Bias in Learning to Rank Caused by Redundant Web Documents

Bachelor's Thesis Defence

Jan Heinrich Reimer

Martin Luther University Halle-Wittenberg
Institute of Computer Science
Degree Programme Informatik

June 3, 2020

Duplicates on the Web

Example

Figure: *The Beatles* article and duplicates on Wikipedia—identical except redirect

Redundancy in Learning to Rank

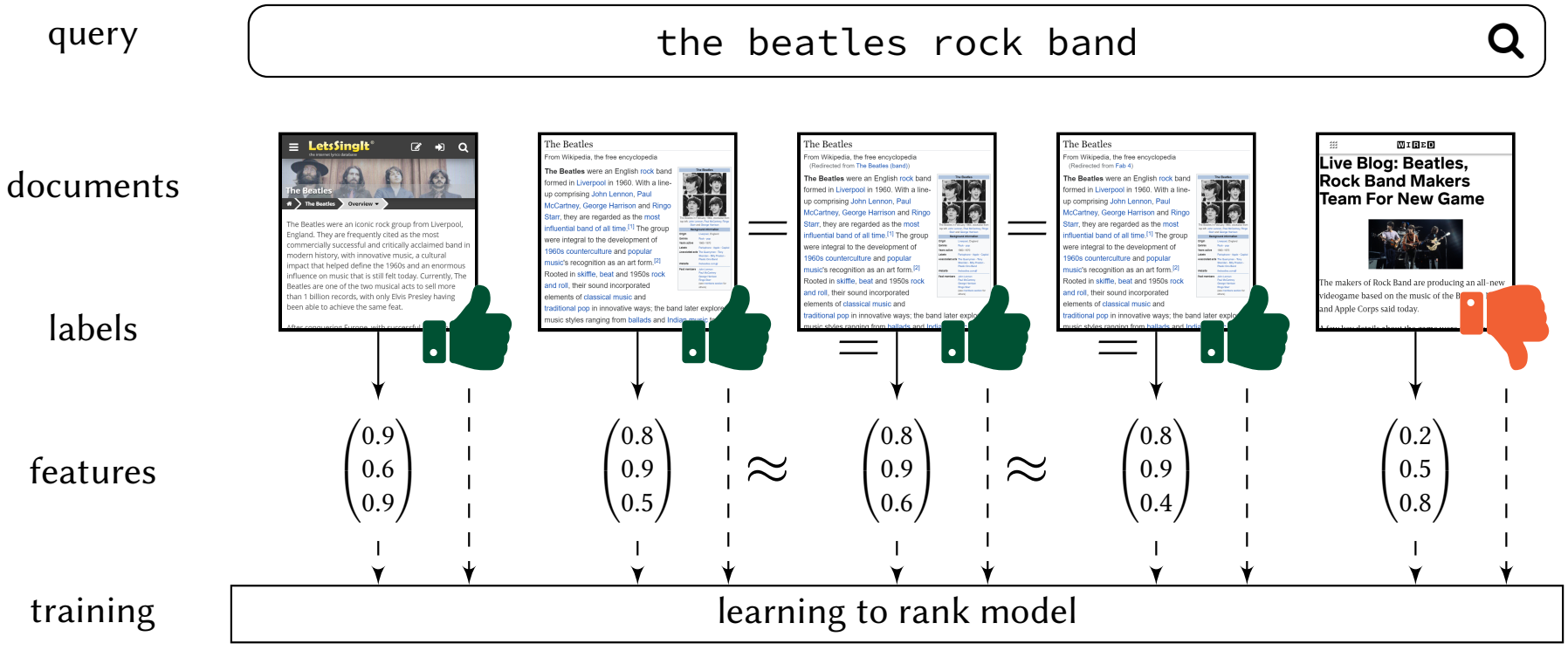


Figure: Training a learning to rank model

Problems

- identical relevance labels (Cranfield paradigm)
- similar features
- double impact on loss functions → overfitting

Duplicates in Web Corpora

- compare fingerprints/hashes of documents, e.g., word n-grams
  - syntactic equivalence
  - near-duplicate pairs form groups
- 20 % duplicates in web crawls, stable in time [Bro+97; FMN03]
  - up to 17 % duplicates in TREC test collections [BZ05; Frö+20]
- few domains make up for most near duplicates
  - redundant domains often popular
- canonical links to select representative [OK12],
  e.g., *Beatles* $\rightarrow$ *The Beatles*
  - if no link assert self-link, then choose most often linked
  - resembles authors' intent

Learning to Rank

- machine learning + search result ranking
- combine predefined features [Liu11, p. 5],
  e.g., retrieval scores, BM25, URL length, click logs, …
- standard approach for ranking: rerank top-k results
  from conventional ranking function
- prone to imbalanced training data

Approaches

pointwise predict ground truth label for single documents

pairwise minimize inconsistencies in pairwise preferences

listwise optimize loss function ranked lists

Learning to Rank Pipeline

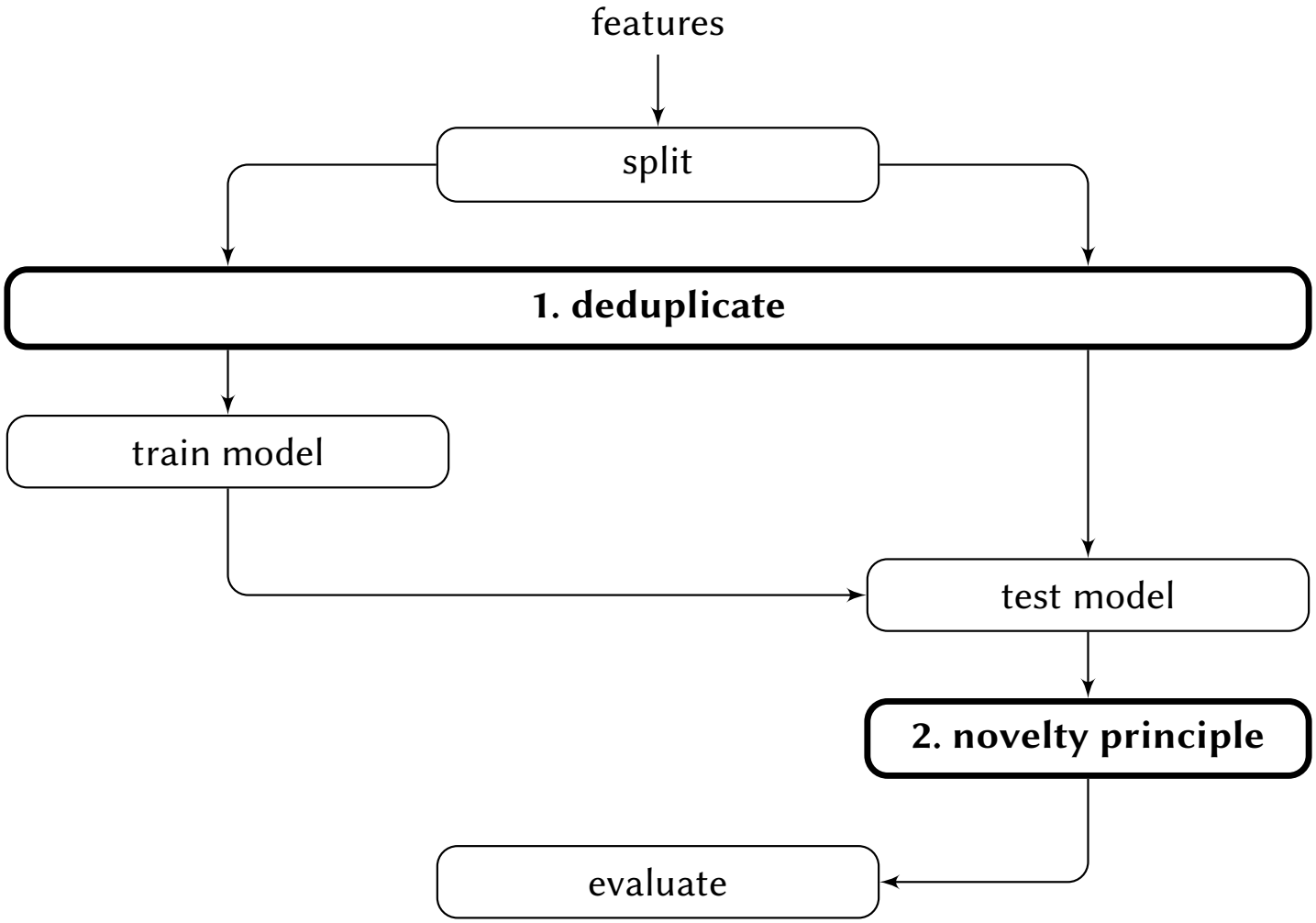


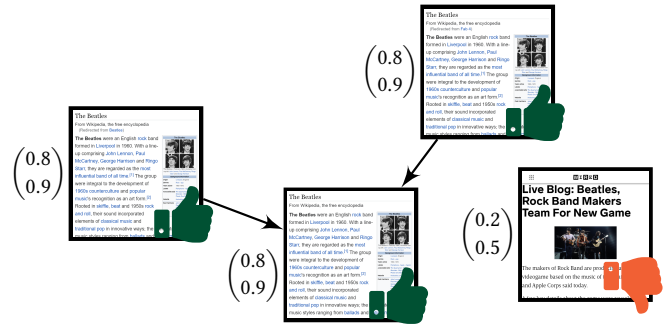
Figure: Novelty awareLlearning to rank pipeline for evaluation

Deduplication of Feature Vectors

- reuse methods for counteracting overfitting → undersampling
- active impact on learning
- deduplicate train/test sets separately

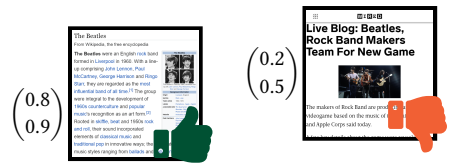
Full redundancy (100 %)

- use all documents for training
- baseline



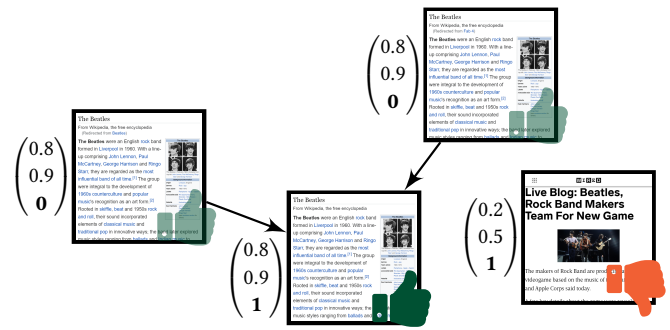
No redundancy (0 %)

- remove non-canonical documents
- algorithms can't learn about non-canonical documents



Novelty-aware penalization (*NOV*)

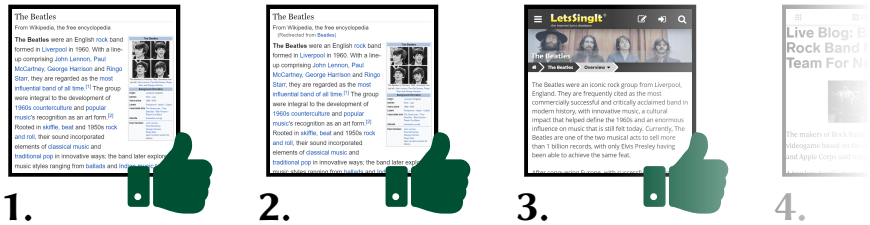
- discount non-canonical documents' relevance
- add flag feature for most canonical document



Novelty Principle [BZ05]

- deduplication of search engine results
- users don't want to see the same document twice

Duplicates unmodifed

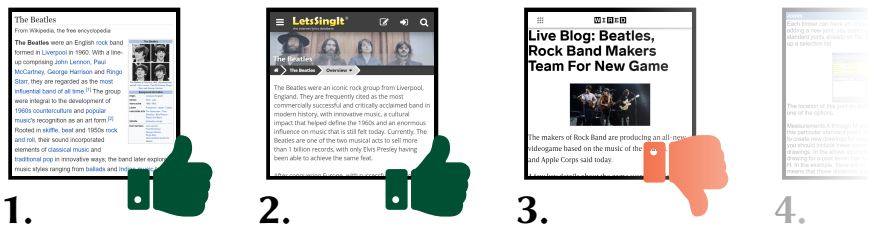


overestimates performance [BZ05]

Duplicates irrelevant

users still see duplicates

Duplicates removed



no redundant content
$\rightarrow$ most realistic

Learning to Rank Datasets

Table: Benchmark datasets

Year	Name	Duplicate detection	Queries	Docs. / Query
2008	LETOR 3.0 [Qin+10]	✗	681	800
2009	LETOR 4.0 [QL13]	✓	2.5K	20
2011	Yahoo! LTR Challenge [CC11]	✗	36K	20
2016	MS MARCO [Ngu+16]	✓	100K	10
2020	our dataset	✓	200	350

- duplicate detection only possible for LETOR 4.0 and MS MARCO
- shallow judgements in existing datasets
- create new deeply judged dataset from TREC Web '09–'12
- worst-/average-case train/test splits for evaluation

Evaluation

- train & rerank common learning-to-rank models:
  regression, RankBoost [Fre+03], LambdaMART [Wu+10],
  AdaRank [XL07], *Coordinate Ascent* [MC07], ListNET [Cao+07]
- settings: no hyperparameter tuning, no regularization, 5 runs
- remove BM25 = 0 (selection bias in LETOR [MR08])
- BM25@body baseline for comparison

Experiments

- retrieval performance / nDCG@20 [JK02]
- ranking bias / rank of irrelevant duplicates
- fairness of exposure [Bie+20]

Retrieval Performance on ClueWeb09

Evaluation with Deep Judgements

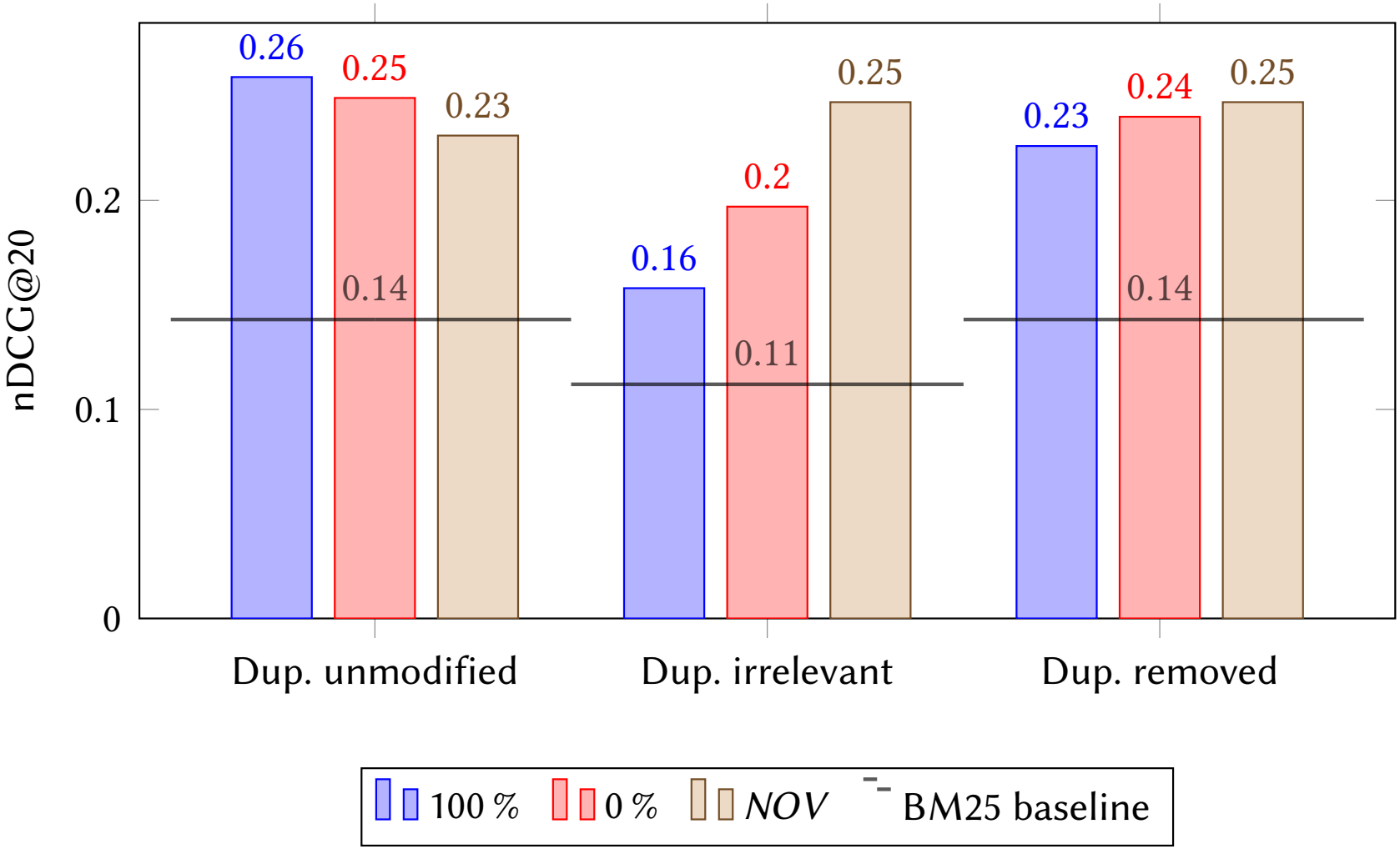


Figure: nDCG@20 performance for ClueWeb09, with Coordinate Ascent

Retrieval Performance on GOV2

Evaluation with Shallow Judgements

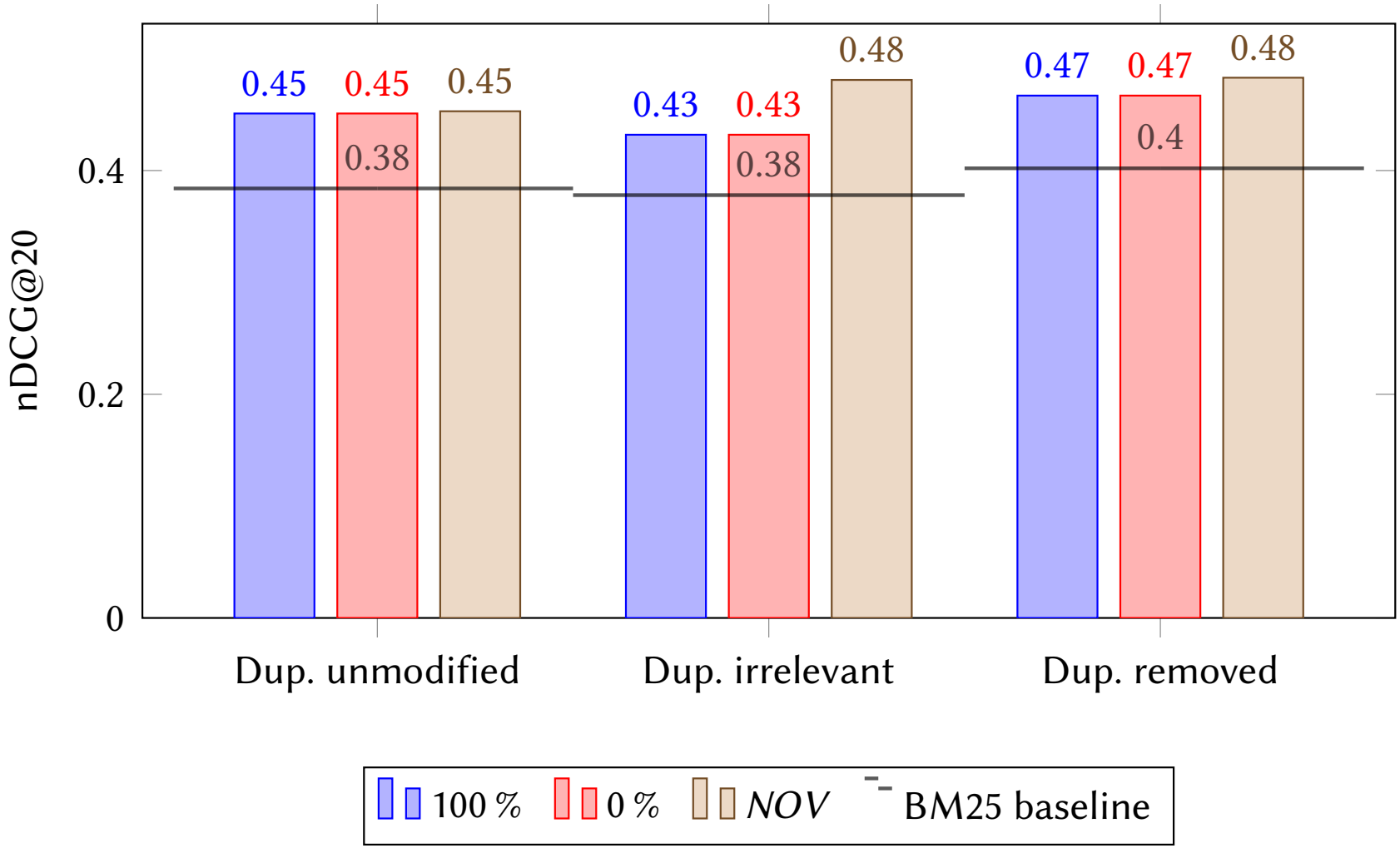


Figure: nDCG@20 performance for GOV2, with AdaRank

Retrieval Performance

Evaluation

- performance decreases by up to 39 % under novelty principle
- improvement with penalization of duplicates,
  compensates novelty principle impact
- significant changes only for some algorithms,
  mostly when duplicates irrelevant
- slightly decreased performance when deduplicating
  without novelty principle
- all learning to rank models better than BM25 baseline

Ranking Bias on ClueWeb09

Evaluation with Deep Judgements

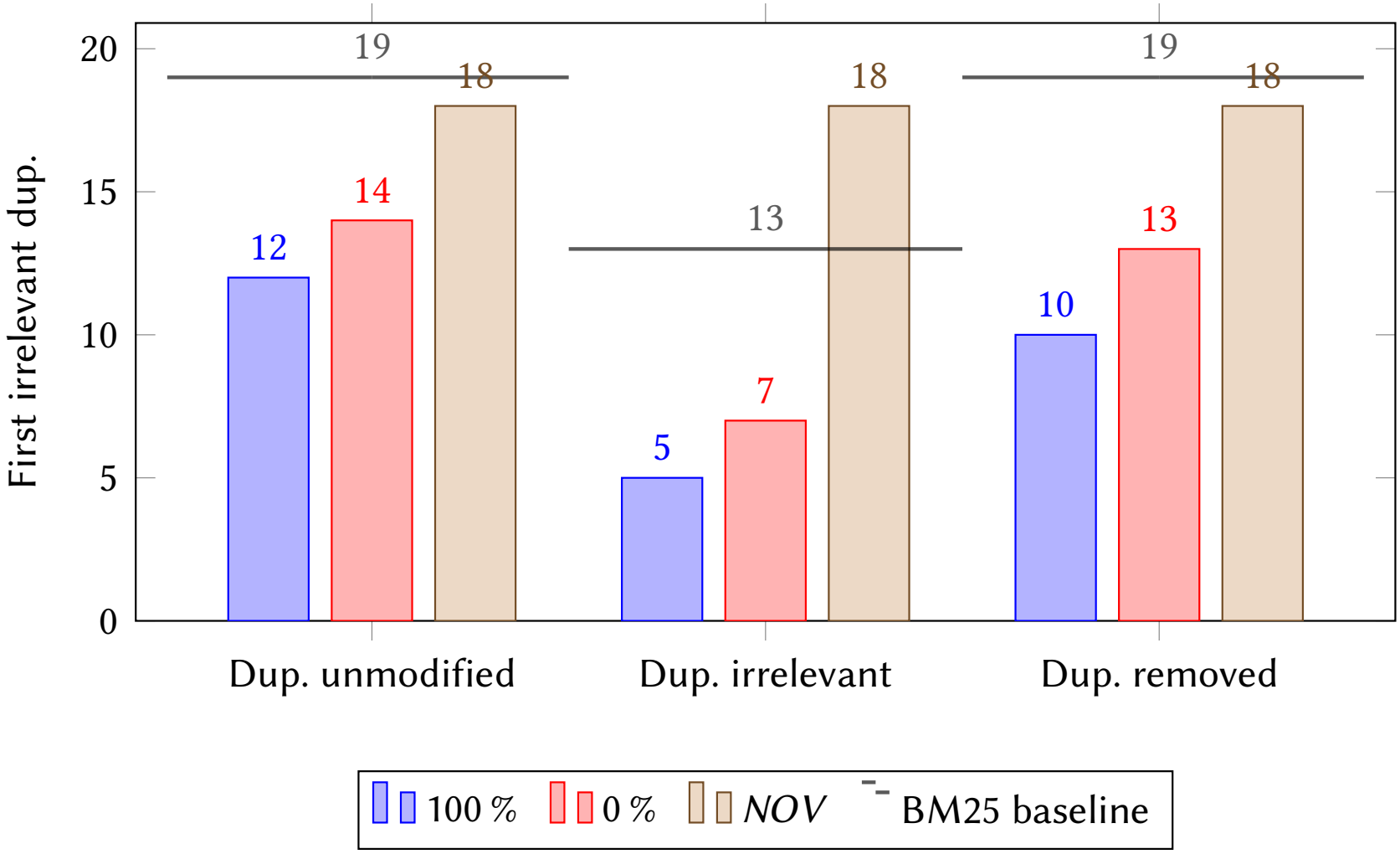


Figure: First irrelevant duplicate rank for ClueWeb09, with Coordinate Ascent

Ranking Bias on GOV2

Evaluation with Shallow Judgements

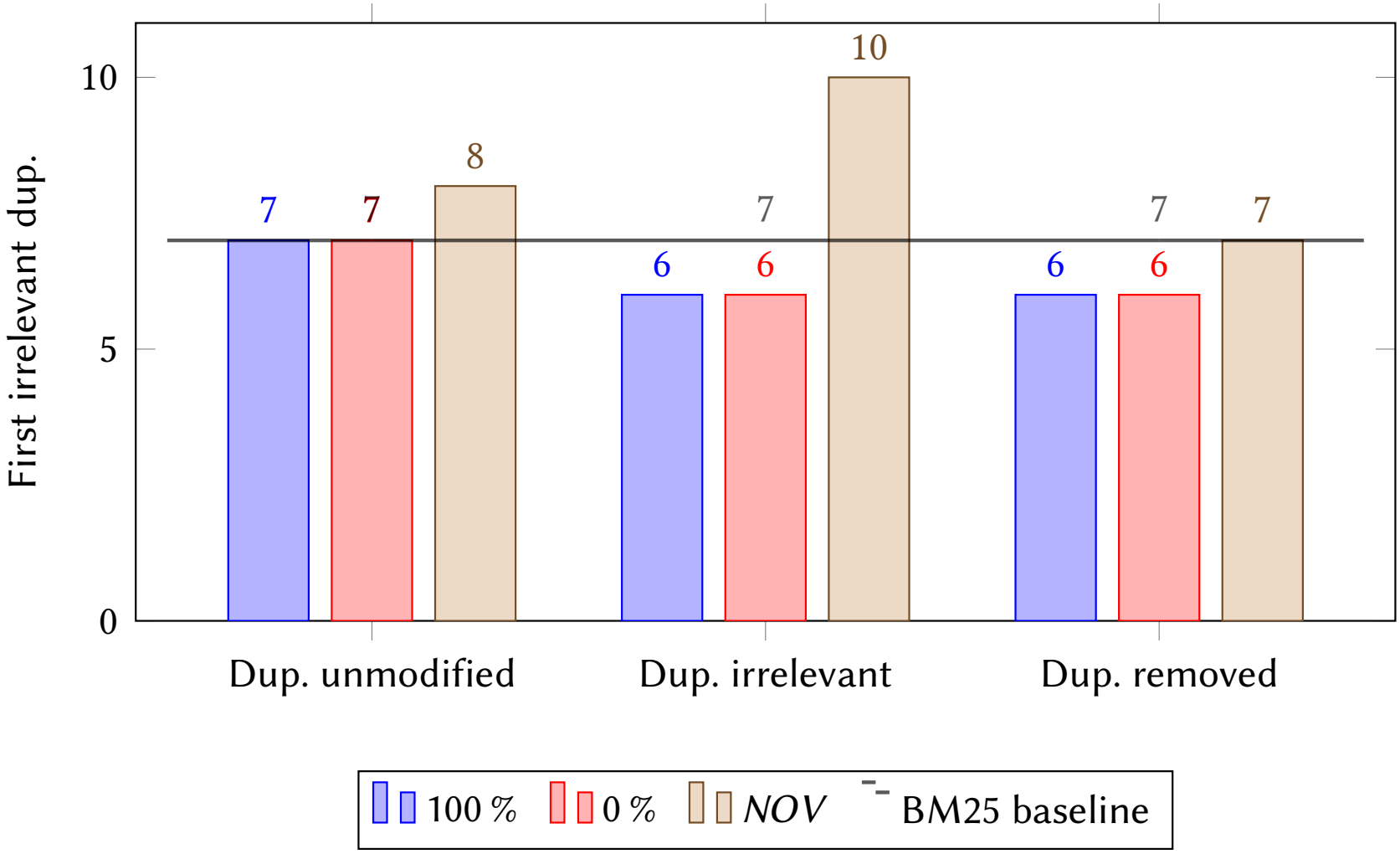


Figure: First irrelevant duplicate rank for GOV2, with AdaRank

Ranking Bias

Evaluation

- irrelevant duplicates ranked higher under novelty principle, often top-10
- bias towards duplicate content
- removing/penalizing duplicates counteracts bias significantly
- more biased than BM25 baseline
- implicit popularity bias as redundant domains are most popular
- poses risk at search engines using learning to rank

Fairness of Exposure [Bie+20]

Evaluation

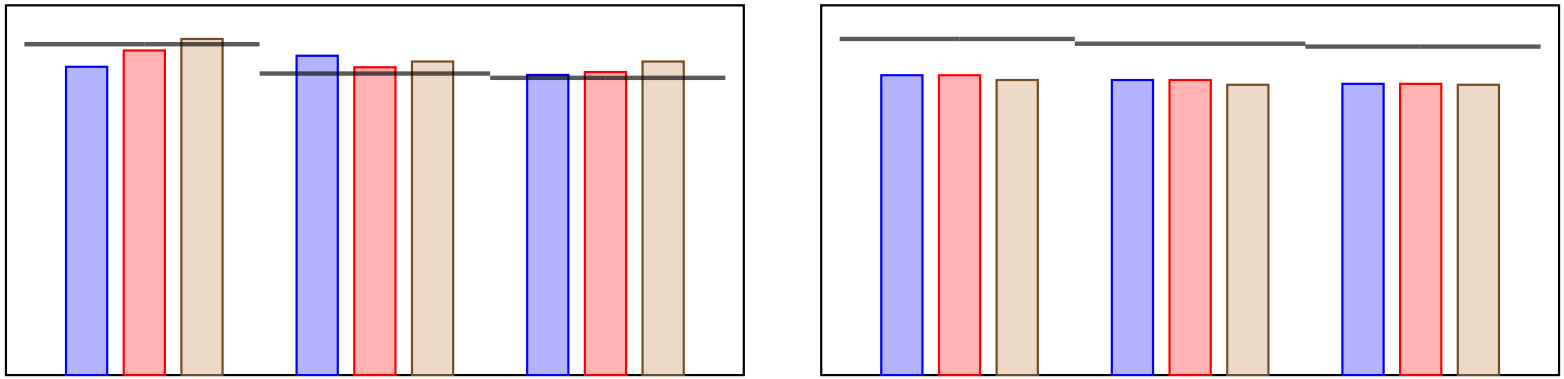


Figure: Fairness of exposure for ClueWeb09 and GOV2

- no significant effects
- fairness measures unaware of duplicates
- duplicates should count for exposure, not for relevance
- tune Biega's parameters $\rightarrow$ trade-off fairness vs. relevance [Bie+20]
- experiment with other fairness measures

Conclusion

- near-duplicates present in learning-to-rank datasets
  - reduce retrieval performance
  - induce bias
  - don't affect fairness of exposure
- novelty principle for measuring impact
- deduplication to prevent

Future Work

- direct optimization [Xu+08] of novelty-aware metrics [Cla+08]
- reflect redundancy in fairness of exposure
- experiments on more datasets (e.g., Common Crawl) and more algorithms (e.g., deep learning)
- detect & remove vulnerable features

Thank you!

Bibliography

- Bernstein, Yaniv et al. (2005). "Redundant documents and search effectiveness." In: CIKM '05. ACM, pp. 736–743.
- Biega, Asia J. et al. (2020). "Overview of the TREC 2019 Fair Ranking Track." In: arXiv: 2003.11650.
- Broder, Andrei Z. et al. (1997). "Syntactic Clustering of the Web." In: *Comput. Networks* 29.8–13, pp. 1157–1166.
- Cao, Zhe et al. (2007). "Learning to rank: from pairwise approach to listwise approach." In: ICML '07. Vol. 227. International Conference Proceeding Series. ACM, pp. 129–136.
- Chapelle, Olivier et al. (2011). "Yahoo! Learning to Rank Challenge Overview." In: Yahoo! Learning to Rank Challenge. Vol. 14. Proceedings of Machine Learning Research, pp. 1–24.
- Clarke, Charles L. A. et al. (2008). "Novelty and diversity in information retrieval evaluation." In: SIGIR '08. ACM, pp. 659–666.
- Fetterly, Dennis et al. (2003). "On the Evolution of Clusters of Near-Duplicate Web Pages." In: *Empowering Our Web*. LA-WEB 2003. IEEE, pp. 37–45.
- Freund, Yoav et al. (2003). "An Efficient Boosting Algorithm for Combining Preferences." In: *J. Mach. Learn. Res.* 4, pp. 933–969.
- Fröbe, Maik et al. (2020). "The Effect of Content-Equivalent Near-Duplicates on the Evaluation of Search Engines." In: *Advances in Information Retrieval*. ECIR 2020. Springer, pp. 12–19.
- Järvelin, Kalervo et al. (2002). "Cumulated gain-based evaluation of IR techniques." In: *ACM Trans. Inf. Syst.* 20.4, pp. 422–446.
- Liu, Tie-Yan (2011). *Learning to Rank for Information Retrieval*. 1st ed. Springer.
- Metzler, Donald et al. (2007). "Linear feature-based models for information retrieval." In: *Inf. Retr. J.* 10.3, pp. 257–274.

Bibliography (cont.)

- Minka, Tom et al. (2008). "Selection bias in the LETOR datasets." In: LR4IR 2008, pp. 48–51.
- Nguyen, Tri et al. (2016). "MS MARCO: A Human Generated MAchine Reading COmprehension Dataset." In: CoCo 2016. Vol. 1773. CEUR Workshop Proceedings. Sun SITE Central Europe.
- Ohye, Maile et al. (Apr. 2012). *The Canonical Link Relation*. RFC 6596.
- Qin, Tao et al. (2010). "LETOR: A benchmark collection for research on learning to rank for information retrieval." In: *Inf. Retr. J.* 13.4, pp. 346–374.
- Qin, Tao et al. (2013). "Introducing LETOR 4.0 Datasets." In: arXiv: 1306.2597.
- Wu, Qiang et al. (2010). "Adapting boosting for information retrieval measures." In: *Inf. Retr. J.* 13.3, pp. 254–270.
- Xu, Jun et al. (2007). "AdaRank: a boosting algorithm for information retrieval." In: SIGIR '07. ACM, pp. 391–398.
- Xu, Jun et al. (2008). "Directly optimizing evaluation measures in learning to rank." In: SIGIR '08. ACM, pp. 107–114.

Wikipedia Bias on ClueWeb09

Evaluation with Deep Judgements

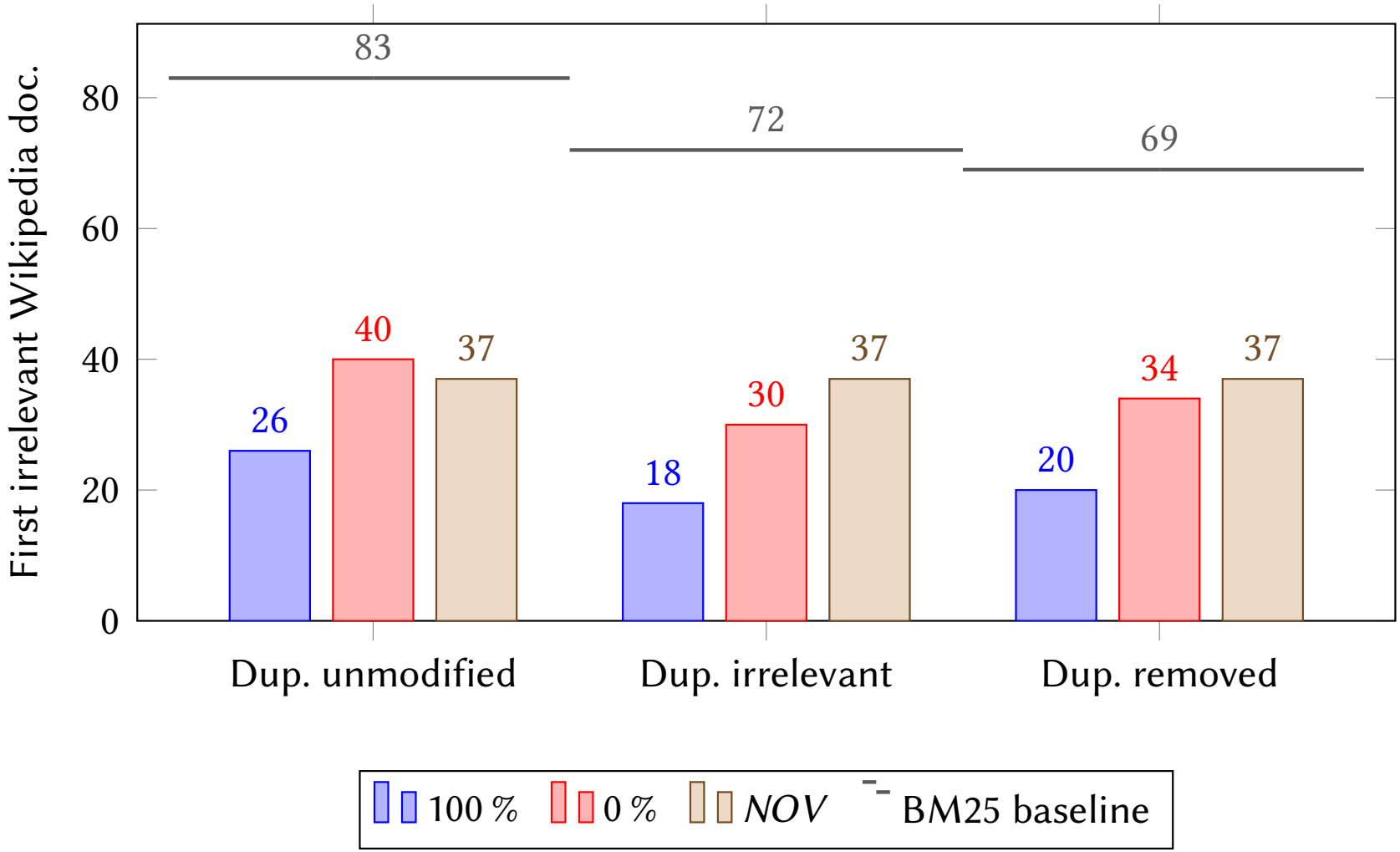


Figure: First irrelevant Wikipedia rank for ClueWeb09, with Coordinate Ascent

Fairness of Exposure on ClueWeb09 [Bie+20]

Evaluation with Deep Judgements

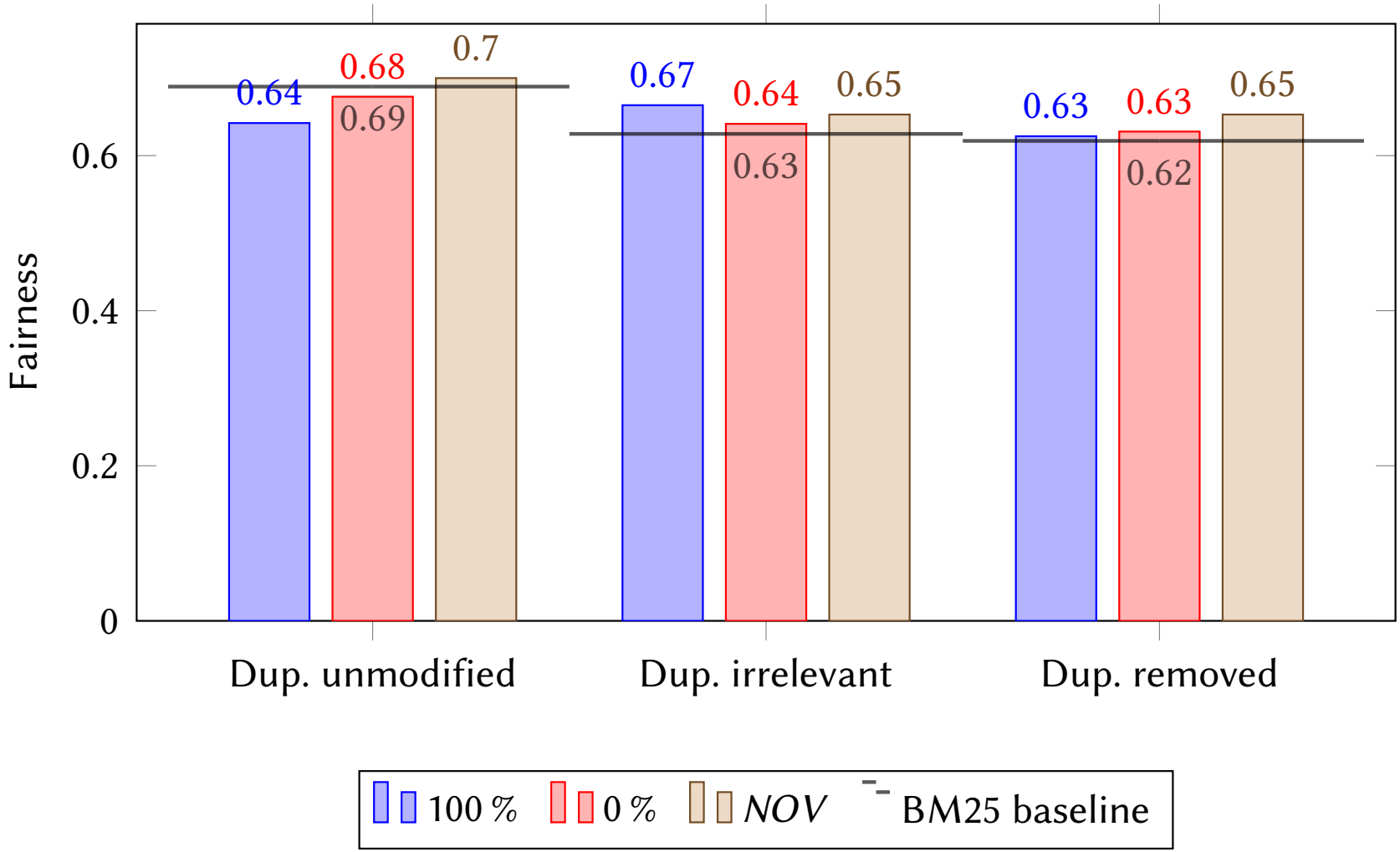


Figure: Fairness of exposure for ClueWeb09, with Coordinate Ascent

Fairness of Exposure on GOV2 [Bie+20]

Evaluation with Shallow Judgements

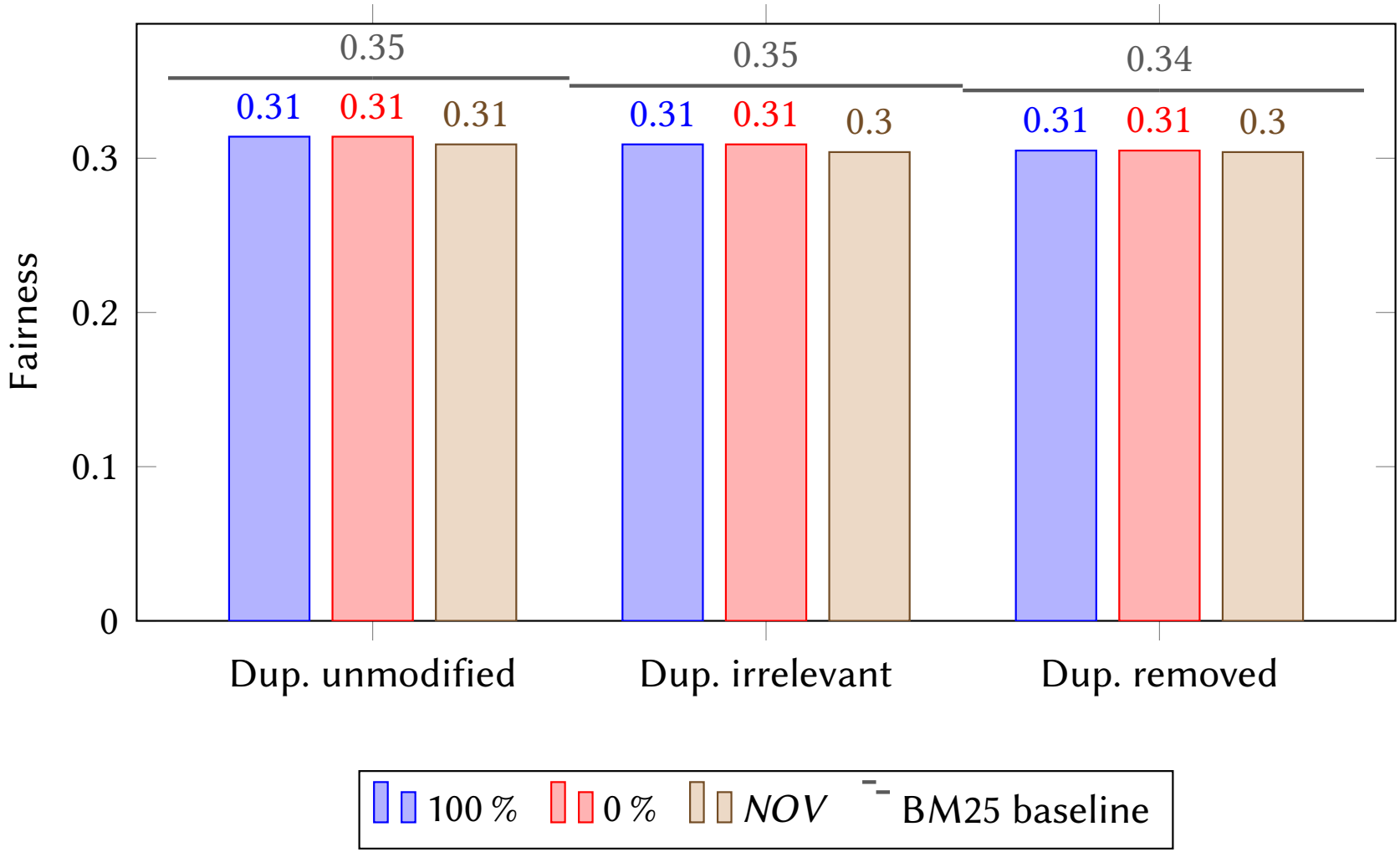


Figure: Fairness of exposure for GOV2, with AdaRank