On CORI Results Merging

Ilya Markov[1], Avi Arampatzis[2], and Fabio Crestani[1]

[1] University of Lugano, Via G. Buffi 13, 6900, Lugano, Switzerland
{ilya.markov,fabio.crestani}@usi.ch
[2] Democritus University of Thrace, Xanthi 67 100, Greece
avi@ee.duth.gr

Abstract. Score normalization and results merging are important components of many IR applications. Recently MinMax—an unsupervised linear score normalization method—was shown to perform quite well across various distributed retrieval testbeds, although based on strong assumptions. The CORI results merging method relaxes these assumptions to some extent and significantly improves the performance of MinMax. We parameterize CORI and evaluate its performance across a range of parameter settings. Experimental results on three distributed retrieval testbeds show that CORI significantly outperforms state-of-the-art results merging and score normalization methods when its parameter goes to infinity.

1 Introduction

Score normalization and results merging are crucial components in distributed retrieval, meta-search and other IR applications. Given a set of scored result lists, produced by multiple document sources, these components are concerned with making document relevance scores comparable across sources [2–4, 6]. Recently it was shown that MinMax score normalization performed quite well in various distributed retrieval testbeds [5]. However, MinMax assumes that

(i) each source contains at least one relevant document, and, (ii) relevant documents are likely to be ranked first. Therefore, it fails when only a few sources out of many contain relevant information, i.e. the first assumption is not satisfied [5].

The CORI results merging technique [2] overcomes this problem to some extent by performing resource selection and weighting each result list by the relevance of the corresponding source. This way, CORI removes the undesirable effect of the presence of many non-relevant sources, relaxes the assumptions of MinMax and significantly improves its performance.

In this work we parameterize CORI and study its behavior with respect to the parameter. Using three distributed retrieval testbeds we show that CORI achieves the best performance when its parameter goes to infinity. In this case CORI significantly outperforms other state-of-the-art results merging and score normalization methods.

2 Parameterizing CORI

CORI uses the following formula to normalize the scores of documents from a source R:

$$s_{norm}(d|q) = \frac{1 + 0.4 \cdot s_{_{MinMax}}(R|q)}{1.4} \cdot s_{_{MinMax}}(d|q), \qquad (1)$$

P. Serdyukov et al. (Eds.): ECIR 2013, LNCS 7814, pp. 752–755, 2013.

where $s_{MinMax}(R|q)$ denotes the relevance of the source R to a query q and is itself MinMax-normalized to the $[0, 1]$ range. The constant 0.4 shows how much importance is given to resource selection scores [2].

In this work we treat the importance of source scores as a parameter and rewrite Eq. 1 as follows:

$$s_{norm}(d|q) = \frac{1 + \lambda \cdot s_{MinMax}(R|q)}{1 + \lambda} \cdot s_{MinMax}(d|q). \quad (2)$$

Depending on λ, Eq. 2 defines a family of techniques. When $\lambda = 0$, i.e. no importance is given to resource selection scores, Eq. 2 simplifies to the standard MinMax. The λ between zero and infinity defines a range of intermediate methods including the original CORI ($\lambda = 0.4$). When $\lambda \to \infty$, Eq. 2 turns into the direct weighting of MinMax by source scores. The latter case is particularly interesting and gives the following non-parametric formula, which we call it *weighted MinMax* here to distinguish from the original CORI technique:

$$s_{norm}(d|q) = s_{MinMax}(R|q) \cdot s_{MinMax}(d|q). \quad (3)$$

Source scores $s_{MinMax}(R|q)$, calculated at the resource selection phase, represent the relevance of each source to a given query. Therefore, it is natural to weigh document scores by $s_{MinMax}(R|q)$ itself and not by its transformation (e.g. linear in the case of CORI).

MinMax ($\lambda = 0$) can be seen as sitting on the one end of CORI's performance spectrum, making the strongest assumptions and representing the lower bound of CORI's possible performance. The weighted MinMax ($\lambda \to \infty$) is sitting on the other end, relaxing MinMax's assumptions and achieving the best accuracy. All other values of the parameter ($0 < \lambda < \infty$) give CORI implementations that lay in between the two extremes. Our experiments support this intuition and show that CORI performance increases with λ, reaching the maximum when λ goes to infinity (i.e. when Eq. 3 is used).

3 Experiments

Experimental Setup. In this work we use three state-of-the-art distributed retrieval testbeds—gov2.1000, gov2.250 and gov2.30—that are the different splits of the TREC GOV2 dataset [1]. They consist of 1000, 250 and 30 sources respectively. The titles of TREC topics 701-850 are used as queries. We process the top-10 documents from each result list. ReDDE is used for resource selection [7].

We use ten retrieval functions implemented by the Terrier toolkit, namely, BM25, tf-idf (Terrier and Lemur versions), language modeling (original and with Dirichlet smoothing), and a number of DFR-based functions (BM25, BB2, IFB2, InL2 and PL2). Retrieval functions are randomly assigned to sources.

Note that this setup is different from the one used by Callan in [2] to evaluate CORI. In particular, (i) we use ReDDE resource selection, (ii) sources run 10 different retrieval functions, and (iii) we use larger Web-based testbeds. Therefore, the results below can be seen as complementary, rather than contradictory, to those in [2]. As a future work we plan to investigate how the above implementation decisions affect the optimal value of the parameter λ.

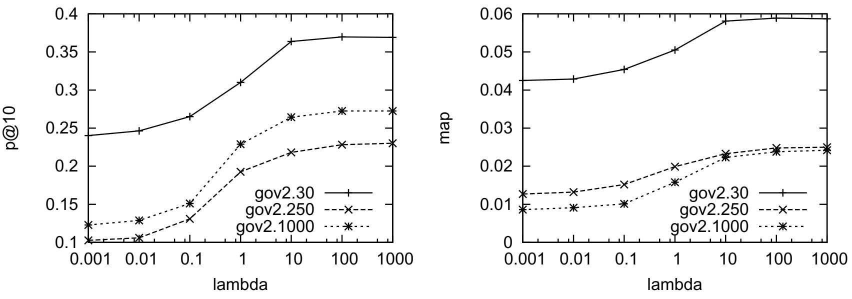

Fig. 1. CORI's P@10 and MAP for varying λ

Varying Lambda. Fig. 1 shows that both P@10 and MAP increase with λ (on log scale) across all testbeds. For example, P@10 increases from 0.24 for $\lambda = 0.001$ to 0.37 for $\lambda = 1000$ (54% improvement) for the gov2.30 testbed, from 0.10 to 0.23 (130%) for gov2.250 and from 0.12 to 0.27 (125%) for gov2.1000 (all improvements are statistically significant). MAP experiences the same trend. Note though that MAP values are very low due to discarding many sources on resource selection phase.

These results support our intuition that MinMax and weighted MinMax sit on different ends of CORI's performance spectrum (with the original CORI being in between the two) and that MinMaxed document scores should be weighted directly by source scores to achieve the best performance.

MinMax, CORI and Weighted MinMax. Tab. 1 shows the performance of the three representative methods, i.e. MinMax ($\lambda = 0$), CORI ($\lambda = 0.4$) and weighted MinMax ($\lambda \to \infty$). We present the results when no resource selection is performed and when only 10 and 3 sources are selected by ReDDE. We do not report MAP here (but only P@10 and P@20), because it is not comparable across different settings due to varying result list lengths.

On the one hand, MinMax is very much affected by resource selection: the less sources are selected, the better MinMax performs. CORI experiences a similar problem but to a much lesser extent, thus mainly overcoming the deficiencies of MinMax. On the other hand, the performance of weighted MinMax does not depend on how many sources are selected: it is almost the same across all settings. This is a desirable behavior of score normalization, as we do not want it to be affected by the number of result lists considered. Overall, the results in Tab. 1 suggest that the weighted MinMax modification should be preferred to the original CORI method in the considered settings.

Overall Performance. We also compare the above methods to the state-of-the-art results merging and score normalization techniques, namely, SAFE [6] and HIS [3]. On the one hand, Tab. 1 shows that all methods achieve similar performance when 3 sources are selected, because these 3 sources contain many relevant documents. On the other hand, most methods, apart from weighted MinMax and to some extent CORI, fail to work on the gov2.1000 testbed when no resource selection is performed, because relevant documents are sparse. Overall, weighted MinMax appears to be the most stable and best performing technique. It is agnostic to resource selection and outperforms other methods in all cases, with mostly statistically significant differences.

Table 1. Performance of score normalization and results merging methods. Best values are given in bold. † denotes statistical significance at the 0.01 level, ‡ at 0.05.

		gov2.30		gov2.250		gov2.1000	
		p@10	p@20	p@10	p@20	p@10	p@20
no selection	MinMax	0.197	0.194	0.022	0.025	0.016	0.015
	CORI	0.291	0.230	0.173	0.099	0.188	0.113
	W-MinMax	**0.366**†	**0.280**†	**0.232**†	**0.176**†	**0.271**†	**0.221**†
	SAFE	0.188	0.170	0.166	0.149	0.070	0.072
	HIS	0.173	0.185	0.093	0.093	0.082	0.082
10 sources	MinMax	0.215	0.206	0.129	0.128	0.137	0.128
	CORI	0.272	0.218	0.189	0.139	0.197	0.154
	W-MinMax	**0.366**†	**0.278**†	**0.231**†	**0.176**†	**0.271**†	**0.223**†
	SAFE	0.195	0.170	0.134	0.113	0.107	0.119
	HIS	0.194	0.188	0.168	0.144	0.177	0.155
3 sources	MinMax	0.334	0.275	0.208	0.166	0.239	0.197
	CORI	0.340	0.277	0.211	0.166	0.241	0.199
	W-MinMax	**0.364**‡	**0.281**	**0.231**‡	**0.170**	**0.268**	**0.203**
	SAFE	0.269	0.249	0.199	0.167	0.248	0.198
	HIS	0.305	0.280	0.192	0.166	0.207	0.188

4 Conclusions

In this work we parameterized CORI and studied its behavior with respect to the parameter λ. In the experimental setup considered, CORI achieved the best performance when document scores were weighted directly by source scores ($\lambda \rightarrow \infty$). In this case CORI significantly outperformed other state-of-the-art results merging and score normalization methods. As a future work we plan to study how the implementation decisions affect the optimal value of λ. We also plan to investigate if strengthening the effect of source scores during normalization (as opposed to dumping it with λ) can further improve CORI performance.

Acknowledgments. The authors would like to thank Jamie Callan and Mostafa Keikha for their feedback on this work.

References

1. Arguello, J., Callan, J., Diaz, F.: Classification-based resource selection. In: Proceedings of the ACM CIKM, pp. 1277–1286. ACM (2009)
2. Callan, J.: Distributed Information Retrieval. In: Advances in Information Retrieval, ch. 5, pp. 127–150. Kluwer Academic Publishers (2000)
3. Fernández, M., Vallet, D., Castells, P.: Using historical data to enhance rank aggregation. In: Proceeding of the ACM SIGIR, pp. 643–644 (2006)
4. Lee, J.H.: Analyses of multiple evidence combination. In: Proceedings of the ACM SIGIR, pp. 267–276. ACM (1997)
5. Markov, I., Arampatzis, A., Crestani, F.: Unsupervised linear score normalization revisited. In: Proceedings of the ACM SIGIR, pp. 1161–1162 (2012)
6. Shokouhi, M., Zobel, J.: Robust result merging using sample-based score estimates. ACM Trans. Inf. Syst. 27(3), 1–29 (2009)
7. Si, L., Callan, J.: Relevant document distribution estimation method for resource selection. In: Proceedings of the ACM SIGIR, pp. 298–305 (2003)