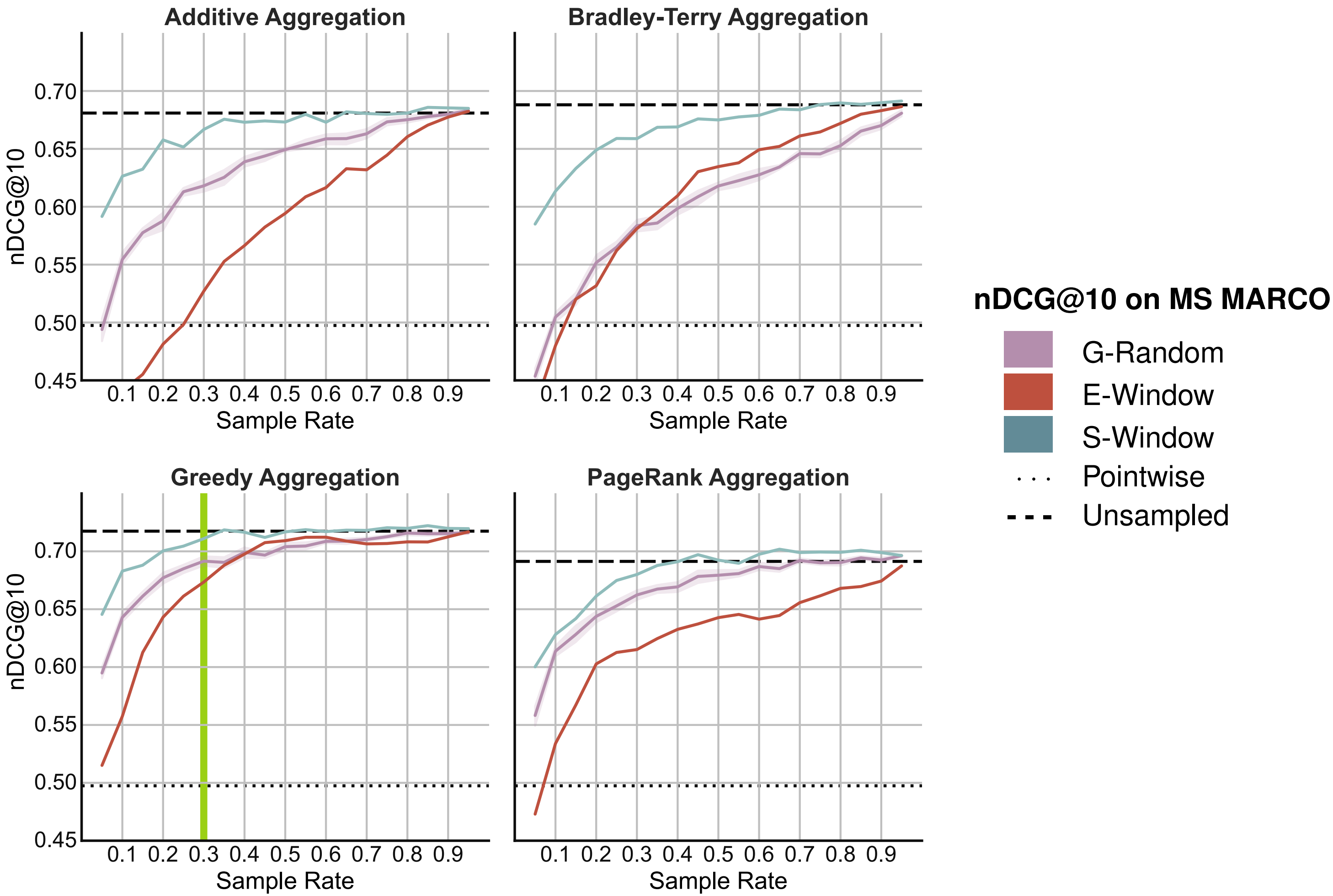
Global sampling context seems more important than local sampling context.

Evaluation



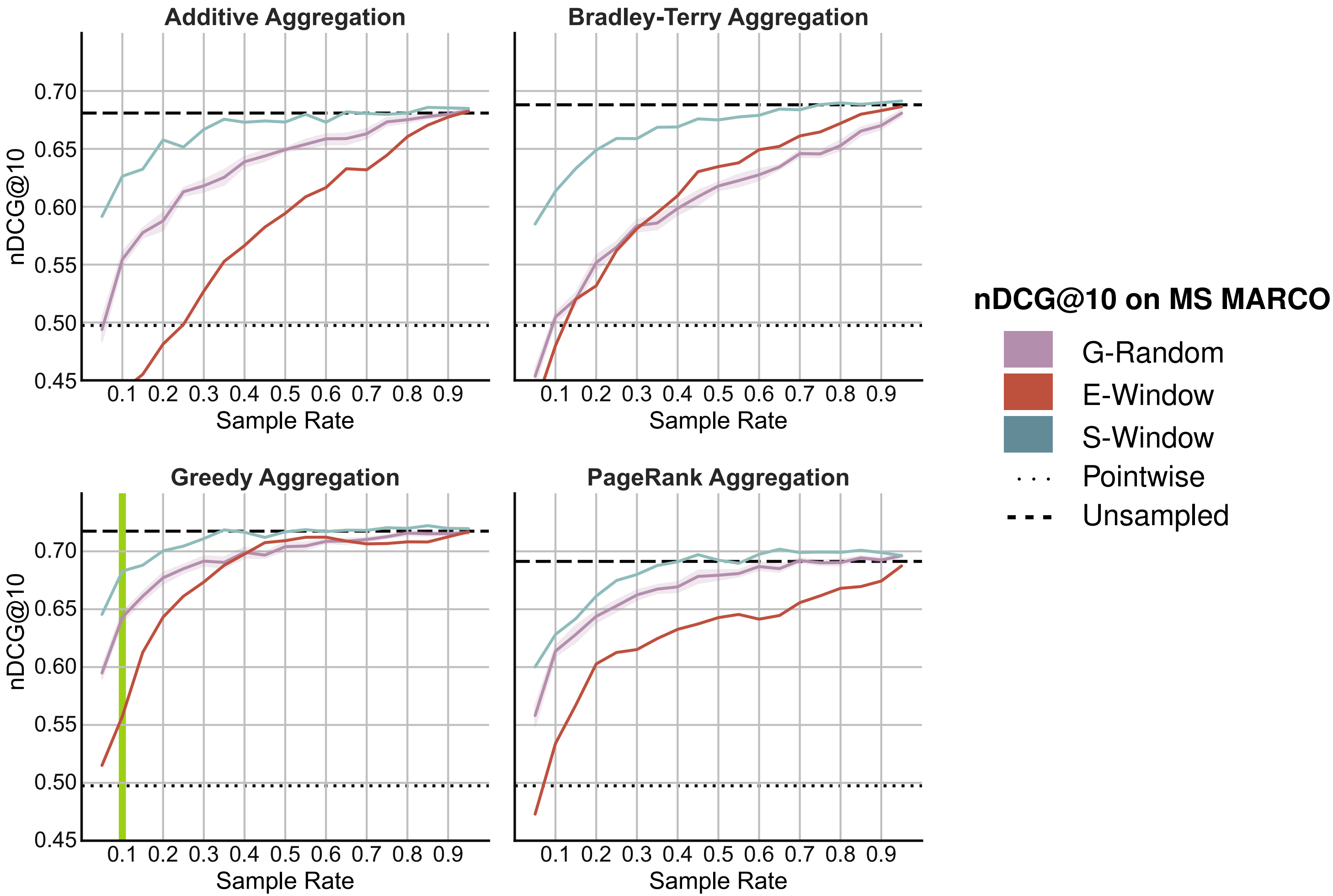
S-Window sampling is best across all aggregation methods.

Evaluation



Best setup matches effectiveness down to 30% of the comparisons.

Evaluation



Best setup is competitive down to 10% of the comparisons. ($\Delta = 0.04$)

Conclusion

Findings:

- Sparse comparison sets are highly effective at increasing the efficiency of pairwise retrieval
- Effectiveness can be increased with better aggregation approaches
- Up to 90% cost savings are possible

Conclusion

Findings:

- Sparse comparison sets are highly effective at increasing the efficiency of pairwise retrieval
- Effectiveness can be increased with better aggregation approaches
- Up to 90% cost savings are possible

Whats more in the paper?

- Replication of evaluation on CW09 and CW12, corroborating results
- More in-depth evaluation of comparison properties
- Statistical testing

Conclusion

Findings:

- Sparse comparison sets are highly effective at increasing the efficiency of pairwise retrieval
- Effectiveness can be increased with better aggregation approaches
- Up to 90% cost savings are possible

Whats more in the paper?

- Replication of evaluation on CW09 and CW12, corroborating results
- More in-depth evaluation of comparison properties
- Statistical testing

Whats more in the future?

- Instead of lower budget at same depth, increase depth at same budget
- Promising for high-recall search applications
- Model adaptions for more consistent predictions, dynamic sampling approaches