Using RankBoost to Compare Retrieval Systems

Huyen-Trang Vu, Patrick Gallinari
Laboratory of Computer Science (LIP6), University Pierre and Marie Curie
8, rue du capitaine Scott - 75015 Paris, France
vu@poleia.lip6.fr, gallinar@poleia.lip6.fr

ABSTRACT

This paper presents a new pooling method for constructing the assessment sets used in the evaluation of retrieval systems. Our proposal is based on RankBoost, a machine learning voting algorithm. It leads to smaller pools than classical pooling and thus reduces the manual assessment workload for building test collections. Experimental results obtained on an XML document collection demonstrate the effectiveness of the approach according to different evaluation criteria.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models; I.2.6 [**Artificial Intelligence**]: Learning—*parameter learning*

General Terms

Algorithms, Experimentation, Measurement

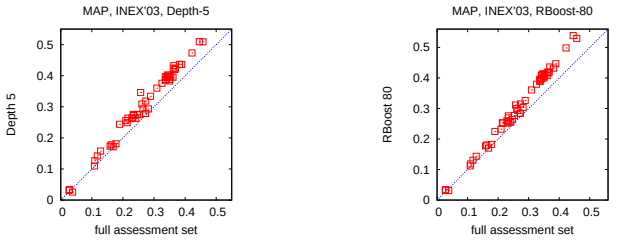
Keywords

Pooling, RankBoost, XML retrieval evaluation

1. INTRODUCTION

For evaluating retrieval systems, relevance assessments of test collections are essential. In order to reduce the manual workload, collection sampling - or pooling - of a limited number of documents to be assessed is performed. In TREC conference, the pool is created by gathering the top n documents (e.g. $n = 100$) from the much longer list delivered by each participating system for each topic. The collected documents are then judged by humans, documents outside the pool are assumed non relevant. This document subset is considerably smaller than the whole collection. However, as the document corpus becomes larger and larger, more robust pooling strategies are necessary. This is particularly true for recently emerging information retrieval (IR) paradigms like XML IR or Web Topic Distillation where very few documents are relevant due to the task nature.

This paper proposes a new pooling method based on RankBoost [2], a machine learning algorithm that selectively combines different rankings. At the same performance level, it allows to use smaller pools compared to the classical pooling method. The algorithm has several properties which make it specially suitable for the system pooling task. First, RankBoost operates on relative judgments, hence there is no need to normalize the relevance scales of the different retrieval systems. Second, it automatically selects systems to be combined and previous studies indicate that the chosen systems are highly independent one from the other. Finally, RankBoost is easy to use in practice, the only parameter to be set is the number of boosting iterations. The proposed method is general enough to be used in any *ad hoc* retrieval, particularly in the aforementioned situations. We present here tests performed on the INEX XML collection.



Depth-5 (0.900) RBoost-80 (0.932)

Figure 1: MAP of systems on Depth-5 pool (left) and on RBoost-80 pool (right) vs on the full assessment set. Kendall's τ value is in parentheses.

2. EXPERIMENTAL SETTING

INEX[1] has been the unique large scale forum for evaluating XML retrieval systems since 2002. Evaluation at INEX being performed at the level of XML elements, the assessing procedure is particularly heavy. This is a representative example which motivates the development of efficient pooling strategies.

The INEX'03 CO collection contains approximately 12,000 XML documents with 32 judged topics and the pool was built from 54 participating systems. Since the assessment workload depends much more on the number of documents the assessor has to check than on the number of XML elements, the baseline pool is limited to around 500 XML documents per topic [3]. For the pooling tests, we consider here a document retrieval scenario where a document is considered relevant if it contains a relevant element in the INEX sense.

RankBoost pooling will be compared to the classical round robin pooling method which has been used in TREC and also adopted in

[1] `http://inex.is.informatik.uni-duisburg.de/`

CIKM'05, October 31–November 5, 2005, Bremen, Germany.
ACM 1-59593-140-6/05/0010.

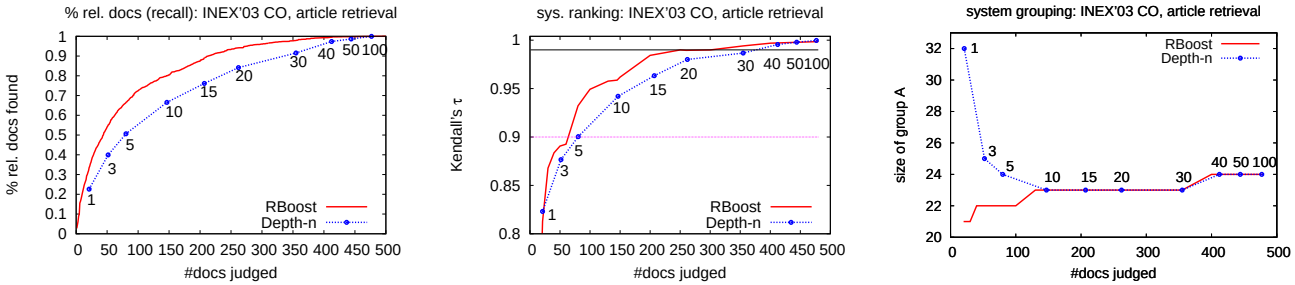


Figure 2: RBoost-m vs Depth-n: (a) % relevant documents found in shallow pools; (b) Kendall's τ correlation of system ranking wrt the full assessment set (two horizontals correspond to $\tau = 0.9$ and $\tau = 0.99$); (c) size of group A calculated by Tukey grouping on arcsine-transformed data. The figures along the dot curves are the n-values in Depth-n. The pool size is presented on the abscissa.

INEX. TREC-style pool of the first n documents retrieved by each INEX submission will be denoted Depth-n. This pooling produces on average a set of m documents per topic. The corresponding pool consisting of the top documents extracted by RankBoost will be denoted RBoost-m. System performance is quantified by mean average precision (MAP).

A training step is needed for RankBoost. The learning data set was collected from the Depth-5 pool. The average size of this pooling set is $m = 80$ per topic. We used the leave-one-out cross validation paradigm in evaluation: a model is trained with 31 topics and tested with the remaining one. There are 32 models in total. The learning process is iterated for 1000 RankBoost rounds like in [2].

Figure 1 illustrates the scatterplots of system scores with the pooling methods Depth-5 and RBoost-80 compared to the scores calculated from the original pool. The scatterplot alone is however not easily interpreted, we need more synthetic statistics for that. Unfortunately, there is no standard way to assert the quality of a pool sample. Three indicators are taken into consideration here. The first is the pool recall, that is the ratio of relevant documents found in the small pools wrt those in the original assessment set (Figure 2.a). The second is Kendall's τ - the correlation value of system ranking produced by a small pool with that by the original assessment set (Figure 2.b). Lastly, we use the discrimination power suggested in [4], namely the maximum number of top ranked systems which are not statistically different according to a multicomparison test (Tukey test in this case). This quantity, called size of group A, is shown in Figure 2.c.

3. RESULTS

The recall curves in Figure 2.a show the higher performance of RankBoost for identifying relevant documents. This suggests that smaller pool size could be used with RankBoost compared to the classical pooling method. We note that the improvement here is not as significant as reported in TREC circumstances [1]. The nature of data collections and the higher recall values of Depth-n pools in our case may be the principal cause of the observed difference.

Figure 2.b reveals that the RankBoost system ranking is much more similar to the reference ranking than round robin ranking is. However, since Kendall's τ averages all item swaps between the considered list and the reference ranking, this quantity alone does not provide enough information about system comparison. A swap between adjacent systems is not important if their performances are not statistically distinct. It is however problematic if there is a swap of two really different runs, especially if they are located at the two ends of the ranking list. Deeper analysis in conjunction with the use of scatterplots like Figure 1 indicate that contrarily to the TREC case [1], there is no typical misranking associated with the top runs in the INEX collection for both pooling methods, even with small pools.

Figure 2.c shows another evidence in favour of RankBoost pooling. For small pools, the group A is much larger for round robin than for RankBoost. In other words, the latter produces much more discriminative pools than the former does.

4. CONCLUSION

Empirical results obtained on the INEX collection validate the feasibility of reducing the pool size, either by shallow TREC-style pools or by the RankBoost method, the latter being the best in our experiments. Since the INEX collection is relatively small, the increase obtained with RankBoost is however limited. The experiments also show that the top runs are quite homogeneous and any effective methods will probably have a similar behavior to these top runs. In order to enlarge the collection size for reliable system evaluation, our experimental results suggest adding more topics rather than increasing the pool size. Finally, our proposal should still be validated on other collections with very different characteristics (in terms of size, of relevant document proportion, etc) to confirm its effectiveness in practice.

Acknowledgements

Thanks to M.-R. Amini, N. Usunier for the source code of RankBoost, J. Blustein for IR-STAT-PAK, B. Piwowarski for proof-reading.

5. REFERENCES

[1] J. A. Aslam, V. Pavlu, and R. Savell. A unified model for metasearch, pooling, and system evaluation. In *CIKM'03*, pages 484–491, 2003.

[2] Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969, Nov. 2003.

[3] B. Piwowarski and M. Lalmas. Providing Consistent and Exhaustive Relevance Assessments for XML Retrieval Evaluation. In *CIKM'04*, Nov. 2004.

[4] J. Tague-Sutcliffe and J. Blustein. A statistical analysis of the TREC-3 data. In *Overview of the Third Text REtrieval Conference (TREC-3), NIST Special Publication 500-225*, pages 385–398, Apr. 1995.