Weighted AUReC: Handling Skew in Shard Map Quality Estimation for Selective Search

Gijs Hendriksen, Djoerd Hiemstra, and Arjen P. de Vries

Radboud University, Nijmegen, The Netherlands
{gijs.hendriksen,djoerd.hiemstra,arjen.devries}@ru.nl

Abstract. In selective search, a document collection is partitioned into a collection of topical index shards. To efficiently estimate the topical coherence (or quality) of a shard map, the AUReC measure was introduced. AUReC makes the assumption that shards are of similar sizes, one that is violated in practice, even for unsupervised approaches. The problem might be amplified if supervised labelling approaches with skewed class distributions are used. To estimate the quality of such unbalanced shard maps, we introduce a weighted adaptation of the AUReC measure, and empirically evaluate its effectiveness using the ClueWeb09B and Gov2 datasets. We show that it closely matches the evaluations of the original AUReC when shards are similar in size, but captures better the differences in performance when shard sizes are skewed.

Keywords: selective search · clustering · evaluation · cluster-based retrieval

1 Introduction

With increasingly complex and expensive retrieval pipelines, efficient retrieval over large document collections can be quite difficult to achieve. Distributed systems may alleviate part of this problem by partitioning the index into shards and searching these in parallel. Another approach, *selective search* [19], avoids the exhaustive search over the full collection, reducing the total number of documents processed per query by partitioning the collection into topically coherent shards. This assumes the Cluster Hypothesis [25] to be true, that a query's relevant documents are allocated to the same (subset of) shards. The *resource selection algorithm* should select the relevant shards for each incoming query.

Previous work in selective search has investigated end-to-end effectiveness [19], runtime efficiency [16,17], partitioning strategies [12,18], resource selection algorithms [1,11,13,20,22,24], and robustness [10]. The AUReC (Area Under Recall Curve) measure [15] estimates *shard map quality*, considered high if, for any query, its relevant documents are indeed clustered in a few shards only. The AUReC measure circumvents the need for manual relevance assessments by marking the top documents retrieved by a strong ranker as pseudo-relevant.

Unfortunately, AUReC makes the assumption that shards have similar size, such that the top k shards can be returned. However, existing clustering algorithms cannot guarantee that the resulting shards have similar sizes; in fact, the shard maps on which AUReC was evaluated originally, generated by Dai et al. [12], also contain shards that are orders of magnitude larger than the smallest ones – even after applying size-bounded clustering [19]. In her dissertation, Kim [14, Section 8.2] already observed that AUReC could be biased towards unbalanced shard maps.

The situation can be expected to become much worse when these shard maps distribute Web documents by language, top level domain or category. Especially category assignments by a supervised classifier like in the ClueWeb22 corpus [23] are likely to follow a Zipfian distribution. In this case, splitting and merging shards to balance the shard sizes may not be desirable, as it might decrease the coherence and interpretability of affected shards. To mitigate this bias, we introduce a weighted variant of the AUReC that takes shard size into account. We show empirically how this *weighted AUReC* is a better measure of shard map quality when these shard maps exhibit skew.

We also consider a budget-constrained situation, where the system processes a fixed number of *documents* per query instead of a fixed number of shards. In this case, it may be a good strategy to select many small shards, instead of the few large shards that AUReC is biased towards. We show that weighted AUReC measures shard map quality also more accurately in this budget-constrained setting. The code used for our experiments is published to GitLab.[1]

2 Weighted AUReC

The AUReC measure [15] builds on the underlying goal of a selective search system: retrieving the same documents as an exhaustive search system, but more efficiently. As such, a strong ranker can be used to exhaustively retrieve the top k documents D_q for a given query q (usually, $k = 1000$). These documents are then marked as pseudo-relevant for the calculation of the AUReC.

Assume a shard map p with n_p shards. For each query q, let $count(D_q, s_i^p)$ be the number of documents from D_q that appear in shard s_i^p. Define a relevance-based ranking (RBR) order of shards, such that each shard s_i^p contains more pseudo-relevant documents from D_q than the next one. Formally:

$$count(D_q, s_i^p) \geq count(D_q, s_{i+1}^p) \qquad \text{for all } i \in \{1 \dots n_p - 1\}$$

Given this ordering, a recall-like measure $R_q(p, k)$ is defined to measure the percentage of pseudo-relevant documents that appear in the first k shards of shard map p[2]:

$$R_q(p, k) = \frac{1}{|D_q|} \sum_{i=1}^{k} count(D_q, s_i^p) \qquad \text{for } k \in \{0 \dots n_p\}$$

[1] https://gitlab.science.ru.nl/informagus/weighted-aurec

[2] We use a simplified version of the formula from Kim and Callan [15], in which we assume that D_q is never empty. Other than that, the formulas are equivalent.

Using this definition, the recall curve is formed by the points $\langle k/n_p,\ R_q(p,k)\rangle$ for all $k \in \{0 \dots n_p\}$. The AUReC for query q is the area under this curve:

$$AUReC_q(p) = \frac{1}{2n_p} \sum_{k=0}^{n_p-1} \big(R_q(p,k) + R_q(p,k+1)\big)$$

To obtain a final quality measurement, we average the $AUReC_q$ over all queries in the query set. AUReC scores range from 0.5 (relevant documents are uniformly distributed) to 1.0 (relevant documents are clustered together).

We will now adjust this measure to handle skewed shard maps, in two steps.

First, instead of ordering the shards by the number of documents from D_q they contain, we simply divide that number by the size of each shard. Formally:

$$count(D_q, s_i^p)/|s_i^p| \geq count(D_q, s_{i+1}^p)/|s_{i+1}^p| \qquad \text{for all } i \in \{1 \dots n_p - 1\}$$

This modification promotes smaller shards with a relatively large proportion of pseudo-relevant documents while pushing back large shards with a higher proportion of irrelevant documents.

Note that this has no impact on the definitions of $count(D_q, s_i^p)$ and $R_q(p,k)$; they are applied the same way, only for a different ordering.

Second, we scale each segment of the recall curve by the size of the corresponding shard. In other words, if D is the full collection, we define the recall curve as the points $\langle \sum_{i=1}^{k} |s_i^p| \ / \ |D|,\ R_q(p,k)\rangle$ for all $k \in \{0 \dots n_p\}$.

Weighted AUReC (wAUReC) follows as the area beneath this adjusted curve:

$$wAUReC_q(p) = \sum_{k=0}^{n_p-1} \frac{|s_{k+1}^p|}{2\sum_{i=1}^{n_p} |s_i^p|} \cdot \big(R_q(p,k) + R_q(p,k+1)\big)$$

When documents are distributed evenly across shards (i.e., every shard has the same size), the value of wAUReC equals the normal AUReC. Therefore, wAUReC is not only applicable in the case of a skewed shard map; it can be used as a full substitute for the normal AUReC.

3 Experimental setup

3.1 Documents, queries and runs

To empirically evaluate the effectiveness of the wAUReC and compare it to the normal AUReC, we ran a set of experiments with a similar setup to Kim and Callan [15]. We used the same document collections: Gov2[3] and ClueWeb09B[4]. We used the topics and relevance assessments provided by the TREC Terabyte Track from 2004 until 2006 [3,4,9] for evaluation on Gov2, and the TREC Web Track from 2009 until 2012 [5,6,7,8] for evaluation on ClueWeb09B. Finally, we

[3] http://ir.dcs.gla.ac.uk/test_collections/access_to_data.html

[4] https://lemurproject.org/clueweb09/

also used SlideFuse-MAP [2,21] to fuse the top 10 runs submitted each year to obtain the results of a 'strong ranker'. This fusion run was used both for gathering D_q for the AUReC measures and for the evaluation of an end-to-end selective search system using different resource selection algorithms.

Since we use topics, relevance judgments and submitted runs from TREC, we do not have to index the corpus ourselves and run a retrieval system against it. This makes our experimental setting easy to setup and replicate.

3.2 Shard maps and shard selection

We reuse the 6 shard maps (3 per dataset) generated by Dai et al. [12] for a basic comparison between the AUReC and wAUReC.[5]

Because these shard maps are fairly balanced in terms of shard size, the differences between the AUReC and its weighted variant might not become fully apparent. To make the problem more pronounced, we therefore generate extra shard maps, using different size distributions: uniform, linear and quadratic. For each collection and distribution type, we generate 10 different shard maps (60 in total). Per query, we distribute the relevant documents randomly over up to 10 shards. We repeat this procedure 50 times per shard map (resulting in 3000 shard maps), which allows us to perform a robust comparison between the two AUReC variants on unbalanced shard maps.

Evidently, these simulated random shard maps are unlikely to be used in practice, and lose the topical coherence that real-world clustering algorithms provide. However, we evaluate the end-to-end selective search performance using oracle resource selection algorithms only, meaning we can still evaluate whether relevant documents are clustered together – both for the end-to-end performance and for the AUReC and wAUReC. As such, the random shard maps are still useful to demonstrate the advantages of wAUReC, even if experiments with more realistic distribution approaches are warranted in future research.

Relevance-based shard ranking Kim and Callan [15] evaluated AUReC on selective search systems using different resource selection algorithms: Rank-S [20], Taily [1], and relevance-based ranking (RBR). RBR is the oracle that provides the theoretically maximum performance, selecting the shards with the highest number of relevant documents at static cutoff k. We only use RBR for our comparison between AUReC and wAUReC (for cutoffs $k \in \{1, 3, 5\}$).

Budget-based shard ranking Like AUReC, RBR assumes that shard sizes are balanced, as it orders the shards based on the absolute number of relevant documents they contain. As a result, the RBR oracle method might not showcase the limitations of the AUReC when it comes to unbalanced shard maps: they follow the same shard ordering. To illustrate the difference between AUReC and wAUReC more clearly, we also consider an alternate setting.

[5] Downloaded from https://boston.lti.cs.cmu.edu/appendices/CIKM2016-Dai/

When shard maps are skewed, selecting a static number of shards may not suffice. Consider a system with a maximum number of documents to process, e.g. in order to keep latency below a certain threshold or limit the allocated resources per query. We call this number the *budget* of the system.

With a budget of 1000 documents, one can either search one shard with 1000 documents, or 10 shards with 100 documents. The larger shard may have a larger absolute number of relevant documents, but the smaller shards combined may contain even more relevant documents. Relevance-based ranking will always return the largest shard first, even if this would result in sub-optimal results.

We therefore introduce *budget-based ranking* (BBR), an alternative for RBR. Unfortunately, finding the optimal set of shards given a budget b is an instance of the knapsack problem, infeasible to solve in practice. Instead, we approximate the optimal selection by ordering the documents according to the fraction of relevant documents they contain, and selecting shards greedily, such that the total number of processed documents stays below b. Since the ordering used for the BBR is similar to that for computing wAUReC, the measure relates to BBR shard selection as normal AUReC relates to RBR.

3.3 Metrics

We follow Kim and Callan's [15] example to evaluate how wAUReC relates to the performance of an end-to-end system. We compute the correlation (Pearson's r) between each of the AUReC measures and the selective search system's end-to-end performance, with either shard selection algorithm. Like Kim and Callan [15], we evaluate the end-to-end system using P@1000, a deep, recall-focused measure that assumes selective search is used as a first-stage retrieval system.

Table 1: Correlation (Pearson's r) between AUReC variants and the P@1000 of end-to-end systems using relevance-based ranking for resource selection.

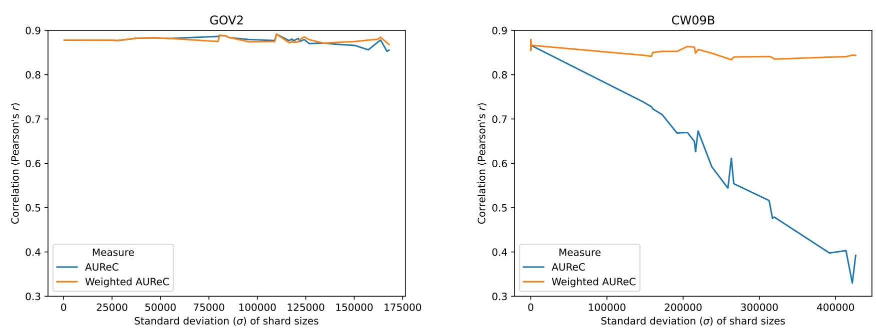
	Dai et al. [12]		Random	
End-to-end	*AUReC*	*wAUReC*	*AUReC*	*wAUReC*
RBR ($k = 1$)	0.932	0.934	0.258	0.891
RBR ($k = 3$)	0.929	0.930	0.261	0.897
RBR ($k = 5$)	0.925	0.927	0.261	0.899

4 Experimental results

4.1 Relevance-based ranking

The left-hand side of Table 1 shows the correlation between an RBR-based end-to-end system and both AUReC variants, across all six shard maps from Dai et al. [12]. We clearly see that the AUReC and wAUReC are more or less equivalent in this setting. In fact, the correlation between AUReC and wAUReC over all topics and datasets is 0.990, indicating the similar outcomes of the two variants.

For the randomly generated shard maps (right-hand side of Table 1), the wAUReC seems to be more correlated with the system's end-to-end performance, showing the added benefit of using shard size in the evaluation of shard maps. Figure 1 additionally shows the correlation as a function of the standard deviation of shard sizes in a shard map: the lower the standard deviation, the more balanced a shard map is. For Gov2, the correlation stays roughly the same, but for ClueWeb09B we clearly see a drop in performance for the AUReC when shard maps become more unbalanced, while the wAUReC remains more or less consistent. We observed similar outcomes for different RBR cutoff values k.

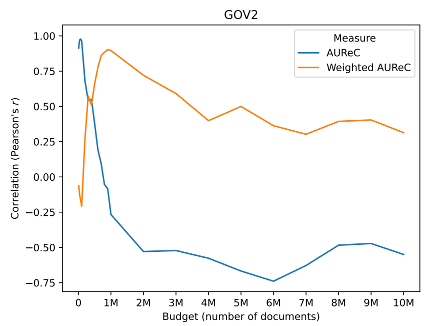


(a) Gov2 (b) ClueWeb09B

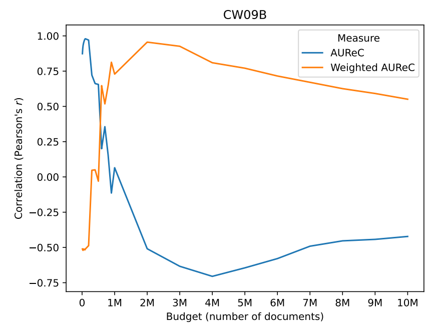
Fig. 1: Correlation (Pearson's r) between AUReC variants and P@1000 of end-to-end systems (with RBR and $k = 3$) for varying degrees of shard size skew.

4.2 Budget-based ranking

Figure 2 shows the correlation between the AUReC measures and end-to-end systems using BBR, for a wide range of budgets b. There is a large difference between the performance of the regular AUReC and the wAUReC for both datasets. As expected, AUReC performs sub-optimally in the setting where a system is limited in the number of documents it can process, rather than the number of shards. The wAUReC is better able to capture this budget-constrained environment, though its correlation also still leaves room for improvement (especially for small values b). A possible explanation for this outcome is that the artificial nature of the generated shard maps makes selective search more difficult in general. Alternatively, our greedy heuristic for determining the optimal BBR could result in suboptimal performance of the end-to-end system. Too small values for b might even make effective retrieval impossible altogether.



(a) Gov2



(b) ClueWeb09B

Fig. 2: Correlation (Pearson's r) between AUReC variants and the P@1000 of end-to-end systems using budget-based ranking for resource selection.

5 Conclusion

This paper introduced a weighted variant of the AUReC measure (wAUReC), which can be used to evaluate shard maps for use in a selective search system when the shards are skewed in size. First, we have shown that the wAUReC performs similarly to the normal AUReC when shard maps are balanced. Then, we showed that the AUReC performance degrades when shard sizes are skewed, and that its weighted counterpart can handle such shard maps better.

We also studied a setup in which a system does not select k shards but instead has a fixed budget b of documents that it can process given limited time or resources. In this case, it might be more worthwhile to select smaller shards with a higher relative number of relevant documents first, to not fill up the budget with a few large shards. In this setting, AUReC was unable to accurately measure the quality of a shard map. The wAUReC achieved a much higher correlation with the end-to-end system, for a wide range of budgets b.

We aim to continue this work and apply the wAUReC on datasets and shard maps with inherent size skew, to evaluate its performance in more realistic scenarios and ensure it can be applied in practice.

Unlike Kim and Callan did for AUReC [15], we have not yet investigated whether wAUReC can be used in significance testing. However, because of the strong similarities between the measures, we hypothesise that those findings also translate to the wAUReC. This hypothesis can be verified in future work.

Acknowledgments

This work has received funding from the European Union's Horizon Europe research and innovation programme under grant agreement No 101070014 (OpenWebSearch.EU, https://doi.org/10.3030/101070014). We also thank Yubin Kim, who kindly helped us with our experimental setup by making her code available.

References

1. Aly, R., Hiemstra, D., Demeester, T.: Taily: shard selection using the tail of score distributions. In: Proceedings of the 36th international ACM SIGIR conference on Research and development in information retrieval. pp. 673–682. SIGIR '13, Association for Computing Machinery, New York, NY, USA (Jul 2013). https://doi.org/10.1145/2484028.2484033, https://dl.acm.org/doi/10.1145/2484028.2484033
2. Anava, Y., Shtok, A., Kurland, O., Rabinovich, E.: A Probabilistic Fusion Framework. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 1463–1472. CIKM '16, Association for Computing Machinery, New York, NY, USA (Oct 2016). https://doi.org/10.1145/2983323.2983739, https://dl.acm.org/doi/10.1145/2983323.2983739
3. Büttcher, S., Clarke, C.L.A., Soboroff, I.: The TREC 2006 Terabyte Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Fifteenth Text REtrieval Conference, TREC 2006, Gaithersburg, Maryland, USA, November 14-17, 2006. NIST special publication, vol. 500-272. National Institute of Standards and Technology (NIST) (2006), http://trec.nist.gov/pubs/trec15/papers/TERA06.OVERVIEW.pdf
4. Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2004 Terabyte Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Thirteenth Text REtrieval Conference, TREC 2004, Gaithersburg, Maryland, USA, November 16-19, 2004. NIST special publication, vol. 500-261. National Institute of Standards and Technology (NIST) (2004), http://trec.nist.gov/pubs/trec13/papers/TERA.OVERVIEW.pdf
5. Clarke, C.L.A., Craswell, N., Soboroff, I.: Overview of the TREC 2009 Web Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Eighteenth Text REtrieval Conference, TREC 2009, Gaithersburg, Maryland, USA, November 17-20, 2009. NIST special publication, vol. 500-278. National Institute of Standards and Technology (NIST) (2009), http://trec.nist.gov/pubs/trec18/papers/WEB09.OVERVIEW.pdf
6. Clarke, C.L.A., Craswell, N., Soboroff, I., Cormack, G.V.: Overview of the TREC 2010 Web Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Nineteenth Text REtrieval Conference, TREC 2010, Gaithersburg, Maryland, USA, November 16-19, 2010. NIST special publication, vol. 500-294. National Institute of Standards and Technology (NIST) (2010), https://trec.nist.gov/pubs/trec19/papers/WEB.OVERVIEW.pdf
7. Clarke, C.L.A., Craswell, N., Soboroff, I., Voorhees, E.M.: Overview of the TREC 2011 Web Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Twentieth Text REtrieval Conference, TREC 2011, Gaithersburg, Maryland, USA, November 15-18, 2011. NIST special publication, vol. 500-296. National Institute of Standards and Technology (NIST) (2011), http://trec.nist.gov/pubs/trec20/papers/WEB.OVERVIEW.pdf
8. Clarke, C.L.A., Craswell, N., Voorhees, E.M.: Overview of the TREC 2012 Web Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Twenty-First Text REtrieval Conference, TREC 2012, Gaithersburg, Maryland, USA, November 6-9, 2012. NIST special publication, vol. 500-298. National Institute of Standards and Technology (NIST) (2012), http://trec.nist.gov/pubs/trec21/papers/WEB12.overview.pdf
9. Clarke, C.L.A., Scholer, F., Soboroff, I.: The TREC 2005 Terabyte Track. In: Voorhees, E.M., Buckland, L.P. (eds.) Proceedings of the Fourteenth Text RE-

trieval Conference, TREC 2005, Gaithersburg, Maryland, USA, November 15-18, 2005. NIST special publication, vol. 500-266. National Institute of Standards and Technology (NIST) (2005), http://trec.nist.gov/pubs/trec14/papers/TERABYTE.OVERVIEW.pdf
10. Dai, Z., Kim, Y., Callan, J.: How Random Decisions Affect Selective Distributed Search. In: Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 771–774. SIGIR '15, Association for Computing Machinery, New York, NY, USA (Aug 2015). https://doi.org/10.1145/2766462.2767796, https://dl.acm.org/doi/10.1145/2766462.2767796
11. Dai, Z., Kim, Y., Callan, J.: Learning To Rank Resources. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 837–840. SIGIR '17, Association for Computing Machinery, New York, NY, USA (Aug 2017). https://doi.org/10.1145/3077136.3080657, https://dl.acm.org/doi/10.1145/3077136.3080657
12. Dai, Z., Xiong, C., Callan, J.: Query-Biased Partitioning for Selective Search. In: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management. pp. 1119–1128. CIKM '16, Association for Computing Machinery, New York, NY, USA (Oct 2016). https://doi.org/10.1145/2983323.2983706, https://dl.acm.org/doi/10.1145/2983323.2983706
13. Ergashev, U., Dragut, E., Meng, W.: Learning To Rank Resources with GNN. In: Proceedings of the ACM Web Conference 2023. pp. 3247–3256. WWW '23, Association for Computing Machinery, New York, NY, USA (Apr 2023). https://doi.org/10.1145/3543507.3583360, https://doi.org/10.1145/3543507.3583360
14. Kim, Y.: Robust Selective Search. Ph.D. thesis, Carnegie Mellon University (2019)
15. Kim, Y., Callan, J.: Measuring the Effectiveness of Selective Search Index Partitions without Supervision. In: Proceedings of the 2018 ACM SIGIR International Conference on Theory of Information Retrieval. pp. 91–98. ICTIR '18, Association for Computing Machinery, New York, NY, USA (Sep 2018). https://doi.org/10.1145/3234944.3234952, https://dl.acm.org/doi/10.1145/3234944.3234952
16. Kim, Y., Callan, J., Culpepper, J.S., Moffat, A.: Does Selective Search Benefit from WAND Optimization? In: Ferro, N., Crestani, F., Moens, M.F., Mothe, J., Silvestri, F., Di Nunzio, G.M., Hauff, C., Silvello, G. (eds.) Advances in Information Retrieval. pp. 145–158. Lecture Notes in Computer Science, Springer International Publishing, Cham (2016). https://doi.org/10.1007/978-3-319-30671-1_11
17. Kim, Y., Callan, J., Culpepper, J.S., Moffat, A.: Load-Balancing in Distributed Selective Search. In: Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval. pp. 905–908. SIGIR '16, Association for Computing Machinery, New York, NY, USA (Jul 2016). https://doi.org/10.1145/2911451.2914689, https://dl.acm.org/doi/10.1145/2911451.2914689
18. Kulkarni, A., Callan, J.: Document allocation policies for selective searching of distributed indexes. In: Proceedings of the 19th ACM international conference on Information and knowledge management. pp. 449–458. CIKM '10, Association for Computing Machinery, New York, NY, USA (Oct 2010). https://doi.org/10.1145/1871437.1871497, https://dl.acm.org/doi/10.1145/1871437.1871497
19. Kulkarni, A., Callan, J.: Selective Search: Efficient and Effective Search of Large Textual Collections. ACM Transactions on Information Systems **33**(4), 17:1–17:33 (Apr 2015). https://doi.org/10.1145/2738035, https://dl.acm.org/doi/10.1145/2738035
20. Kulkarni, A., Tigelaar, A.S., Hiemstra, D., Callan, J.: Shard ranking and cutoff estimation for topically partitioned collections. In: Proceedings of the 21st ACM

international conference on Information and knowledge management. pp. 555–564. ACM, Maui Hawaii USA (Oct 2012). https://doi.org/10.1145/2396761.2396833, https://dl.acm.org/doi/10.1145/2396761.2396833
21. Lillis, D., Toolan, F., Collier, R., Dunnion, J.: Extending Probabilistic Data Fusion Using Sliding Windows. In: Macdonald, C., Ounis, I., Plachouras, V., Ruthven, I., White, R.W. (eds.) Advances in Information Retrieval. pp. 358–369. Lecture Notes in Computer Science, Springer, Berlin, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78646-7_33
22. Mohammad, H.R., Xu, K., Callan, J., Culpepper, J.S.: Dynamic Shard Cutoff Prediction for Selective Search. In: The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval. pp. 85–94. SIGIR '18, Association for Computing Machinery, New York, NY, USA (Jun 2018). https://doi.org/10.1145/3209978.3210005, https://dl.acm.org/doi/10.1145/3209978.3210005
23. Overwijk, A., Xiong, C., Callan, J.: ClueWeb22: 10 Billion Web Documents with Rich Information. In: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval. pp. 3360–3362. SIGIR '22, Association for Computing Machinery, New York, NY, USA (Jul 2022). https://doi.org/10.1145/3477495.3536321, https://dl.acm.org/doi/10.1145/3477495.3536321
24. Si, L., Callan, J.: Relevant document distribution estimation method for resource selection. In: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval. pp. 298–305. SIGIR '03, Association for Computing Machinery, New York, NY, USA (Jul 2003). https://doi.org/10.1145/860435.860490, https://dl.acm.org/doi/10.1145/860435.860490
25. Van Rijsbergen, C.J.: Information Retrieval. Butterworths (1979)