Revisiting Spam Filtering in Web Search

Luke Gallagher
RMIT University
Melbourne, Australia

Joel Mackenzie
RMIT University
Melbourne, Australia

J. Shane Culpepper
RMIT University
Melbourne, Australia

ABSTRACT

The Waterloo spam scores are now a commonly used static document feature in web collections such as ClueWeb. This feature can be used as a post-retrieval filter, as a document prior, or as one of many features in a Learning-to-Rank system. In this work, we highlight the risks associated with using spam scores as a post-retrieval filter, which is now common practice in experiments with the ClueWeb test collection. While it increases the average evaluation score and boosts the performance of some topics, it can significantly harm the performance of others. Through a detailed failure analysis, we show that simple spam filtering is a high risk practice that should be avoided in future work, particularly when working with the ClueWeb12 test collection.

1 INTRODUCTION

Static document features are an integral component of modern web search engines. They have been used extensively in all stages of search, from early phases such as static index reordering and first stage retrieval, through to late stage retrieval processes of Learning-to-Rank and post-retrieval filtering [11–13]. The focus of this work is on one particular static document feature – spam scoring. The spam score of a web document can provide a signal as to its overall quality. For large web corpora, spam identification and removal is an essential component, applied either within a complex model, or directly as an independent filtering stage (white / black listing).

Risk is characterized by the effect of downside losses where a new system performs worse on given query topics when compared to a baseline system, or for example, comparisons of a new version of a system with a previous one. Users are often sensitive to failures in a search session, which can result in user abandonment [14]. This issue is often ignored in IR evaluation exercises, as measuring system performance using aggregate scores does not penalize a new system for significant failures on a small subset of topics. A family of risk-aware evaluation measures based on URisk were designed to quantify the level of risk (or reward) in a system and are grounded in statistical hypothesis testing theory. The intention is to identify the elements of risk between systems [7, 15].

In this work, we assess the risk-reward trade-off of spam scores through a careful failure analysis of simplified models to identify the properties of the collection or query/spam score combination that pose the most risk.

ADCS '18, December 11–12, 2018, Dunedin, New Zealand

ACM ISBN 978-1-4503-6549-9/18/12…$15.00
https://doi.org/10.1145/3291992.3291999

Contributions. We investigate the use of simple spam filtering for index pruning, post-retrieval filtering, and as a document prior. Through a detailed failure analysis we highlight the risks associated with these techniques on the ClueWeb test collection using the Waterloo spam score dataset provided by Cormack et al. [4]. To our knowledge, no comprehensive risk-reward failure analysis has been performed on the document spam score across both ClueWeb collections, and little is known about the behavior of spam scoring in the ClueWeb12 test collection.

2 BACKGROUND

Waterloo Spam Scores. Cormack et al. [4] provide an extensive analysis on the construction of efficient and effective spam rankers for the English portion of the ClueWeb09 collection. The spam filter design was comprised of three filters based on different spam corpora. *UK2006* – A small labeled corpus of 767 spam pages and 7,474 non-spam pages. *Britney* – An automatically labeled set of spam/non-spam pages from a set of 1,000 popular queries derived from commercial search engines and likely to be an attractive target for spammers. *Group X* – Manually labeled training examples for queries from the TREC 2009 Ad-hoc task. A fourth meta-filter known as *Fusion* was constructed using model combination of all previous filters. The fusion filter provided the best effectiveness over any other independent filter. The results show significant improvements in effectiveness to nearly all submitted runs for the TREC 2009 Ad-hoc task. Cormack et al. made the datasets of the fusion scores publicly available for both ClueWeb09[1] and ClueWeb12[2]. The fusion score assigns a binned percentile $p(D)$ to each of the documents in each corpus, where a lower score implies a lower quality document.

Index Augmentation. Zuccon et al. [17] provide a comprehensive analysis on the effect of indexing the ClueWeb09B collection without spam. Through the construction of a series of pruned indexes, the authors show that even though the collection statistics are different as a result of pruning, retrieval effectiveness does not reveal significant evidence to suggest degraded performance. The unsurprising upside is that indexing throughput is increased with reduced storage – a line of research also explored by Crane and Trotman [5]. Another possible index augmentation technique is to statically reorder an index based on a machine learned model where the spaminess of a document could be one of many features in such a model [12]. We leave exploration of this interesting technique to future work.

Risk-Sensitive Measures. Risk-sensitive approaches to system evaluation promote system robustness. Improvements in effectiveness on average can obfuscate poor performance on individual topics, and is usually ignored. A risk-reward trade-off exists where a given system should not perform significantly worse than a given baseline system at the topic level. The TREC web track in 2013 defined this

[1] http://plg.uwaterloo.ca/~gvcormac/clueweb09spam/
[2] http://mansci.uwaterloo.ca/~msmucker/cw12spam/

formulation as a utility risk measure URisk [3]. Given a set of matched pairs of results between two systems, URisk is the sum of the wins minus the sum of losses, normalized by the number of topics. The free parameter $\alpha \geq 0$ controls the amplification of risk towards topics that degrade in performance. More recently, Dinçer et al. [7] proposed TRisk, which is a formalization of URisk grounded in statistical hypothesis testing. By presenting URisk as a transformation that follows a Student's t distribution, TRisk can be used to convey if a system exhibits any real risk not ascribed to chance when compared to a baseline system. While TRisk only considers a single baseline, methods that utilize multiple baselines also exist – ZRisk and GeoRisk [8]

The per-topic expectation of robustness is calculated with the premise that a robust system is measured with respect to a population of baselines. However, only TRisk is *inferential*. This allows risk to be quantified in a statistically grounded way, and therefore we explore only TRisk in our current evaluation. The formulation of risk-sensitivity is orthogonal to the evaluation metric of choice, and is therefore an extension that can be used in conjunction with a variety of existing evaluation measures.

3 SPAM, CLUEWEB AND RISK

Spam. A number of pragmatic solutions are currently employed by researchers to incorporate spam relevance in search. Each have their own advantages and disadvantages. Index pruning [17] is one such strategy already discussed and most notably changes the collection statistics. The other common approach is post-retrieval filtering via a blacklist [3, 4]. Given a spam ranking for a retrieved document, a threshold is used to perform a binary judgment classing the document as either spam or legitimate. However, a binary decision may be overly aggressive in certain settings, and so a more fine grained approach would be to incorporate the spam ranking of a document as a *prior* or feature in a weighted ranking function [6].

Properties of ClueWeb. In the recent work of Xiong et al., the authors comment on spam filtering when using ClueWeb12B: "Spam filtering was not used for ClueWeb12 because its effectiveness is unclear." [16]. This remark suggests that spam filtering on ClueWeb12B appears to degrade retrieval performance, but the impact has never been fully explored. For ClueWeb09B, Bendersky et al. [2] detail in their TREC report from 2010 that a spam filtering stage post-retrieval is optimal at 60% for metrics ERR@10 and NDCG@10 [2]. Recent experiments in our lab also indicate similar characteristics – spam filtering improves effectiveness on ClueWeb09B but consistently hurts it for ClueWeb12B. This surprising result warrants further study. For instance, consider the hypothetical case of submitting system runs for a new TREC track on the ClueWeb12B collection, where relevance judgments are not yet available for the track topics. Deciding to include a system run that filters spam documents would likely be based on the premise that it improves effectiveness for ClueWeb09B, but this seems to not be true. Indeed, there have been instances of this occuring in prior work [9]. Figure 1 provides insight into why spam filtering helps on ClueWeb09B but not for ClueWeb12B. ClueWeb09B exhibits a skewed distribution of spam scores across the Category-B subset, whereas every other quadrant of Figure 1 shows a uniform distribution. This indicates that the ClueWeb09B subset contains a higher percentage of higher quality documents (also observed by Cormack et al. [4] for ClueWeb09B),

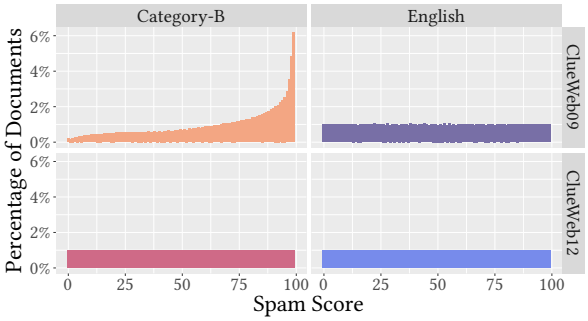

Figure 1: Distribution of spam scores across both ClueWeb09 and ClueWeb12 for both the Category-B subset and the entire English corpus.

which may indicate why filtering is effective. This collection bias lies in how the ClueWeb09B subset was created. The first 50 million English pages[3] were used to form this subset which includes a large portion of Wikipedia documents, and is likely to contain a number of other high quality domains that were used to seed the crawling process. However, the distribution of ClueWeb12B is uniform as a result of selecting every 14th document[4] from the entire crawl to form the Category-B subset, which is a uniform subset. This may be one of the reasons why applying the same spam filtering techniques appears to degrade retrieval performance on ClueWeb12B.

Risk. It is clear the use of spam across both ClueWeb collections can dramatically change the overall effectiveness. However, no prior work has performed a comprehensive failure analysis that quantifies the risk associated with the feature when used as a post-retrieval filter, despite it being common practice. Our goal is to compare different methods related to post-retrieval filtering to ascertain if there is any real risk associated with these practices, and to understand if there is any discrepancy between the individual methods themselves. For this task we employ the TRisk measure [7].

4 EXPERIMENTS

Experimental Configuration. All experiments were conducted using the Lemur toolkit. Indexing was performed with Indri[5] using Krovetz stemming and no stopping. In addition to the full index, a series of pruned indexes were created using the spam scores below percentiles 50, 60 and 70 as a blacklist for both ClueWeb collections. Retrieval was performed using Indri 5.11 with Query likelihood (QL) and a Sequential dependence model with field information (SDMF). Smoothing parameters were left at the default ($\mu = 2500$). Spam priors were created by converting the fusion percentile scores into a log probability that can be directly incorporated into a language model within Indri.

Collections. The web collections ClueWeb09B and ClueWeb12B were used, which represent a large scale crawl of the web from 2009 and 2012 respectively. Each collection contains around 50 million English documents. For ClueWeb09B, the 200 topics from the TREC

[3] http://lemurproject.org/clueweb09/index.php
[4] http://lemurproject.org/clueweb12/specs.php
[5] http://lemurproject.org/indri.php

web tracks of 2009–2012 are used (with the exception of topic 70 which is composed of mostly stop words). A new ad-hoc set of topics were used for the ClueWeb12B corpus from the NTCIR-13 We Want Web track [10]. The 100 topics provided for the English sub-task of this track are used. A third query set employed is the UQV100 topics for the ClueWeb12B collection [1]. The UQV100 dataset consists of query variations for each of the 100 topics from TREC 2013–2014 web tracks. We take the most common query variant for each topic to form a set of 100 unique queries that can be treated as an ad-hoc query set. The TREC 2013–2014 web track topics were not considered, as these queries contain subtopics for intent-aware and diversification tasks, and not ad-hoc retrieval.

Parameters. Experiments using spam as a weighted document prior consisted of queries of the form `#weight(`λ `#prior(`D`)` $(1-\lambda)Q$`)`, where D is a document and Q is the original query (QL, SDMF). The SDMF model is taken directly from Gallagher et al. [9], which is one of the systems which is part of the 2018 TREC CENTRE reproducibility track, and is publicly available[6]. Tuning of λ was performed via 5-fold train-test split at 80%, 20%. Training involved a sweep over values $\{0.0, 0.1, \ldots, 1.0\}$ to select λ for each test fold.

Evaluation. Results are reported using NDCG@10 for both the ClueWeb09B and ClueWeb12B collections. The `gdeval`[7] program was used for evaluation. The TRisk measure is used to compute the risk sensitivity of a system compared to another baseline system [8]. TRisk results are reported where $\alpha = 2$, posing the situation where the downside risk of degrading query performance is two times more costly than improving query performance. A TRisk score ≤ -2 means that the new system exhibits statistically significant risk relative to the baseline.

Results. As discussed previously, the spam score distribution is determined by the quality of the documents in each collection (Figure 1). This artifact can also be observed in Figure 3, where baseline systems for QL and SDMF are plotted by increasing NDCG@10 score for each topic. The points deviating away from these lines is the same system performing post-retrieval filtering at the suggested 70th percentile. The horizontal lines represent the mean effectiveness of each system. Figure 3 shows that filtering spam on ClueWeb09B improves overall effectiveness over both QL and SDMF baselines respectively. When contrasted with NTCIR-13 and UQV100, the same filtering operation on ClueWeb12B degrades performance more often than it helps. Pruned indexes suffer from the same effects which can be observed in Table 1, suggesting that index pruning or post-retrieval filtering of spam documents should be handled carefully when working with the ClueWeb collections. Using spam as a weighted document prior follows the same trend as filtering or pruning but the overall impact is less volatile.

Turning our attention to Figure 2, we experiment with post-retrieval filtering for each binned percentile score. That is, we evaluate every possible post-retrieval filter from 0 to 99, and plot the mean effectiveness score. As more documents are pruned on NTCIR-13 and UQV100 the overall effectiveness of NDCG@10 decreases. However, the same experiment on ClueWeb09B shows that filtering anywhere between the 1st and 85th percentiles will result in improved effectiveness.

[6] http://www.centre-eval.org/

[7] http://github.com/trec-web/trec-web-2014

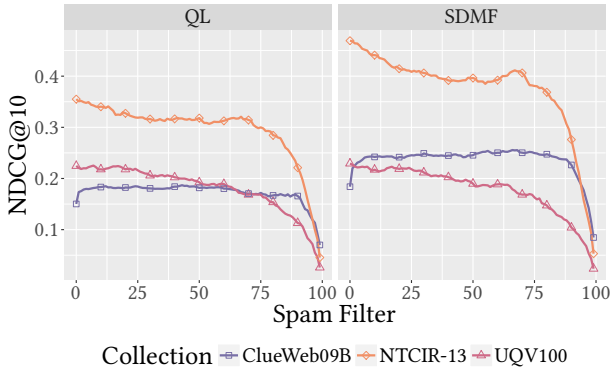

Figure 2: The mean NDCG@10 for both QL and SDMF for each possible level of spam filtering from 0 to 99. Recall that higher spam scores refer to higher quality documents.

Spaminess and Risk. From a practical perspective, the assumption that spam filtering is useful based on evidence from its use on ClueWeb09B would appear legitimate when considering its use on ClueWeb12B. However, the results in Table 1 show this is not the case. Both index pruning and post-retrieval filtering operations on NTCIR-13 and UQV100 result in a significantly riskier system compared to the baseline of each retrieval method (QL, SDMF). Interestingly, for QL on ClueWeb09B, filtering or pruning improves effectiveness while increasing risk. At the 70th percentile, the imposed risk is significant when compared to the QL baseline, and its effectiveness does not exhibit enough evidence to produce a significant improvement. Looking at SDMF on ClueWeb09B, utilizing spam scores shows significant improvements and poses no risk when compared to the SDMF baseline. However, when comparing these results to NTCIR-13 and UQV100, effectiveness is degraded, and in some cases, significantly so.

5 CONCLUSION AND DISCUSSION

In this work, we have investigated the behavior of spam scores across two large web corpora and three query sets. Our findings show the use of index pruning or post-retrieval filtering is inconsistent across ClueWeb09B and ClueWeb12B and lead to the conclusion that performing spam pruning or filtering on the ClueWeb collections should be avoided with the current Waterloo spam scores. We make this recommendation based on the details of how ClueWeb09B was built, resulting in a skewed distribution of high quality documents, which appears to either overemphasize the result of spam filtering on ClueWeb09B or cause the limited effectiveness apparent when they are used in conjunction with ClueWeb12B.

In the future, a study involving the log-odds estimate from Craswell et al. [6] combined with an analysis of the interplay of multiple static document features would be an interesting endeavor to understand the effect spam has within more complex models.

Acknowledgments. This work was supported by the Australian Research Council's *Discovery Projects* Scheme (DP170102231), and an Australian Government Research Training Program Scholarship.

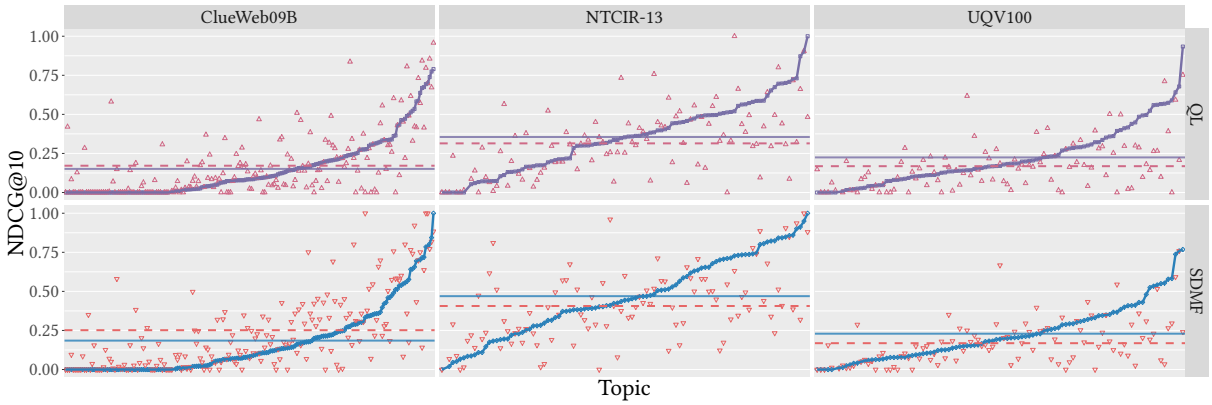

Figure 3: Per-topic NDCG@10 for each collection, increasing by score w.r.t. the baseline system (QL or SDMF). Each point denotes the corresponding score where spam is filtered at the 70th percentile. Solid horizontal lines denote the mean effectiveness *without* spam filtering, and the dashed line represents the mean effectiveness with spam filtering applied at the 70th percentile.

Table 1: Effectiveness scores across all collections. NDCG@10 is reported using a pairwise Bonferroni corrected t-test with † indicating $p < 0.05$ and ‡ indicating $p < 0.01$ using each system as a pairwise group (QL, SDMF). Significant downside risk for TRisk is reported compared to a single baseline using the same p-value notation as NDCG@10 (†, ‡). TRisk was computed with $\alpha = 2$. Wins and losses are reported where the per-topic delta is 10% greater than or 10% less than the baseline for each system type (QL, SDMF).

		ClueWeb09B				ClueWeb12B (NTCIR-13)				ClueWeb12B (UQV100)			
System	Spam	NDCG@10	TRisk	Win	Loss	NDCG@10	TRisk	Win	Loss	NDCG@10	TRisk	Win	Loss
QL	–	0.1502	–	–	–	0.3551	–	–	–	0.2245	–	–	–
QL-Prior	–	0.1827‡	-0.214	81	44	0.3496	-2.187†	5	7	0.2183	-2.056†	10	15
QL-Filter	50	0.1817‡	-0.599	83	39	0.3181	-4.485‡	30	43	0.1926	-4.704‡	31	46
QL-Filter	60	0.1802†	-1.388	83	48	0.3127	-4.703‡	30	45	0.1891	-4.502‡	36	41
QL-Filter	70	0.1705	-2.590†	79	56	0.3144	-4.775‡	32	45	0.1684†	-5.579‡	31	57
QL-Pruned	50	0.1821‡	-0.579	84	39	0.3191	-4.468‡	30	42	0.1914	-4.749‡	30	45
QL-Pruned	60	0.1804†	-1.382	82	48	0.3139	-4.683‡	31	43	0.1880	-4.559‡	36	41
QL-Pruned	70	0.1720	-2.468†	79	56	0.3154	-4.730‡	32	42	0.1684†	-5.591‡	31	56
SDMF	–	0.1840	–	–	–	0.4694	–	–	–	0.2292	–	–	–
SDMF-Prior	–	0.2523‡	1.369	102	36	0.4694	0.000	0	0	0.2217	-2.223†	4	11
SDMF-Filter	50	0.2454‡	0.671	111	32	0.3965‡	-6.856‡	23	54	0.1897	-4.862‡	25	45
SDMF-Filter	60	0.2503‡	0.844	113	30	0.3923‡	-6.574‡	26	55	0.1885	-4.857‡	32	44
SDMF-Filter	70	0.2504‡	0.273	110	41	0.4064	-5.580‡	32	51	0.1684‡	-5.827‡	28	56
SDMF-Pruned	50	0.2453‡	0.678	108	33	0.3965‡	-6.833‡	23	54	0.1907	-4.923‡	26	47
SDMF-Pruned	60	0.2505‡	0.825	111	31	0.3926‡	-6.603‡	26	55	0.1901	-4.820‡	31	46
SDMF-Pruned	70	0.2506‡	0.252	109	42	0.4060	-5.611‡	31	51	0.1697‡	-5.798‡	30	57

REFERENCES

[1] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. 2016. UQV100: A Test Collection with Query Variability. In *Proc. SIGIR*. 725–728.

[2] M. Bendersky, D. Fisher, and W. B. Croft. 2010. UMass at TREC 2010 Web Track: Term Dependence, Spam Filtering and Quality Bias. In *Proc. TREC*.

[3] K. Collins-Thompson, P. Bennett, F. Diaz, C. L.A. Clarke, and E. Voorhees. 2013. TREC 2013 Web Track Overview. In *Proc. TREC*.

[4] G. Cormack, M. Smucker, and C. L.A. Clarke. 2011. Efficient and Effective Spam Filtering and Re-ranking for Large Web Datasets. *Inf. Retr.* 14, 5 (2011), 441–465.

[5] M. Crane and A. Trotman. 2012. Effects of Spam Removal on Search Engine Efficiency and Effectiveness. In *Proc. Aust. Doc. Comp. Symp.* 1–8.

[6] N. Craswell, S. Robertson, H. Zaragoza, and M. Taylor. 2005. Relevance Weighting for Query Independent Evidence. In *Proc. SIGIR*. 416–423.

[7] B. T. Dinçer, C. Macdonald, and I. Ounis. 2014. Hypothesis Testing for the Risk-sensitive Evaluation of Retrieval Systems. In *Proc. SIGIR*. 23–32.

[8] B. T. Dinçer, C. Macdonald, and I. Ounis. 2016. Risk-Sensitive Evaluation and Learning to Rank Using Multiple Baselines. In *Proc. SIGIR*. 483–492.

[9] L. Gallagher, J. Mackenzie, R. Benham, R-C. Chen, F. Scholer, and J. S. Culpepper. 2017. RMIT at the NTCIR-13 We Want Web task. In *Proc. NTCIR*.

[10] C. Luo, T. Sakai, Y. Liu, Z. Dou, C. Xiong, and J. Xu. 2017. Overview of the NTCIR-13 We Want Web Task. In *Proc. NTCIR-13*.

[11] T. Qin and T. Y. Liu. 2013. Introducing LETOR 4.0 Datasets. *CoRR* (2013). http://arxiv.org/abs/1306.2597

[12] M. Richardson, A. Prakash, and E. Brill. 2006. Beyond PageRank: Machine Learning for Static Ranking. In *Proc. WWW*. 707–715.

[13] D. Shan, S. Ding, J. He, H. Yan, and X. Li. 2012. Optimized Top-k Processing with Global Page Scores on Block-max Indexes. In *Proc. WSDM*. 423–432.

[14] E. M. Voorhees. 2004. Overview of the TREC 2004 Robust Retrieval Track.. In *Proc. TREC*. 69–77.

[15] L. Wang, P. Bennett, and K. Collins-Thompson. 2012. Robust Ranking Models via Risk-sensitive Optimization. In *Proc. SIGIR*. 761–770.

[16] C. Xiong, J. Callan, and T. Y. Liu. 2016. Bag-of-Entities Representation for Ranking. In *Proc. ICTIR*. 181–184.

[17] G. Zuccon, A. Nguyen, T. Leelanupab, and L. Azzopardi. 2011. Indexing without spam. In *Proc. Aust. Doc. Comp. Symp.* 6–13.