An Empirical Analysis of Pruning Techniques

Performance, Retrievability and Bias

Ruey-Cheng Chen
RMIT University
Melbourne, Australia
ruey-cheng.chen@rmit.edu.au

Leif Azzopardi
University of Strathclyde
Glasgow, United Kingdom
leifos@acm.org

Falk Scholer
RMIT University
Melbourne, Australia
falk.scholer@rmit.edu.au

ABSTRACT

Prior work on using retrievability measures in the evaluation of information retrieval (IR) systems has laid out the foundations for investigating the relation between retrieval performance and retrieval bias. While various factors influencing retrievability have been examined, showing how the retrieval model may influence bias, no prior work has examined the impact of the index (and how it is optimized) on retrieval bias. Intuitively, how the documents are represented, and what terms they contain, will influence whether they are retrievable or not. In this paper, we investigate how the retrieval bias of a system changes as the inverted index is optimized for efficiency through static index pruning. In our analysis, we consider four pruning methods and examine how they affect performance and bias on the TREC GOV2 Collection. Our results show that the relationship between these factors is varied and complex - and very much dependent on the pruning algorithm. We find that more pruning results in relatively little change or a slight decrease in bias up to a point, and then a dramatic increase. The increase in bias corresponds to a sharp decrease in early precision such as NDCG@10 and is also indicative of a large decrease in MAP. The findings suggest that the impact of pruning algorithms can be quite varied - but retrieval bias could be used to guide the pruning process. Further work is required to determine precisely which documents are most affected and how this impacts upon performance.

KEYWORDS

Indexing; Pruning; Retrievability

1 INTRODUCTION

Index optimization is important to ensure that documents can be retrieved efficiently. This is because delays associated with the delivery of search results can have a negative impact on the user experience, e.g. lower satisfaction (consciously and unconsciously [3]), lower volumes of queries [10], etc. One type of index optimization that is commonly employed is index pruning [1, 8, 11–13], where documents or terms are removed from the index or document representation (i.e. postings lists). However, such techniques tend to be one-sided, focusing on search efficiency at the expense of retrieval performance. As a result of this trade-off, an open question is how much pruning can be undertaken without serious loss in retrieval effectiveness.

In this paper, we consider another side of this problem and the possible trade-offs: What is the influence of pruning on the retrievability of documents, and therefore on retrieval bias? Retrievability is essentially a measure of the capabilities of a particular retrieval system to return documents in the collection, and retrieval bias is a measurement on the system about how the retrieval process makes certain documents more retrievable than others [4]. In prior work, it has been shown that the fairness of a system tends to correlate with retrieval performance [20], and that tuning the system via minimizing the retrieval bias may lead to improved performance [19].

The impact of a fair system is that users would experience less difficulties formulating queries to retrieve any document. In the context of pruning it could be argued that removing postings (i.e., essentially term-document pairs stored in a succinct way) makes documents less retrievable, thereby affecting the retrieval effectiveness across the indexed collection. However, not all terms in a document are likely to retrieve a document at a high enough rank that a user would see it. If we had a million documents, for example, which contained the term "*computer*", then in theory it is plausible to remove this term from a select subset of documents without impacting on the retrievability in general, because most of these documents were already ranked low even prior to pruning. When this happens, it can be expected that such a pruning has less impact on early precision as well. The ordering of documents may change, but top k precision should remain roughly the same. This is the original research aim of index pruning [12].

Given a particular pruning method, if a relationship between retrievability and performance holds, then it may be possible to exploit such a relationship, for instance, to decide when to stop pruning, or to provide ways to evaluate pruning methods, without requiring the use of relevance judgments. To this end, we perform an initial investigation of how pruning affects retrievability and retrieval bias across four static pruning algorithms on the TREC GOV2 Collection, and consider the following research questions:

- How does the retrieval bias of an IR system change as the inverted index is optimized for search efficiency?
- What is the relationship between retrieval performance and bias as the inverted index is optimized for search efficiency?

2 BACKGROUND

Research on index pruning can be divided into two areas, called static and dynamic index pruning, based on when and how the

pruning is performed. An inverted index can be pruned *statically* to reduce its size, by removing some part of the indexed content (e.g., postings) permanently. Alternatively, a retrieval system can choose to perform pruning *dynamically* while evaluating queries, skipping over postings that are not critical to document scoring at runtime, to speed up the retrieval process. To date, many successful approaches in both areas have relied on term importance measures, such as various forms of impact [2, 9, 11, 12], to determine whether a specific posting should be pruned or evaluated.

In this study, we focus on the use of static index pruning methods and their effect on retrievability and retrieval bias. In the next section, we review the pruning methods investigated in this study, before explaining what retrievability is and how it can be calculated to provide an estimate of retrieval bias.

2.1 Pruning Methods

Term-Based Pruning Carmel et al. [12] formulated static index pruning as a task of preserving the top-k search results in a retrieval system, and the formulation gave rise to an intuitive and efficient approach based on the traversal of term posting lists. The solution is essentially to scan through each term posting list in descending order of term impact, determine an adaptive cutting threshold τ (with a guarantee that top k postings in the list will remain intact), and prune away every posting that scores lower.

Document-Centric Pruning An alternative approach proposed by Büttcher and Clarke [11] is to remove from each document the term postings that contribute little to document relevance. In language modeling, document relevance can be represented as the KL divergence from document model to the collection model. The per-term contribution to this divergence score is then used to measure how important a term is to the document.

Uniform Pruning In term-based pruning the cutting threshold τ is computed adaptively according to the impact of the k-th most highly scored posting. A simpler version [12] sets this threshold as a constant uniformly to all terms. This approach was later shown to respond well to mean average precision by Chen and Lee [13].

Divergence-Based Pruning Chen et al. used various divergence measures (e.g. KL divergence and the like) to estimate the importance of a term-document pair in the inverted index [14]. One of the proposed variant using Rényi divergence of order infinity was shown to be comparable to the state of the art document-centric pruning method.

2.2 Retrievability

Retrievability, proposed by Azzopardi and Vinay [5], provides a way to quantify the influence of a system on a collection, and measures how *likely* a document is to be retrieved by a particular configuration of an IR system. The retrievability $\mathbf{r}$ of a document $\mathbf{d}$ with respect to the configuration of an IR system is defined as:

$$\mathbf{r}(\mathbf{d}) \propto \sum_{\mathbf{q} \in \mathbf{Q}} \mathbf{f}(\mathbf{k_{dq}}, \mathbf{c}, \beta)$$

where $\mathbf{q}$ is a query from a large query set $\mathbf{Q}$, and $\mathbf{k_{dq}}$ is the rank at which $\mathbf{d}$ is retrieved given $\mathbf{q}$. The utility function $\mathbf{f}(\mathbf{k_{dq}}, \mathbf{c}, \beta)$ determines the score that document $\mathbf{d}$ attains for query $\mathbf{q}$ given the rank cutoff $\mathbf{c}$ and a discount β. $\mathbf{r}(\mathbf{d})$ is calculated by summing over all queries $\mathbf{q}$ in query set $\mathbf{Q}$. Theoretically, $\mathbf{Q}$ represents the universe of all possible queries, but in practice it is commonly approximated with a large set of queries [5, 7]. The standard measure of retrievability used employs the utility function $\mathbf{f}(\mathbf{k_{dq}}, \mathbf{c}, \beta)$, such that if a document $\mathbf{d}$ is retrieved in the top $\mathbf{c}$ documents given $\mathbf{q}$, then $\mathbf{f}(\mathbf{k_{dq}}, \mathbf{c}, \beta) = 1/\mathbf{k}_{\mathbf{dq}}^{\beta}$, otherwise $\mathbf{f}(\mathbf{k_{dq}}, \mathbf{c}, \beta) = 0$. When $\beta = 0$, the measure is essentially cumulative i.e. the number of times that the document is retrieved in the top $\mathbf{c}$ documents, whereas when $\beta > 0$, documents further down the ranked list are assigned less utility (this is referred to as a gravity-based measure by Azzopardi and Vinay [5]).

To measure the retrieval bias of the system on the population of documents, an inequality measure is used [21]. The Gini Coefficient is a measure used in economics to calculate the level of inequality in a population [16]. Intuitively, if all the documents have the same level of $\mathbf{r}(\mathbf{d})$, then there is no inequality within the population of documents, and so Gini = 0.0. However, if all the documents have an $\mathbf{r}(\mathbf{d}) = 0$, except one document, then there is high inequality within the population (i.e. a King and all the peasants), so Gini = 1.0 denoting total inequality.

Given this measure of retrieval bias, a number of studies have been undertaken examining the relationship between bias and performance [6, 7, 18–20]. These works show that different retrieval algorithms exhibit different levels of retrieval bias across the collection, and this is affected by document length normalization, query length, query expansion and, of course, the retrieval algorithm itself. In general, it has been shown that optimizing the IR system, such that it minimizes retrieval bias, tends to lead to good performance on standard retrieval measures such as P@10 and MAP, and for more recent measures, such as Time Biased Gain and the U-measure, it tends to lead to the best performance [20].

In this work, we explore the relationship between pruning algorithms and their effect on retrieval bias and performance. Intuitively, if a document is not indexed, then its retrievability is zero, as the retrieval system cannot retrieve the document. This was a major concern for ensuring the accessibility of the document [17] and results, either due to the document not being indexed/crawled or it being removed/filtered prior to the retrieval (i.e. spam removal, which is seen as beneficial). However, here we will be focusing on index pruning methods, as they effectively change the representation of the document within the index and may affect its retrievability.

3 EXPERIMENTAL METHOD

3.1 Data and Materials

Collections We performed our analysis using the GOV2 test collection. The collection was indexed by using Indri,[1] with all documents Krovetz-stemmed and stopwords removed using the InQuery stoplist. The final index contains 25,205,179 documents and 39,177,923 unique terms. All retrieval runs were performed using BM25 with the parameters optimized ($k_1 = 0.9$ and $b = 0.4$).[2] To measure retrieval effectiveness we used TREC topics 701–850.

Pruning Methods As previously mentioned we used four pruning methods: Term-Based Pruning (TCP), Document-Centric Pruning (DCP), Uniform Pruning (UP) and Divergence-Based Pruning (REN).

[1] https://www.lemurproject.org/indri.php

[2] Following the setting in https://github.com/lintool/IR-Reproducibility.

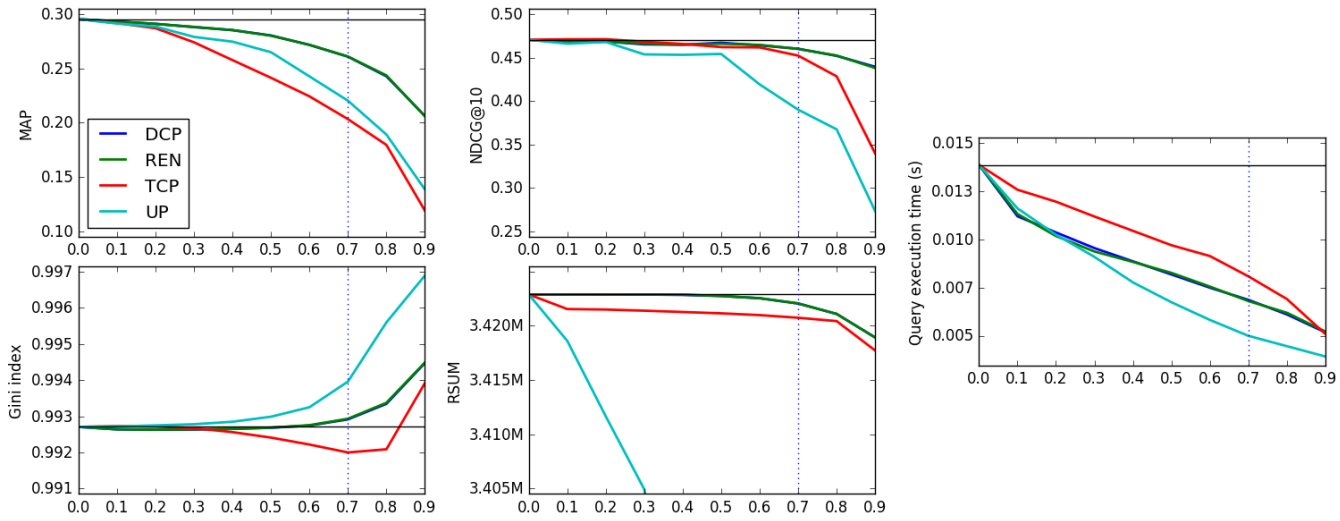

Figure 1: Changes in metrics across prune ratios (p-ratio) for each pruning algorithm.

These methods are implemented using the package released by Chen et al [14]. For TCP, we set the parameter $k = 10$ following the default setting. The other three methods do not have any parameters. In our experiments, we investigate the performance of the inverted index under prune ratios (p-ratios) $\{0.1, 0.2, \ldots, 0.9\}$, the fraction of postings permanently removed from the full index.

Effectiveness and Bias All experiments were executed on a Intel Xeon E5-2690 with 256 GB of RAM using at most 24 cores in parallel for retrieval. To measure retrieval effectiveness in each of our experiments we use Mean Average Precision (MAP) and Normalized Discounted Cumulative Gain at 10 (NDCG@10). Two-tailed t-test is used throughout the experiments for significance testing.

Retrievability bias was quantified using the Gini Coefficient, similar to previous studies [5, 6, 20]. To quantify the bias of the system we followed these steps. First, we extracted all bigrams (excluding ones with digits) from the collection which occurred at least 10 times, and randomly sampled 200,000 frequency-weighted bigrams out of this set. These queries were executed against each index, given each pruning configuration. Query execution time and the run output were recorded. Finally we computed the retrievability score for a series of cumulative and gravity-based measures, but due to space constraints we only report results based on the gravity-based measure where discount $\beta = 0.5$ and cutoff $c = 100$. Following this, we use Gini to compute the bias of the system on the overall collection using the $\mathbf{r}(\mathbf{d})$ scores. We also report the total retrievability (RSUM), which is $\sum_d r(d)$, to provide a measure of how much retrievability is afforded to the collection (a similar access measure is used by Garcia et al. [15]).

4 ANALYSIS

Figure 1 provides plots of MAP, NDCG@10, Gini coefficient, RSUM, and query time across the p-ratios for each of the pruning algorithms.[3] From these plots, we can see that as the p-ratio increases the effectiveness metrics and the Gini coefficient also change. A trend that can be broadly seen for the Gini coefficient is that bias remains fairly stable until a turning point, between p-ratio 0.3–0.7 depending on the pruning algorithm, after which bias increases. TCP behaves slightly differently before reaching the turning point, where bias decreases between p-ratio 0.4–0.7.

For DCP and REN, MAP is found to decrease fairly early on and steadily across the space. A significant drop in MAP is first seen at p-ratio 0.2 for DCP (p-value = 0.0316) and REN (p-value = 0.0321). Interestingly, DCP and REN have little impact on early precision, as NDCG@10 remains at a similar level until p-ratio 0.8. It is not until p-ratio reaches 0.9 that a statistically significant decrease in NDCG@10 first takes place for DCP (p-value = 0.0262) and REN (p-value = 0.0205). In most cases, DCP and REN perform similarly across metrics such as RSUM and the Gini coefficient. Note that the Blue (DCP) and Green (REN) lines are often on top of each other.

For TCP and UP, MAP begins to decrease slowly up to a p-ratio 0.4, and then MAP starts to fall off more sharply (more so for TCP than UP), losing approx. 0.13 in MAP from p-ratio 0.4 to 0.9. Interestingly, it is at around p-ratio 0.4 that the bias of the system starts to change (for TCP it decreases, while for UP it starts to increase), suggesting that this is an inflection point of sorts. With respect to early precision, NDCG@10 performance degrades more slowly, until p-ratio 0.5 (UP) or 0.7 (TCP), and then it drops sharply as the bias for increases for both methods. An interesting difference between TCP and UP is that the total retrievability (RSUM) for UP decreases substantially compared with TCP (and the other methods). UP at p-ratio 0.9 loses approximately 1,800,000 in retrievability, i.e. meaning that there is effectively less access to documents throughout the collection. Given that bias increases, it suggests that not only has access to documents been removed, but now certain documents have more access relative to others. On the other hand, while TCP maintains the level of access across the collection, the retrievability is redistributed among the population of documents (resulting in only a small reduction in bias, but at the expense of performance).

[3] A full analysis on other performance measures, e.g. MRR, P@10, etc, and the source data are given at https://github.com/rueycheng/pruning_retrievability

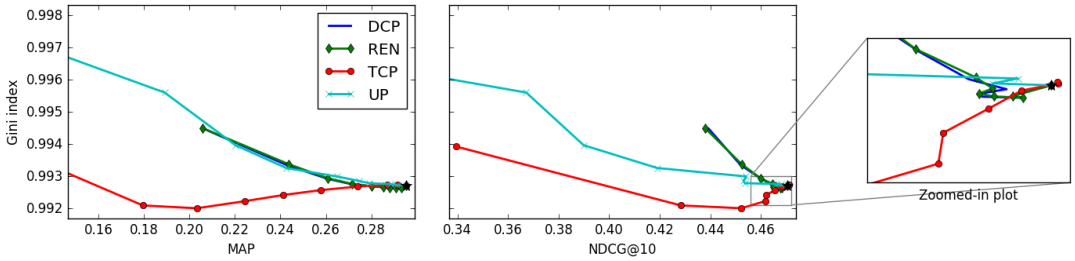

Figure 2: The relationship between Gini and MAP (left) and NDCG@10 (right, with zoomed-in plot)

With respect to efficiency, the rightmost plot in Figure 1 shows how the total query execution time in seconds changes across p-ratios for each algorithm. UP results in the fastest execution time at the cost of lower performance. While all methods result in similar curves, TCP appears to be less advantageous in general despite the fact that it results in less retrieval bias compared to the others.

Figure 2 provides plots of Gini versus MAP (left) and NDCG@10 (right, with a zoomed-in plot) to show how bias and performance relate to each other. The star indicates when the p-ratio is 0.0, which is the "starting point" for an un-pruned index with each subsequent point corresponding to an 0.1 increase in the prune ratio. Also note, that since the bias and performance do not change until around p-ratio 0.6 (excluding the global early drop at p-ratio 0.1) for DCP and REN, many of the points are overlapping in this region. For DCP and REN, we can see that it is not until the later few points, when the p-ratio is greater than 0.5, that NDCG@10 decreases, while the Gini coefficient increases. For TCP and UP, there is an immediate trade-off between performance and bias, before the turning point after which there are increasing drops in performance.

5 DISCUSSION AND FUTURE WORK

In this paper, we conducted an initial exploration into the relationship between retrievability bias, performance and efficiency across four different pruning algorithms. Our results suggest that the relationship is complex and very much algorithm-dependent. We found one general pattern, which is bias either remained stable or slightly decreased before reaching a turning point and increasing. It would appear that selecting the p-ratio based on Gini would result in good performance (on early precision such as NDCG@10) without an increase in bias or sizable loss in retrievability. However, there are also cases where the situation is more complicated (TCP and UP), and while it appears that the pruning can help mitigate bias at the expense of performance, going beyond the turning point for bias is highly detrimental to performance. In this case, UP also has the further consequence of substantially reducing the overall retrievability of documents. So while retrieval efficiency is greatly improved it comes at a cost. This is an important result of our analysis.

These findings provide novel insights into the influence and effect of index pruning and how it changes the retrievability of documents given a retrieval system. These findings also indicate that it may be possible to select the level of index optimization using retrievability, and thus provide a way to quantify/evaluate the influence of a pruning algorithm on the index. However, future work is required to explore these directions on other collections, other pruning algorithms, and to examine interactions with different retrieval models.

ACKNOWLEDGMENTS

This research was supported in part by the Australian Research Council (DP140102655).

REFERENCES

[1] Ismail S Altingovde, Rifat Ozcan, and Özgür Ulusoy. 2012. Static index pruning in web search engines: Combining term and document popularities with query views. *ACM Transactions on Information Systems (TOIS)* 30, 1 (2012), 2.
[2] Vo N. Anh and Alistair Moffat. 2006. Pruned query evaluation using pre-computed impacts. In *Proceedings of SIGIR '06*. ACM, 372–379.
[3] Ioannis Arapakis, Xiao Bai, and B. Barla Cambazoglu. 2014. Impact of Response Latency on User Behavior in Web Search. In *Proceedings of SIGIR '14*. 103–112.
[4] Leif Azzopardi and Vishwa Vinay. 2008. Accessibility in information retrieval. In *Proceedings of ECIR '08*. 482–489.
[5] Leif Azzopardi and Vishwa Vinay. 2008. Retrievability: An Evaluation Measure for Higher Order Information Access Tasks. In *Proceedings of CIKM '08*. 561–570.
[6] Shariq Bashir and Andreas Rauber. 2009. Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In *Proceedings of CIKM '09*. 1863–1866.
[7] Shariq Bashir and Andreas Rauber. 2010. Improving retrievability of patents in prior-art search. In *Proceedings of ECIR '10*. 457–470.
[8] Roi Blanco and Alvaro Barreiro. 2010. Probabilistic static pruning of inverted files. *ACM Transactions on Information Systems* 28, 1 (Jan. 2010).
[9] Andrei Z Broder, David Carmel, Michael Herscovici, Aya Soffer, and Jason Zien. 2003. Efficient query evaluation using a two-level retrieval process. In *Proceedings of CIKM '03*. ACM, 426–434.
[10] Jake D Brutlag, Hilary Hutchinson, and Maria Stone. 2008. User preference and search engine latency. *JSM Proceedings, Qualtiy and Productivity Research Section* (2008).
[11] Stefan Büttcher and Charles L. A. Clarke. 2006. A document-centric approach to static index pruning in text retrieval systems. In *Proceedings of CIKM '06*. ACM, 182–189.
[12] David Carmel, Doron Cohen, Ronald Fagin, Eitan Farchi, Michael Herscovici, Yoelle S. Maarek, and Aya Soffer. 2001. Static index pruning for information retrieval systems. In *Proceedings of SIGIR '01*. ACM, 43–50.
[13] Ruey-Cheng Chen and Chia-Jung Lee. 2013. An Information-Theoretic Account of Static Index Pruning. In *Proceedings of SIGIR '13*. ACM, 163–172.
[14] Ruey-Cheng Chen, Chia-Jung Lee, and W Bruce Croft. 2015. On divergence measures and static index pruning. In *Proceedings of ICTIR '15*. ACM, 151–160.
[15] Steven Garcia, Hugh E. Williams, and Adam Cannane. 2004. Access-ordered Indexes. In *Proceedings of the 27th Australasian Conference on Computer Science - Volume 26 (ACSC '04)*. 7–14.
[16] J Gastwirth. 1972. The Estimation of the Lorenz Curve and Gini Index. *The Review of Economics and Statistics* 54 (1972), 306–316. Issue 3.
[17] Steve Lawrence and C. Lee Giles. 2000. Accessibility of Information on the Web. *Intelligence* 11, 1 (April 2000), 32–39.
[18] Colin Wilkie and Leif Azzopardi. 2013. Relating retrievability, performance and length. In *Proceedings of SIGIR '13*. 937–940.
[19] Colin Wilkie and Leif Azzopardi. 2014. Best and Fairest: An Empirical Analysis of Retrieval System Bias. *Advances in Information Retrieval* (2014), 13–25.
[20] Colin Wilkie and Leif Azzopardi. 2014. A Retrievability Analysis: Exploring the Relationship Between Retrieval Bias and Retrieval Performance. In *Proceedings of CIKM '14*. 81–90.
[21] Colin Wilkie and Leif Azzopardi. 2015. Retrievability Bias: A Comparison of Inequality Measures. *Advances in Information Retrieval* (2015), 209–214.