SIGIR 2025
Padova
ITALY

# Breaking the Lens of the Telescope: Online Relevance Estimation over Large Retrieval Sets

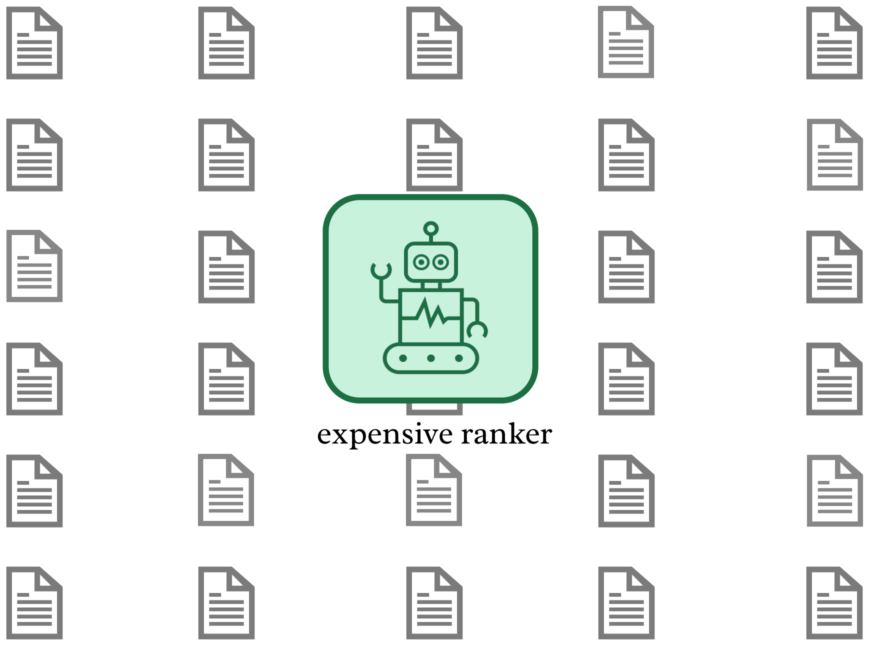
**Mandeep Rathee**, Venktesh V, Sean MacAvaney, and Avishek Anand

L3S AI Research

University of Glasgow

TU Delft Delft University of Technology

# Search Systems




Given:

- query from the user
- huge corpus with millions or billions of documents
- a ranker (e.g., MonoT5)

Our Goal:

- top-k documents

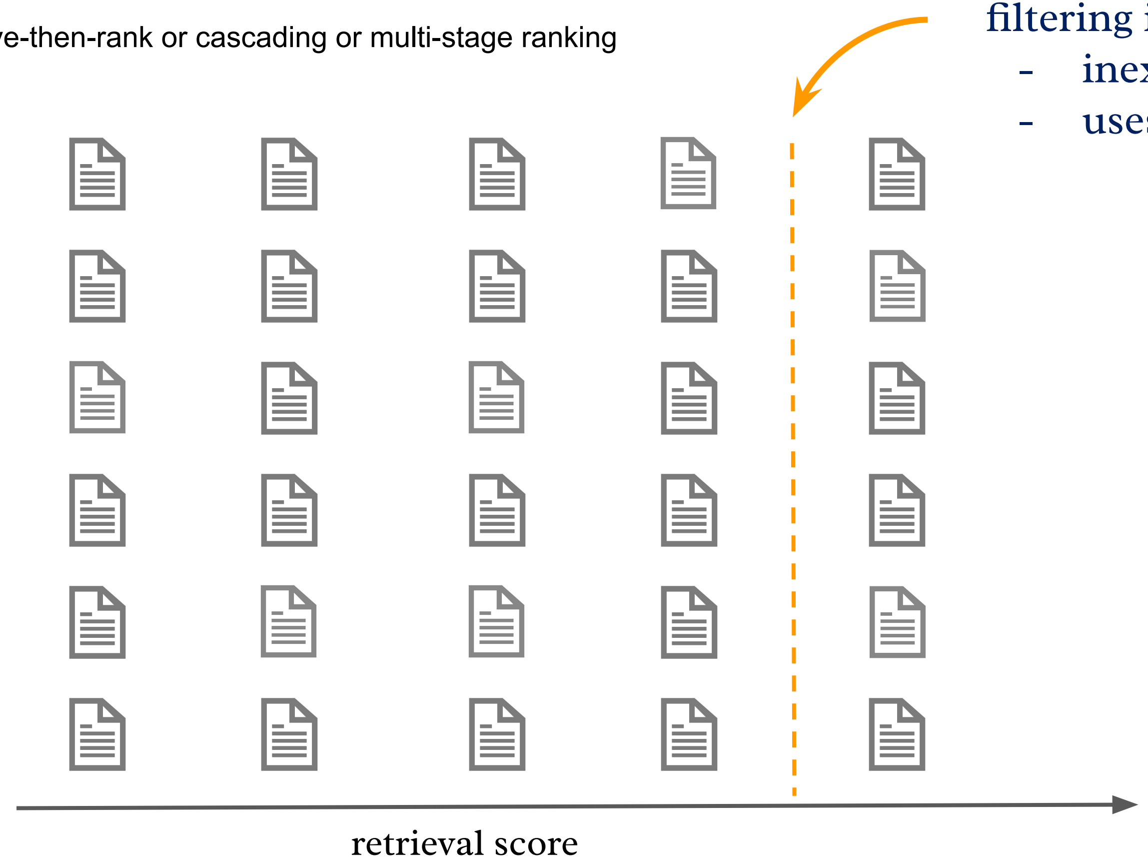
Constraint:

- a ranking budget of c documents.

# Telescoping Setting

retrieve-then-rank or cascading or multi-stage ranking

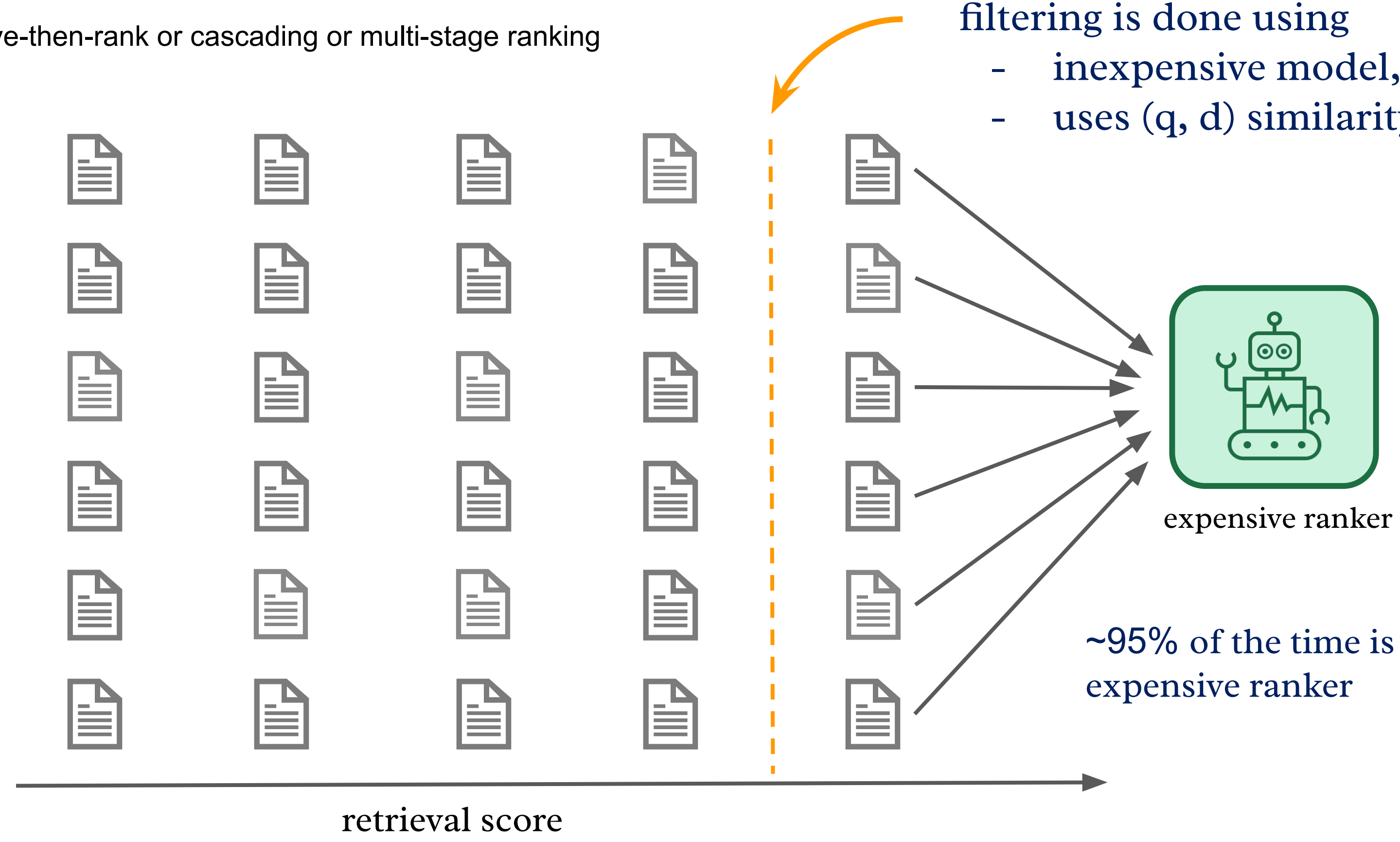
filtering is done using
- inexpensive model, e.g., BM25
- uses (q, d) similarity



# Telescoping Setting

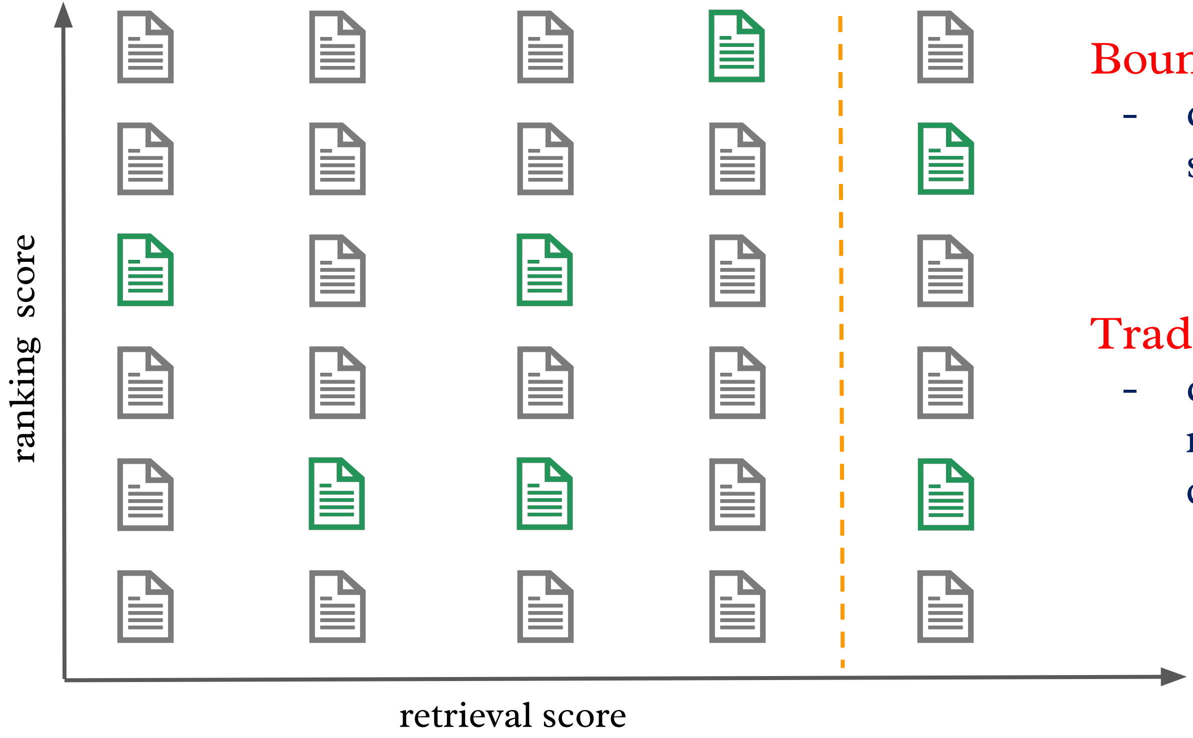
retrieve-then-rank or cascading or multi-stage ranking

filtering is done using
- inexpensive model, e.g., BM25
- uses (q, d) similarity




~95% of the time is taken by the expensive ranker

# Limitations in Telescoping Setting



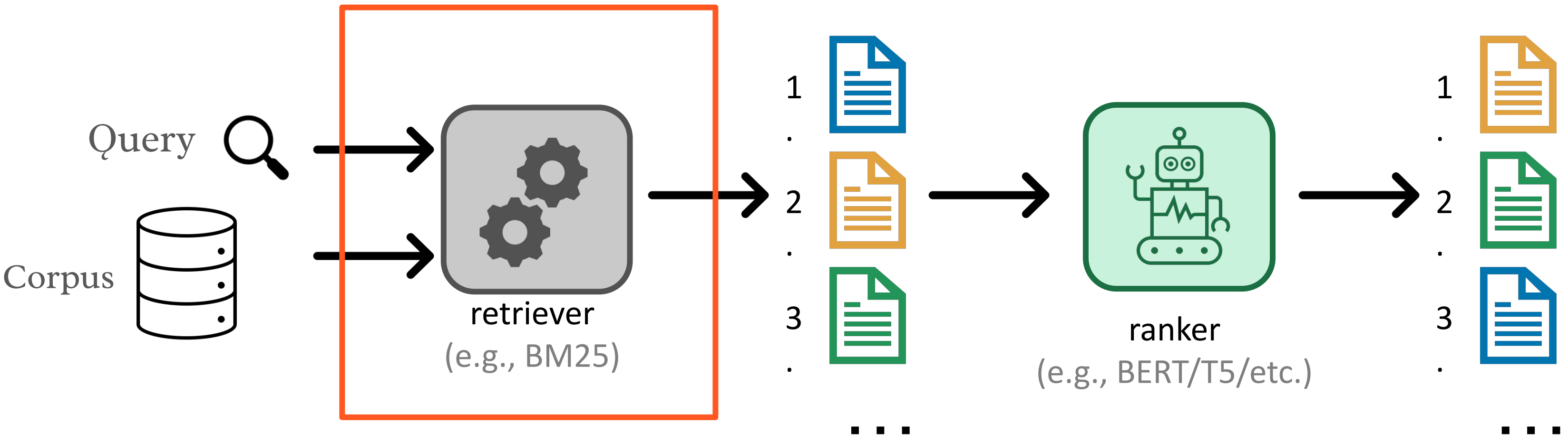

**Bounded Recall problem**

- documents missed at first stage will never be recovered.

**Traditional Philosophy**

- documents are scheduled for ranking only based on the query-document similarities.

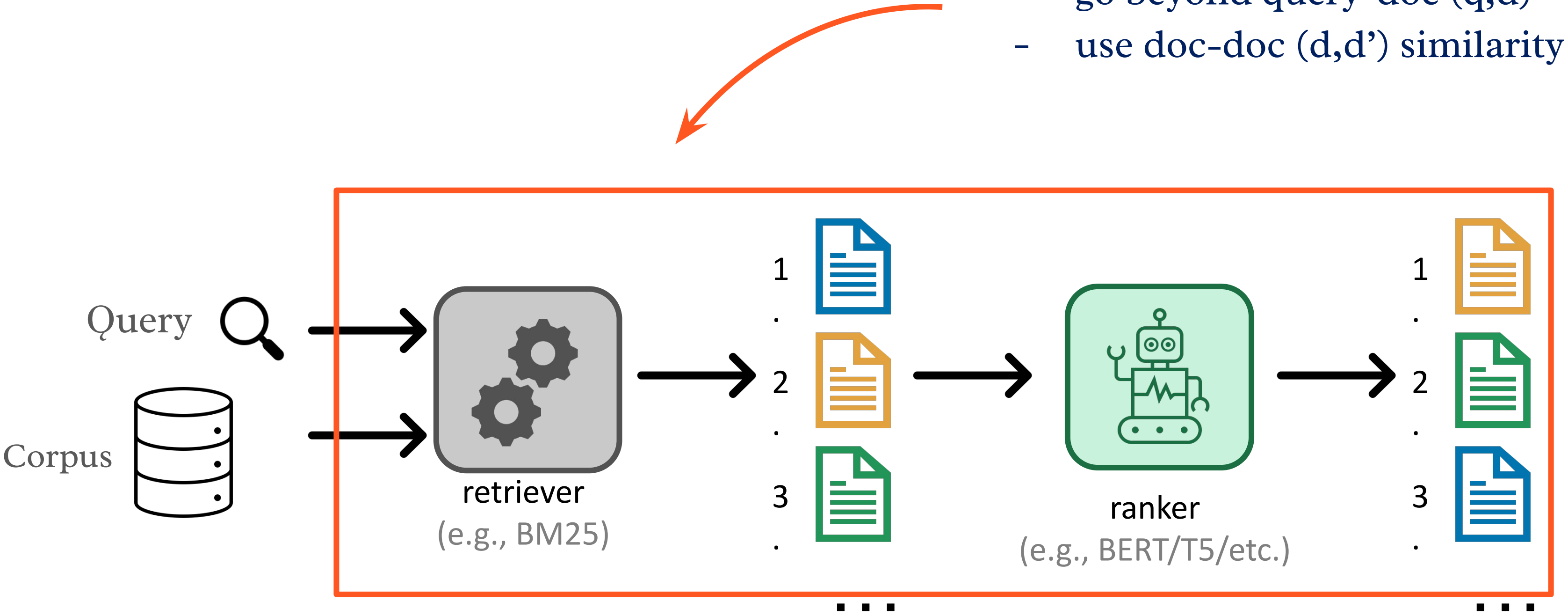
- focus on improving query-doc (q,d) similarity




Hybrid Retrieval

Reciprocal Rank Fusion (RRF)
Convex Combination (CC)

- go beyond query-doc (q,d)
- use doc-doc (d,d') similarity




Adaptive Retrieval

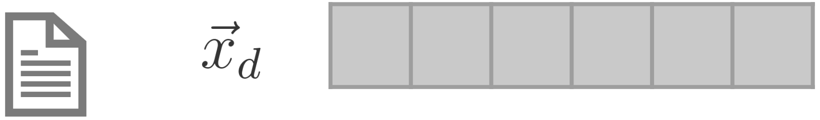
GAR (MacAvaney et al. 2022)
Quam (Rathee et al. 2025)

# Our Method

## ORE: Online Relevance Estimation

- Based on the **Multi-Arm-Bandit** framework
- Dynamically selects batches during re-ranking using **exploration and exploitation**
- Approximates the expensive ranker by a simple linear model

# Online Relevance Estimation



$\vec{x}_d$

exploration and exploitation

Features taxonomy:

**Q2D:**

- Lexical : $BM25(q, d)$
- Semantic: $TCT(q, d)$

**D2D:**

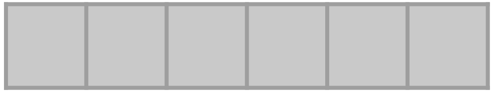
- Lexical: $BM25(d, d')$
- Semantic: $TCT(d, d')$
- Learned affinity: $\textsc{Laff}(d, d')$ [1]

**D2Set:**

already high ranked documents

[1] Quam: Adaptive Retrieval through Query Affinity Modelling, Rathee et. al., 2025

# Online Relevance Estimation

$\vec{x}_d$ 

$$\textsc{EstRel}(\vec{\alpha}, \vec{x}_d) = \vec{\alpha} \cdot \vec{x}_d^T$$

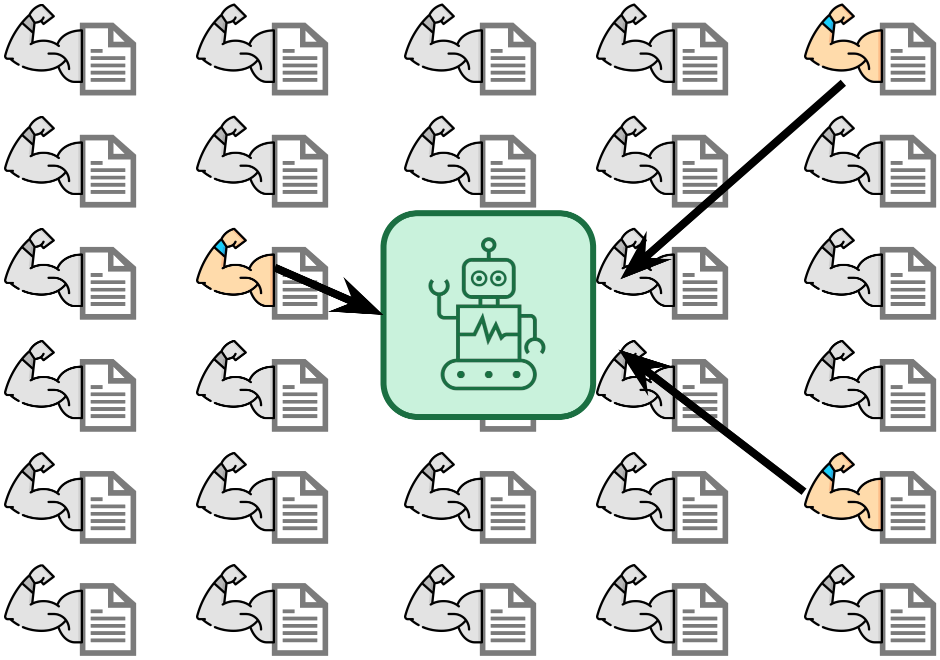
$$E(\vec{\alpha}; q, d, \vec{x}_d) = \frac{1}{2} \left| \phi(q, d) - \textsc{EstRel}\left(\vec{\alpha}, \vec{x}_d\right) \right|^2$$

*minimize* $E(\vec{\alpha}; q, d, \vec{x}_d)$

- We treat each document as an arm.

- Estimate the utility or relevance (EstRel) of each arm.

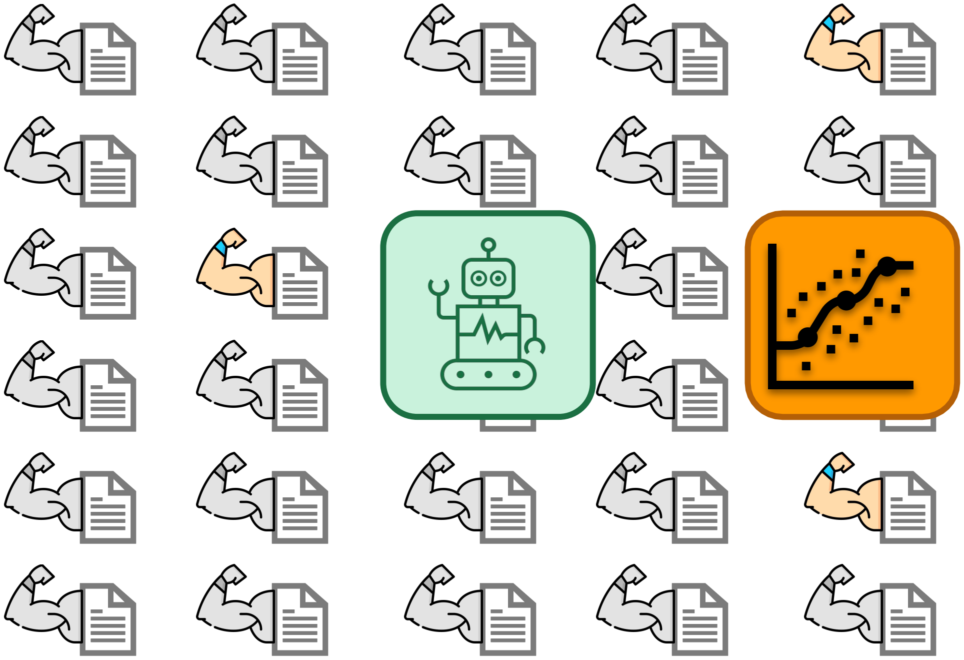
- We treat each document as an arm.
- Estimate the utility or relevance (EstRel) of each arm.
- Select b most promising arms.



- We treat each document as an arm.

- Estimate the utility or relevance (EstRel) of each arm.

- Select b most promising arms and rank these b arms.



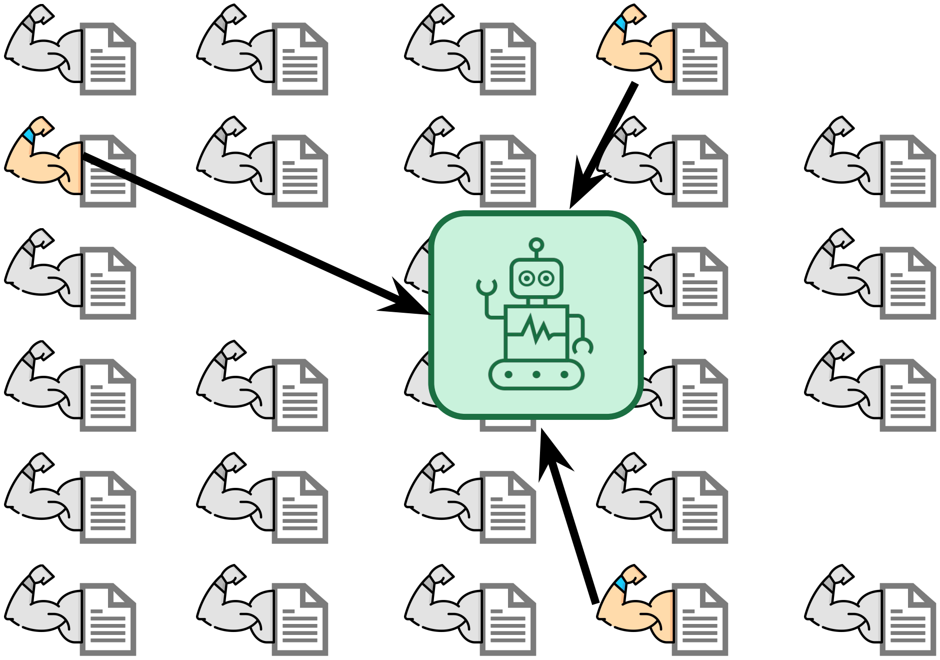
$$E(\vec{\alpha}; q, d, \vec{x}_d) = \frac{1}{2} \left| \phi(q, d) - \textsc{EstRel}\left(\vec{\alpha}, \vec{x}_d\right) \right|^2$$

*minimize* $E(\vec{\alpha}; q, d, \vec{x}_d)$

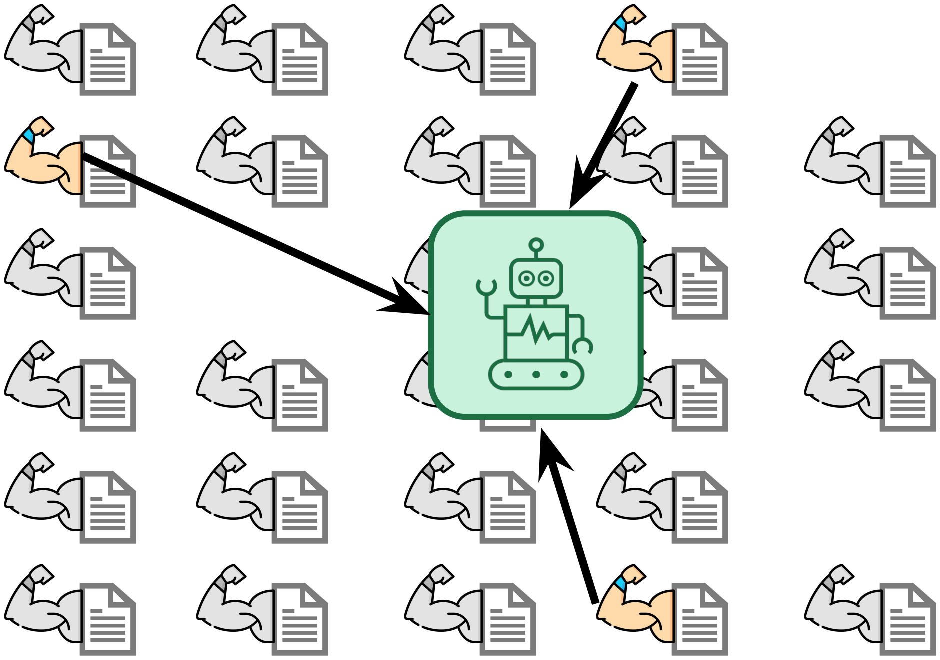
- We treat each document as an arm.
- Estimate the utility or relevance (EstRel) of each arm.
- Select b most promising arms and rank these b arms.
- Compute regret/error and update alphas.

- We treat each document as an arm.
- Estimate the utility or relevance (EstRel) of each arm.
- Select b most promising arms and rank these b arms.
- Compute regret/error and update alphas
- Remove these already ranked arms.

- We treat each document as an arm.
- Estimate the utility or relevance (EstRel) of each arm.
- Select b most promising arms and rank these b arms.
- Compute regret/error and update alphas
- Remove these already ranked arms.
- Re-evaluate the utility of remaining arms.

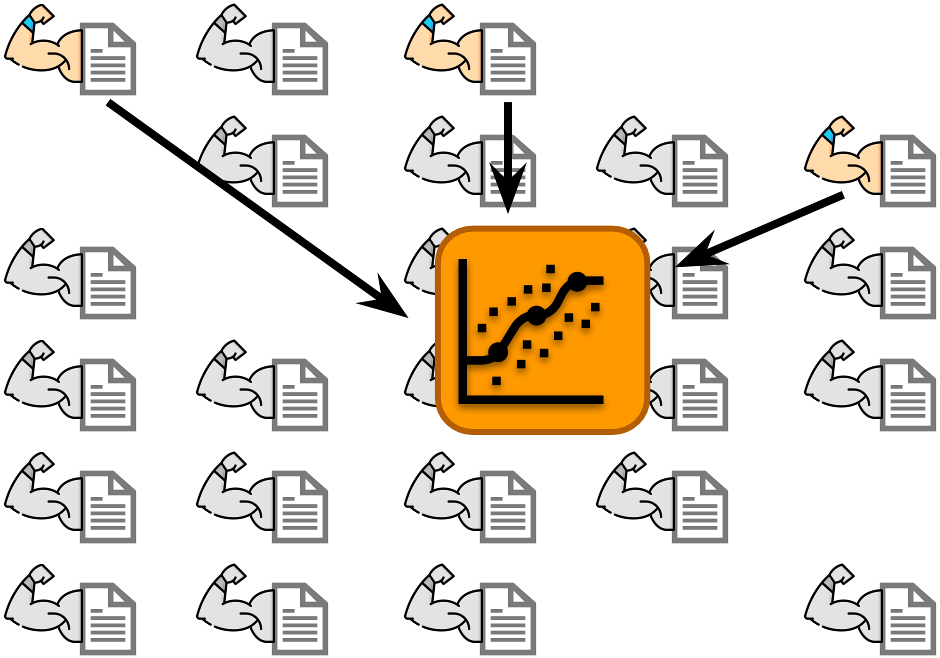


- We treat each document as an arm.
- Estimate the utility or relevance (EstRel) of each arm.
- Select b most promising arms and rank these b arms.
- Compute regret/error and update alphas.
- Remove these already ranked arms.
- Re-evaluate the utility of remaining arms.




until we meet the
stopping criteria.

- We treat each document as an arm.
- Estimate the utility or relevance (EstRel) of each arm.
- Select b most promising arms and rank these b arms.
- Compute regret/error and update alphas.
- Remove these already ranked arms.
- Re-evaluate the utility of remaining arms.



Now we can:

- approximate the scores of the expensive ranker by a simple model.

- dynamically select documents for ranking based on the utility (EstRel) scores.

# Experimental Setup

Retrievers

- sparse (BM25) and dense (TCT)

Retrieval Setting

- Hybrid (RRF and CC)
- Adaptive (GAR and Quam)

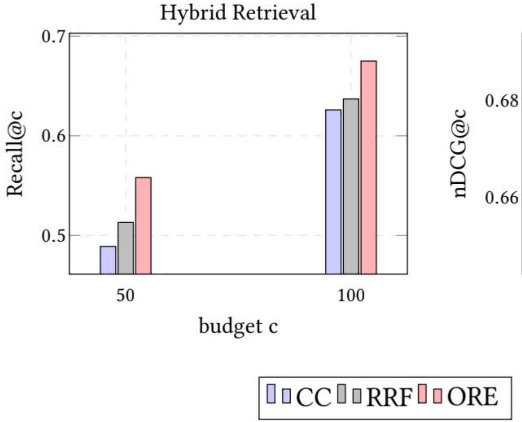
Rerankers

- MonoT5 and RankLLaMA

Evaluation

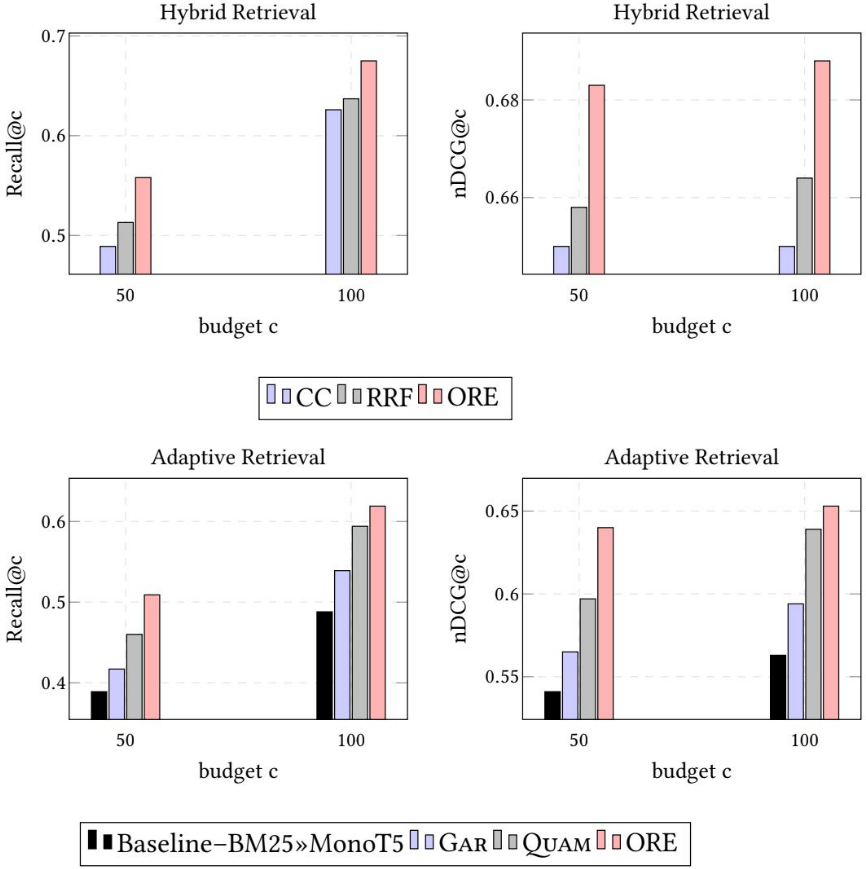
- TREC DL19 and DL20 using msmarco-passage corpus.
- TREC DL21 and DL22 using msmarco-passage-v2 corpus.

# EstRel helps in documents prioritization



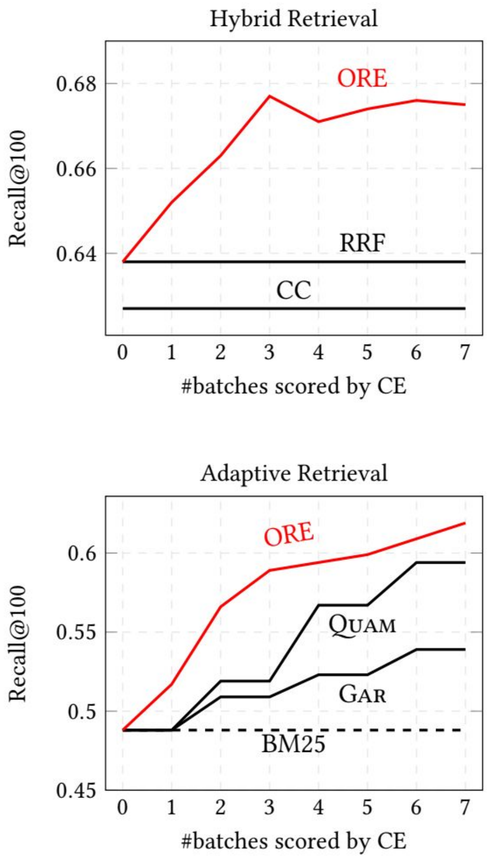

- Ranker: MonoT5
- ORE outperforms the baselines in both recall and nDCG at different ranking budgets.
- In hybrid: Recall@50 improves upto 14%.

# EstRel helps in documents prioritization



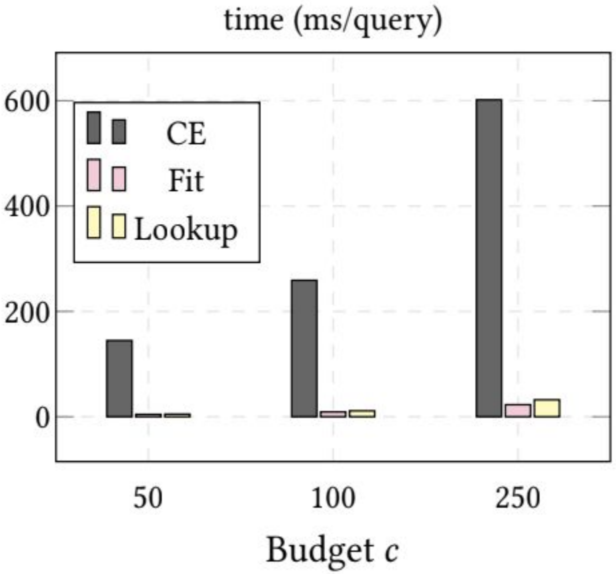

- Ranker: MonoT5
- ORE outperforms the baselines in both recall and nDCG at different ranking budgets.
- In hybrid: Recall@50 improves upto 14%.
- Similarly, ORE shows significant gains over baselines in adaptive setting.
- In adaptive: Recall@50 improves upto 30% over standard baseline, upto 11% over Quam.

# Quality of EstRel




- Estimated Relevance (EstRel) shows performance gains just after scoring first batch.
- ORE clearly shows the performance gains over baselines.
- ORE can achieve high recall while ranking the same number of batches.

# Computational Efficiency




- The majority of the latency come from the expensive ranker CE (Cross-Encoder).

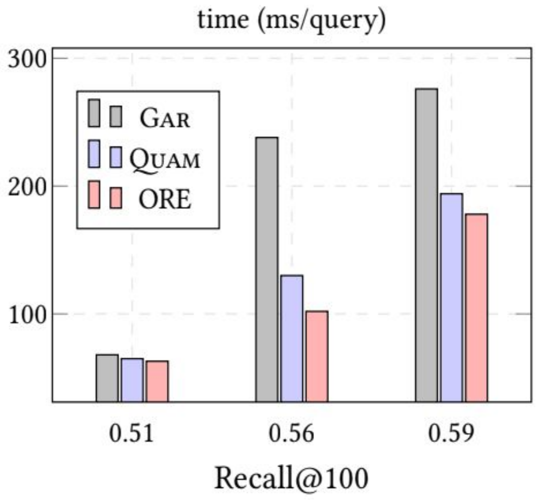
- Latency of fitting the simple linear model is negligible in comparison to the expensive cross encoder.

# Thank you

rathee@l3s.de

Find it on Github

# Computational Efficiency




- ORE is computationally efficient in comparison to baselines.
- ORE can achieve same recall in lower latency.

# Sample Efficiency

when use RankLLaMA to rerank 1000 documents with batch size of 16

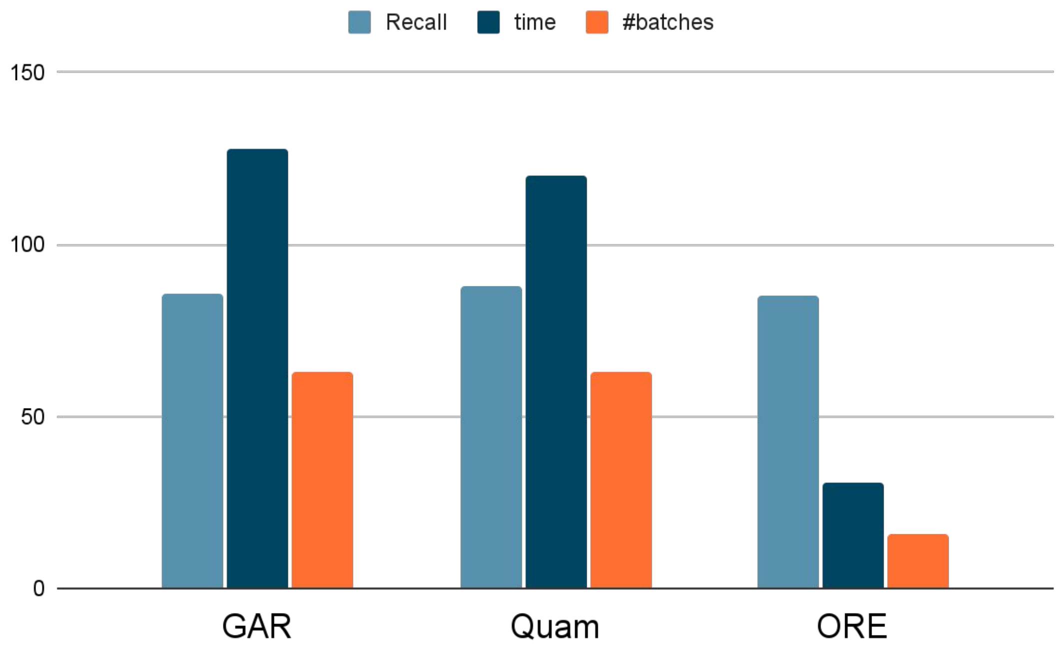


Table 1: **Description of different features used in ORE. These features can be divided into two levels, Q2DAFF and D2DAFF.**

| Feature | Notation | Taxonomy | Source | | Description |
|---|---|---|---|---|---|
| | | | Offline | Online | |
| $x_1$ | $BM25(q, d)$ | Q2DAFF | | ✓ | Lexical similarity between query and document. |
| $x_2$ | $TCT(q, d)$ | Q2DAFF | | ✓ | Semantic similarity between query and document. |
| $x_3$ | $RM3(q', d)$ | D2DAFF | | ✓ | Lexical similarity between expanded query using RM3 and document. |
| $x_4$ | $BM25(d, d')$ | D2DAFF | ✓ | | Lexical similarity between pair of documents. |
| $x_5$ | $TCT(d, d')$ | D2DAFF | ✓ | | Semantic similarity between pair of documents. |
| $x_6$ | $\textsc{Laff}(d, d')$ | D2DAFF | ✓ | | Learnt affinity or similarity between pair of documents [1]. |

Table 2: Effectiveness comparison of ORE with hybrid and adaptive retrieval methods on TREC DL19 and DL20 test sets. Significant improvements using paired t-test, $p < 0.05$, with Bonferroni correction, over CC, RRF, baseline (BM25»MonoT5), Gar, and Quam are marked with $B$, $C$, $R$, $G$ and $Q$ respectively. The best scores are highlighted in bold.

| | | $c = 50$ | | | $c = 100$ | | | $c = 1000$ | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Dataset | Pipeline | nDCG@10 | nDCG@c | Recall@c | nDCG@10 | nDCG@c | Recall@c | nDCG@10 | nDCG@c | Recall@c |
| DL19 | **Exhaustive Retrieval** | | | | | | | | | |
| | MonoT5 | 0.672 | 0.625 | 0.512 | 0.672 | 0.611 | 0.599 | 0.672 | 0.691 | 0.834 |
| | **Hybrid Retrieval: (BM25 & TCT)** | | | | | | | | | |
| | RRF»MonoT5 [R] | **0.735** | 0.658 | 0.513 | 0.729 | 0.664 | 0.637 | **0.703** | 0.740 | 0.879 |
| | CC»MonoT5 [C] | 0.729 | 0.650 | 0.489 | **0.730** | 0.650 | 0.626 | 0.698 | 0.738 | 0.878 |
| | ORE | 0.734 | $^{RC}$**0.683** | $^{RC}$**0.558** | 0.721 | $^{RC}$**0.688** | $^{RC}$**0.675** | **0.703** | **0.741** | **0.882** |
| | **Adaptive Retrieval** | | | | | | | | | |
| | BM25»MonoT5 [B] | 0.681 | 0.541 | 0.389 | 0.699 | 0.563 | 0.488 | 0.719 | 0.697 | 0.755 |
| | w/ RM3 | 0.682 | 0.560 | 0.409 | 0.686 | 0.574 | 0.507 | 0.728 | 0.717 | 0.786 |
| | w/ $\text{GAR}_{\text{BM25}}$ [G] | 0.689 | 0.565 | 0.417 | 0.716 | 0.594 | 0.539 | 0.727 | 0.742 | 0.836 |
| | w/ $\text{QUAM}_{\text{BM25}}$ [Q] | **0.698** | 0.597 | 0.460 | **0.729** | 0.639 | 0.594 | **0.742** | **0.770** | **0.874** |
| | w/ $\text{ORE}_{\text{BM25}}$ | **0.698** | $^{GQ}_{B}$**0.640** | $^{GQ}_{B}$**0.509** | 0.711 | $^{G}_{B}$**0.653** | $^{G}_{B}$**0.619** | 0.723 | $_{B}$0.759 | $_{B}$**0.874** |
| DL20 | **Exhaustive Retrieval** | | | | | | | | | |
| | MonoT5 | 0.649 | 0.592 | 0.576 | 0.649 | 0.593 | 0.670 | 0.649 | 0.682 | 0.852 |
| | **Hybrid Retrieval: (BM25 & TCT)** | | | | | | | | | |
| | RRF»MonoT5 [R] | **0.721** | 0.655 | 0.633 | 0.707 | 0.659 | 0.725 | 0.676 | 0.727 | 0.885 |
| | CC»MonoT5 [C] | 0.718 | 0.654 | 0.632 | **0.709** | 0.660 | 0.721 | **0.681** | 0.727 | 0.884 |
| | ORE | 0.720 | **0.674** | **0.658** | 0.702 | $^{RC}$**0.683** | $^{RC}$**0.759** | 0.676 | $^{C}$**0.731** | $^{C}$**0.892** |
| | **Adaptive Retrieval** | | | | | | | | | |
| | BM25»MonoT5 [B] | 0.676 | 0.559 | 0.478 | 0.685 | 0.581 | 0.584 | **0.720** | 0.711 | 0.807 |
| | w/ RM3 | 0.684 | 0.585 | 0.520 | 0.701 | 0.611 | 0.633 | **0.720** | 0.729 | 0.834 |
| | w/ $\text{GAR}_{\text{BM25}}$ [G] | 0.690 | 0.577 | 0.496 | 0.703 | 0.607 | 0.617 | 0.714 | 0.750 | 0.884 |
| | w/ $\text{QUAM}_{\text{BM25}}$ [Q] | **0.714** | 0.615 | 0.553 | **0.717** | **0.652** | 0.678 | 0.709 | 0.756 | **0.901** |
| | w/ $\text{ORE}_{\text{BM25}}$ | 0.684 | $^{G}_{B}$**0.621** | $^{G}_{B}$**0.583** | 0.681 | $^{G}_{B}$0.651 | $^{G}_{B}$**0.705** | 0.700 | **0.757** | $_{B}$0.892 |

Table 3: Effectiveness comparison* of ORE with hybrid and adaptive retrieval methods on TREC DL21 and DL22 test sets. The letter in subscript or superscript shows significant improvements (using paired t-test, $p < 0.05$, with Bonferroni correction) over the corresponding baseline.

| | | $c = 50$ | | $c = 100$ | |
|---|---|---|---|---|---|
| Dataset | Pipeline | nDCG@c | Recall@c | nDCG@c | Recall@c |
| DL21 | **Hybrid** | | | | |
| | RRF»MonoT5 [R] | 0.576 | 0.401 | 0.558 | 0.520 |
| | CC»MonoT5 [C] | 0.584 | 0.419 | 0.569 | 0.545 |
| | ORE | $^{R}$**0.604** | $^{R}$**0.444** | $^{RC}$**0.609** | $^{RC}$**0.609** |
| | **Adaptive** | | | | |
| | BM25»MonoT5 [B] | 0.436 | 0.242 | 0.433 | 0.331 |
| | w/ RM3 | 0.455 | 0.274 | 0.457 | 0.375 |
| | w/ $\text{GAR}_{\text{BM25}}$ [G] | 0.457 | 0.290 | 0.465 | 0.414 |
| | w/ $\text{QUAM}_{\text{BM25}}$ [Q] | 0.478 | 0.310 | **0.499** | 0.454 |
| | w/ $\text{ORE}_{\text{BM25}}$ | $^{GQ}_{B}$**0.503** | $^{GQ}_{B}$**0.364** | $_{B}$0.481 | $^{G}_{B}$**0.463** |
| | w/ $\text{GAR}_{\text{TCT}}$ [G] | 0.502 | 0.331 | **0.520** | 0.489 |
| | w/ $\text{QUAM}_{\text{TCT}}$ [Q] | 0.491 | 0.311 | 0.518 | 0.477 |
| | w/ $\text{ORE}_{\text{TCT}}$ | $^{GQ}_{B}$**0.532** | $^{GQ}_{B}$**0.406** | $_{B}$0.512 | $_{B}$**0.502** |
| DL22 | **Hybrid** | | | | |
| | RRF»MonoT5 [R] | 0.452 | 0.260 | 0.430 | 0.341 |
| | CC»MonoT5 [C] | 0.459 | 0.278 | 0.433 | 0.362 |
| | ORE | $^{RC}$**0.481** | $^{R}$**0.297** | $^{RC}$**0.459** | $^{RC}$**0.389** |
| | **Adaptive** | | | | |
| | BM25»MonoT5 [B] | 0.290 | 0.115 | 0.275 | 0.164 |
| | w/ RM3 | 0.287 | 0.115 | 0.275 | 0.161 |
| | w/ $\text{GAR}_{\text{BM25}}$ [G] | 0.287 | 0.121 | 0.290 | 0.191 |
| | w/ $\text{QUAM}_{\text{BM25}}$ [Q] | **0.308** | 0.135 | **0.303** | **0.196** |
| | w/ $\text{ORE}_{\text{BM25}}$ | 0.292 | **0.137** | 0.284 | 0.195 |
| | w/ $\text{GAR}_{\text{TCT}}$ [G] | 0.329 | 0.157 | **0.348** | 0.256 |
| | w/ $\text{QUAM}_{\text{TCT}}$ [Q] | 0.329 | 0.155 | 0.334 | 0.237 |
| | w/ $\text{ORE}_{\text{TCT}}$ | $^{GQ}_{B}$**0.364** | $^{GQ}_{B}$**0.206** | $_{B}$0.342 | $_{B}$**0.260** |

**Table 4: Mean re-ranking latency per query (in ms) using MonoT5 and RankLLaMA rerankers when the first-stage retrieval of different budgets ($c$) is done using BM25. The number of batches re-ranked ORE is enclosed in parentheses.**

| | MonoT5 | | | | | | RankLLaMA | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | time (ms/query) | | | Recall@c | | | time (ms/query) | | | Recall@c | | |
| c | GAR | QUAM | ORE | GAR | QUAM | ORE | GAR | QUAM | ORE | GAR | QUAM | ORE |
| 50 | 179.92 | 173.21 | 125.91(2) | 0.417 | 0.460 | 0.500 | 6269.77 | 6027.12 | 3925.54(2) | 0.421 | 0.449 | 0.492 |
| 100 | 356.53 | 328.82 | 272.04(4) | 0.539 | 0.594 | 0.594 | 12746.55 | 12074.67 | 7830.41(4) | 0.542 | 0.600 | 0.600 |
| 250 | 877.19 | 816.90 | 599.24(8) | 0.692 | 0.745 | 0.715 | 32312.97 | 30539.78 | 16092.16(8) | 0.684 | 0.761 | 0.719 |
| 1000 | 3418.98 | 3219.45 | 1848.26(8) | 0.836 | 0.874 | 0.827 | 127617.45 | 120885.39 | 16327.25(8) | 0.854 | 0.881 | 0.829 |
| | | | 2188.29(16) | | | 0.841 | | | 31939.78(16) | | | 0.853 |

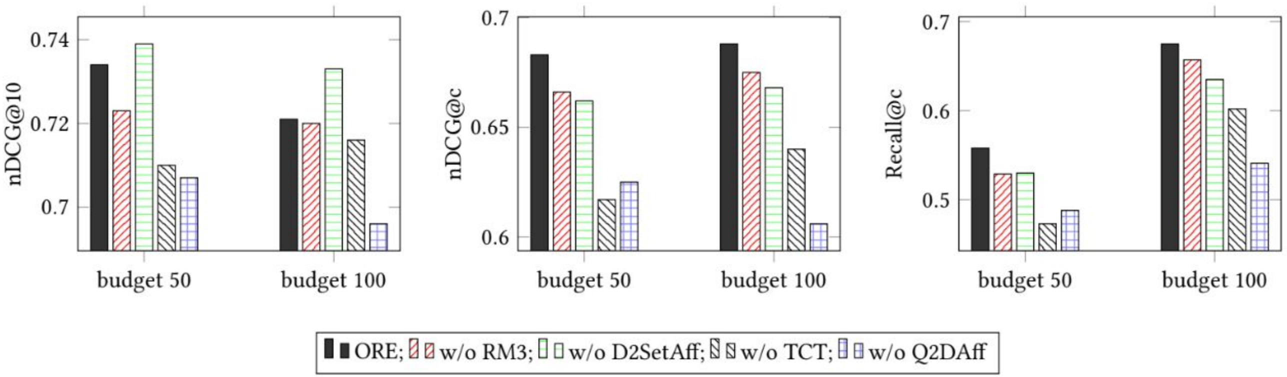



**Figure 6: Effect of different features on ORE's performance in Hybrid retrieval setting.**

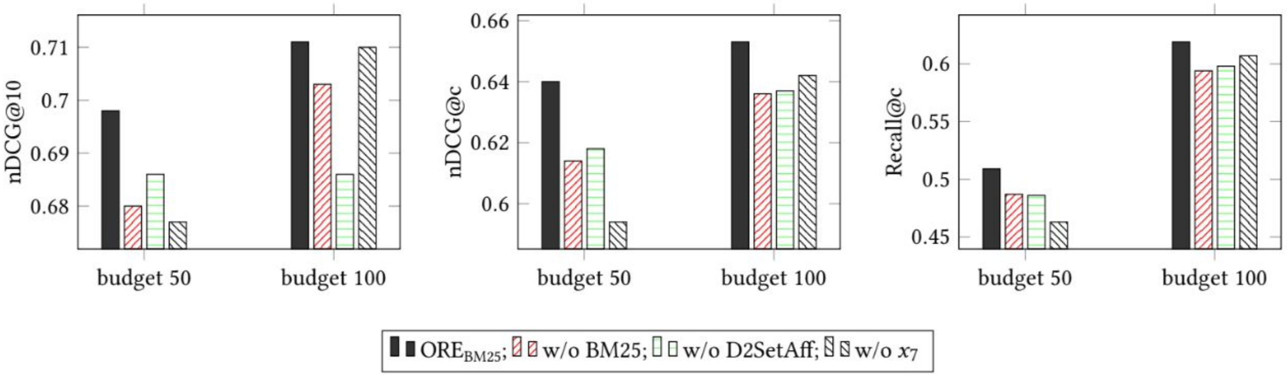
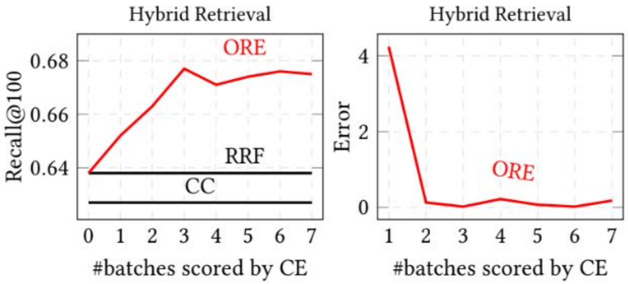
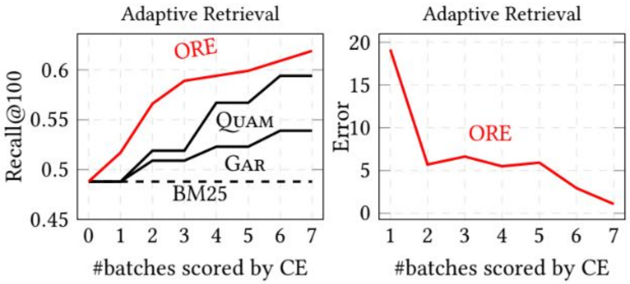



**Figure 7: Effect of different features on ORE's performance in Adaptive retrieval setting.**




**(a)** **(b)**

**Figure 3: Recall (left) and estimation error (right) comparison for hybrid retrieval setting on the TREC DL19 dataset when the number of batches of scored by cross-encoder (CE) varies for ORE for ranking budget of 100 and batch of size 16.**




**(a)** **(b)**

**Figure 4: Recall (left) and estimation error (right) comparison on the TREC DL19 dataset for adaptive retrieval, for ranking budget of 100 and batch of size 16.**