Leveraging Prior Relevance Signals in Web Search

CLEF 2024 - LongEval

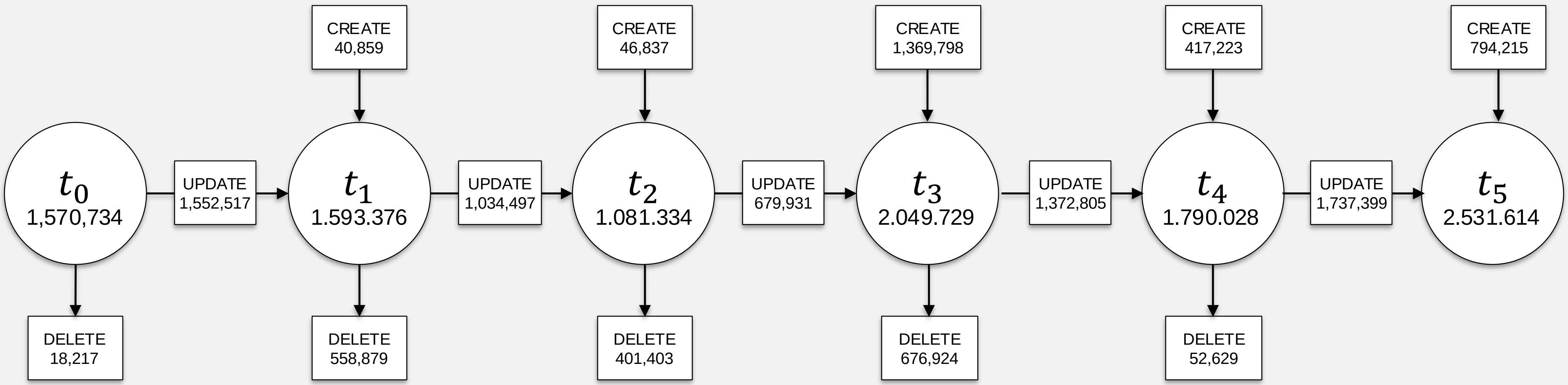
Jüri Keller, Timo Breuer, Philipp Schaer

24-12-11 – Grenoble, France
https://ir.web.th-koeln.de

Technology
Arts Sciences
TH Köln

Introduction

- The information landscape is ever evolving
- The LongEval test collection represent this
- but last year no system directly made use of it



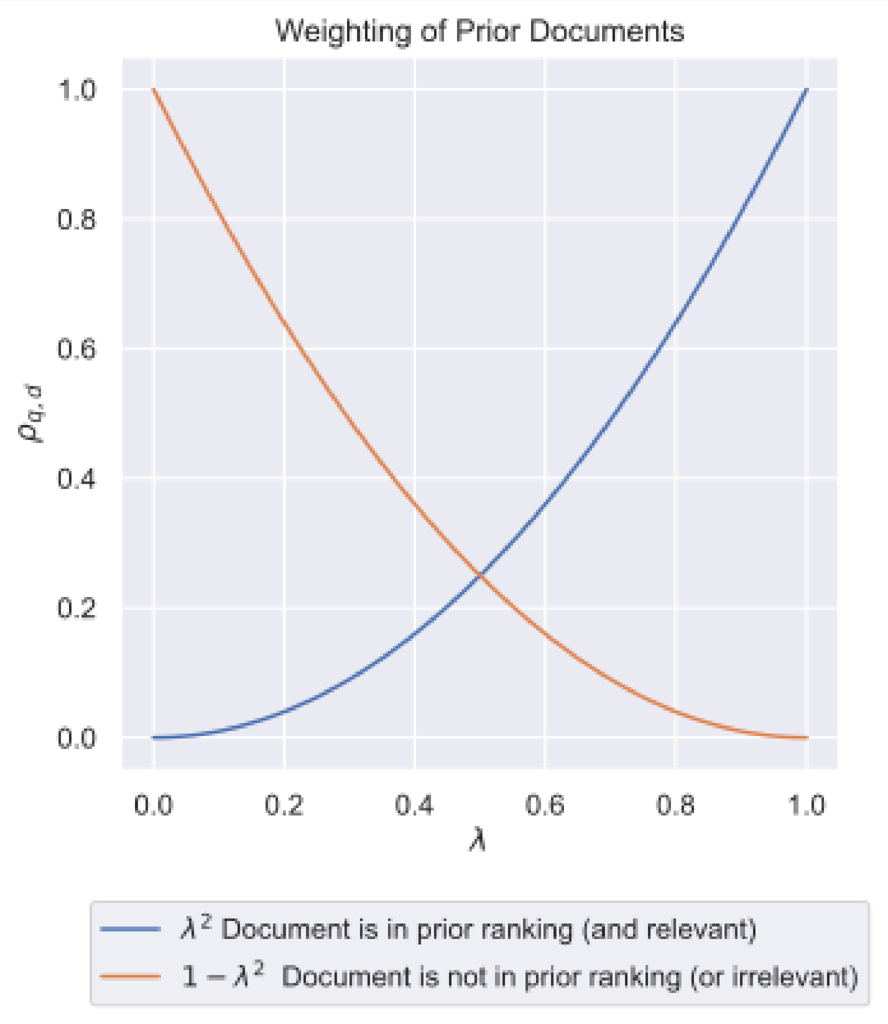
Method

- Hypothesis: Relevance stays

- Boost previously (relevant) documents

- $$\rho_{q,d}(\lambda) = \begin{cases} \lambda^2 & \text{if } d \in r_{q,t_{n-1}} \\ (1-\lambda)^2 & \text{otherwise} \end{cases}$$

- Baselines:
  - *cir_run_1:* **BM25**
  - *cir_run_2:* **BM25 + monoT5**



cir_run_5: BM25 + time boost

- Boost by time:
  - Relevant because **new to the ranking**
  - Relevant because **still in the collection**

- High fidelity of λ
- Grid search based on LT sub-collection from 2023
  - $\lambda = 0.503$
  - Slightly boost known documents

cir_run_3: **BM25 + qrel boost**

- Naive approach: Boost by relevant query – document pairs
- Only affects known query – document pairs
- Despite:
  - Change in documents
  - Change in topic
  - Data leakage?
- $\lambda = 0.7$
- History: $\{t_3, t_2, t_1, t_0\}$

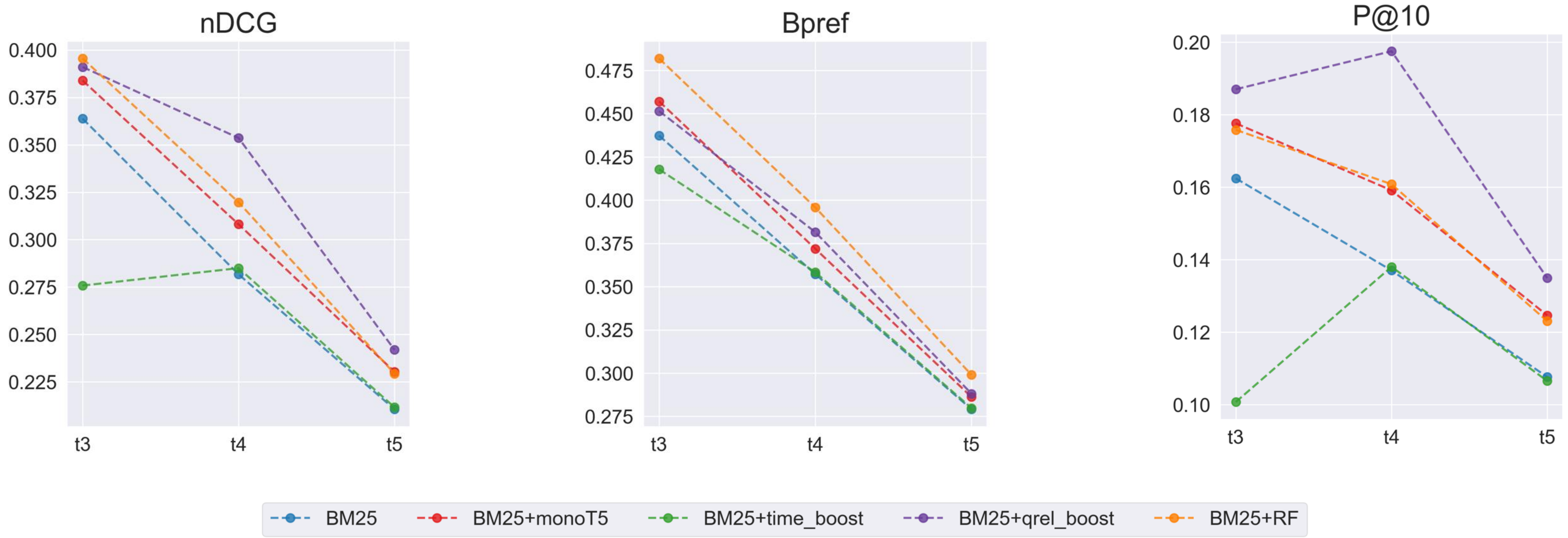
cir_run_4: **BM25 + RF**

- Generalize boosting based on prior click feedback across new docs

- Known queries:
  - History of relevance labels from the train split and last year's dataset
  - Construct vocabulary from relevant documents
  - Expand query with top 10 tf-idf terms

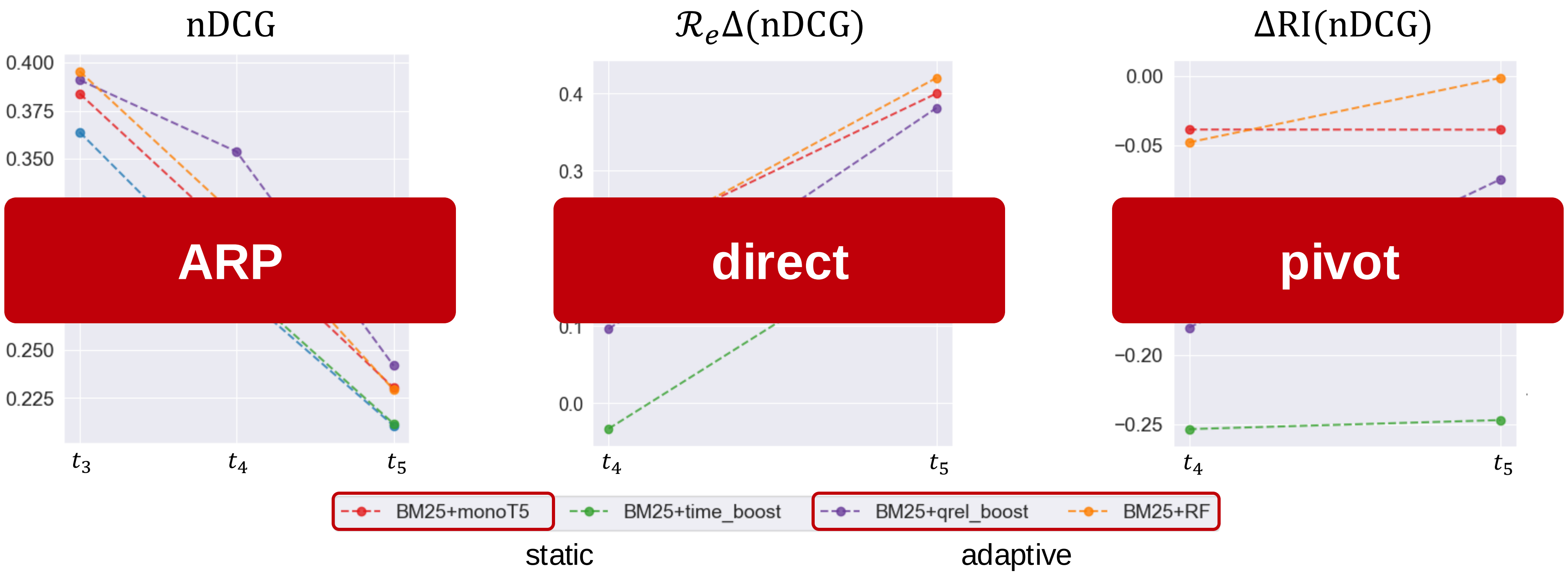
- New Queries:
  - BM25 + RM3

Results

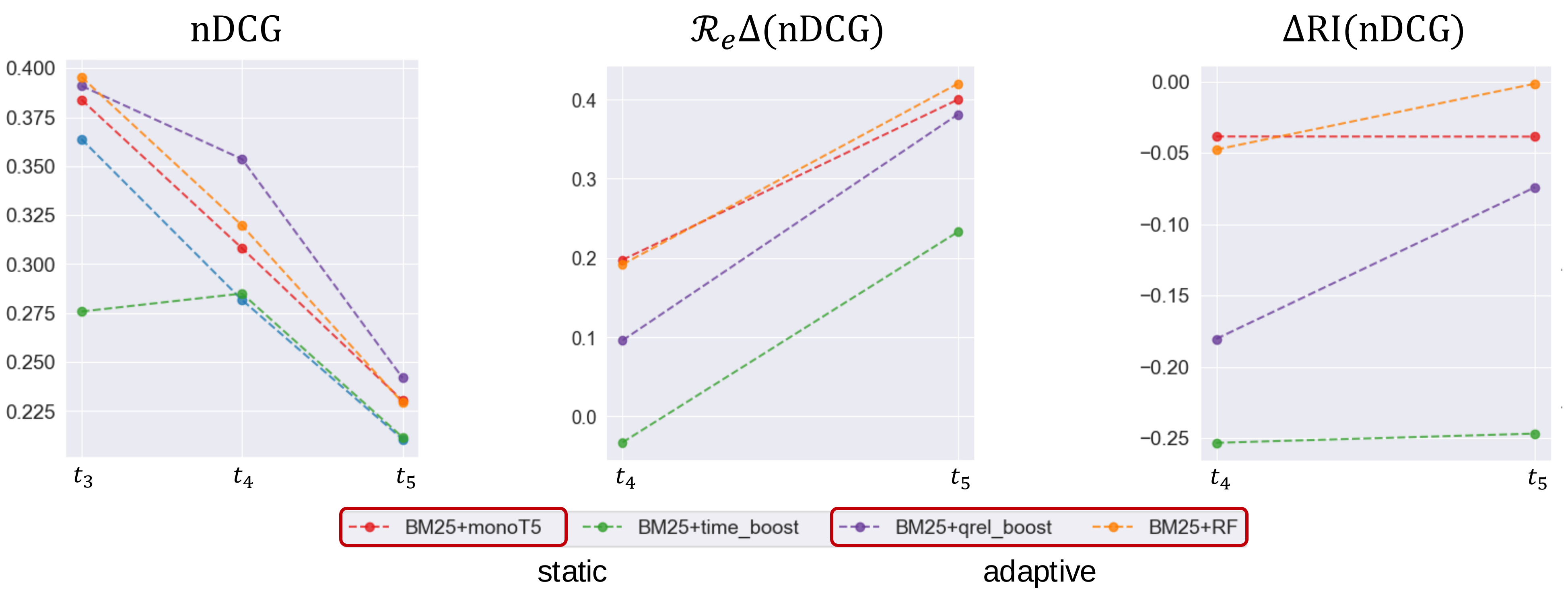


- Changes over time and measure
- Exploiting historic relevance feedback can outperform neural models
  - … at a much lower cost

Results


nDCG
0.400
0.375
0.350
0.250
0.225
t_3
t_4
t_5
ARP
$\mathcal{R}_e\Delta(\text{nDCG})$
0.4
0.3
0.1
0.0
t_4
t_5
direct
$\Delta\text{RI}(\text{nDCG})$
0.00
−0.05
−0.20
−0.25
t_4
t_5
pivot
BM25+monoT5
BM25+time_boost
BM25+qrel_boost
BM25+RF
static
adaptive

Results


nDCG
$\mathcal{R}_e\Delta$(nDCG)
ΔRI(nDCG)
0.400
0.375
0.350
0.325
0.300
0.275
0.250
0.225
t_3 t_4 t_5
0.4
0.3
0.2
0.1
0.0
t_4 t_5
0.00
−0.05
−0.10
−0.15
−0.20
−0.25
t_4 t_5
BM25+monoT5
BM25+time_boost
BM25+qrel_boost
BM25+RF
static
adaptive

cir_run_3: **BM25 + qrel boost**

- Naive approach: Boost by relevant query – document pairs
- Only affects known query – document pairs
- Despite:
  - Change in documents
  - Change in topic
  - Data leakage?

cir_run_3: **BM25 + qrel boost**

- Naive approach: Boost by relevant query – docume
- Only affects k
- Despit
  - Change in documents
  - Change in topic
  - Data leakage?

“worse robust system against changes”

cir_run_3: **BM25 + qrel boost**

- Naive approach: Boost by relevant query – docume
- C
- Despit
  - Change in documents
  - Change in topic
  - Data leakage?

inst changes”

Best English system at lag6 and 3rd best at lag8

What is robustness?

- Relative change in effectiveness:

> *“small RND values mean more robust systems against changes, and large RND values mean that the systems are not able to generalize well between lag6 and lag8”*

- Counterintuitive: An improving system would be robust?
- Should we optimize for it?

Conclusion

Results depend on the point in time

- Effectiveness changes
- due to the dataset

Relevance feedback is awesome

- Analysis of data leakage needed
- Validity of queries
- How can we exploit this safe

Wanted: Deep pools

- Excited for the relevance judgements
- Could explain observed effects better

Thank You!

CEUR Workshop Proceedings
ceur-ws.org