Boosting Static Pruning of Inverted Files

Roi Blanco
IRLab, Computer Science Department
University of A Coruña, Spain
rblanco@udc.es

Alvaro Barreiro
IRLab, Computer Science Department
University of A Coruña, Spain
barreiro@udc.es

ABSTRACT

This paper revisits the static term-based pruning technique presented in [2] for ad-hoc retrieval, addressing different issues concerning its algorithmic design not yet taken into account. Although the original technique is able to retain precision when a considerable part of the inverted file is removed, we show that it is possible to improve precision in some scenarios if some key design features are properly selected.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Performance, Experimentation

Keywords

Pruning, indexing, efficiency

1. INTRODUCTION

Static pruning of inverted files is an attractive technique to reduce storage needs and query processing times. Pruning is intended to remove the *pointers* (term-document pairs) most likely not to affect query performance. The algorithm presented in [2] yields excellent results at keeping the top-k answers for a given query and scoring function (Smart's tf-idf), making this approach suitable for high-demanding environments. It involves two parameters, k and ϵ, that set the number of top-documents (k) that are scored the same (within an error of ϵ) whether the original or the pruned inverted file is used, for any query less than a certain size ($\frac{1}{\epsilon}$). Hence, the rankings produced are similar. In brief, the pruned inverted file is produced by discarding, for every term, the documents that score lower than a certain threshold. The cut-off value is set to $z_t * \epsilon$ where z_t is the k-th highest score for a given term. Results presented at [1] and [2] prove that the technique conveys excellent results for maintaining precision when a high quantity of the data is removed.

SIGIR'07, July 23–27, 2007, Amsterdam, The Netherlands.
ACM 978-1-59593-597-7/07/0007.

Table 1: Collections and Topics

Collection	size	Topics	# queries
TREC disks 4&5	2G	301-450+601-700	250
WT2g	2G	401-450	50
WT10g	10G	451-550	100

Our belief is that by issuing some key features, the technique can go further than just faithfully preserving the document ranking, and be able to increase precision values at some pruning levels and under certain conditions. We used a probabilistic-based scoring function (BM25) instead of Smart's tf-idf, addressed some features not considered in the original model, and identified such conditions through trec-style intensive evaluation.

2. EXPERIMENTS AND RESULTS

We experimented with three different document/topics sets, described in table 1, and measured P@10 and MAP. Terms were stemmed using Porter's algorithm. As well, we considered three types of queries: short (title only), medium (title + description) and long (title + description + narrative).

The k and ϵ values in the original description of the algorithm must be selected carefully, as they hold important properties regarding to ranking similarity before and after pruning; especially, ϵ should be low. In this paper we consider k and ϵ just as parameters that determine the number of pointers removed (percentage of pruning), without focusing on theoretical guarantees. Choosing different k values yields very correlated results, although higher values ($k = 30$) are preferable for obtaining better MAP values whereas lower ($k = 10$) values seem more adequate if a good behaviour at P@10 is desired, which goes accordingly with the top-k preserving property construction of the algorithm. Some of the other considerations we took into account are summarised next.

Terms with document frequency $df > N/2$ can be discarded from the inverted file, where N is the total number of documents. This value comes naturally from BM25's idf formula, $\log\left(\frac{N-df+0.5}{df+0.5}\right)$, as those terms have a negative score for every document. It turned out that ruling out terms

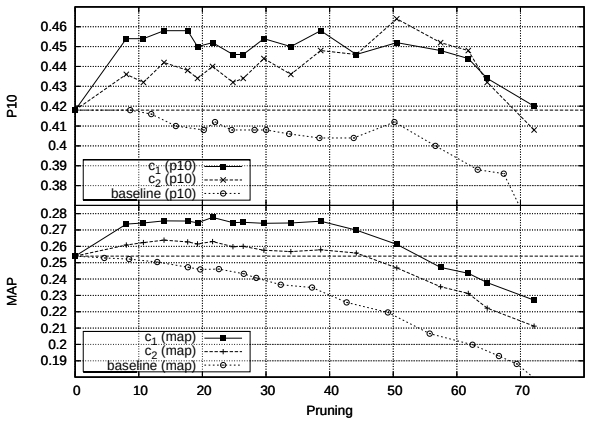

Figure 1: MAP and P@10 for short queries at different pruning levels, baseline and different settings (WT2g collection)

with $df > N/2$ is always a good option. This holds true especially for long queries, where those terms are more likely to appear.

In the original work ([2]) it is stated that every score must be shifted, otherwise the pruning level obtained is negligible. This shifting is done by subtracting the global minimum score. As we considered ϵ as just a parameter, we omitted the shifting step.

Following the description of the *idealised* pruning algorithm in [2], document lengths (dl) should be the same in both the original and the pruned inverted file. However, it makes sense to update them in order for the inverted file to be *coherent*. Experiments demonstrated that the best pruning and precision tradeoffs are obtained when the document lengths are updated.

A last design consideration was whether to update the average document length *avgdl* in the pruned inverted file or not. Correcting this value portrays a new term frequency (tf) normalisation factor. Updating *avgdl* performs slightly better than not doing it for long queries, whereas for short queries updating is outperformed by not doing it. Therefore, the new term frequency normalisation introduced by the algorithm seems adequate for a short-queries scenario.

To be more concrete, we present next some of the precision vs. pruning results using standard MAP and P@10 measures. To illustrate the effects of some of these design considerations figure 1 shows three precision curves obtained using the WT2g collection and short queries. Our baseline is the result using BM25, not updating the dls, not updating the *avgdl* and not removing terms with $df > N/2$ (this setting is the one that retains better the original ranking). The other two curves are obtained removing terms, updating the dls (the best setting found empirically), and c_2 updates the *avgdl* while c_1 does not. With the baseline settings, the method is better for maintaining the top-k results, which is easier for shorter queries, but the precision curves are decreasing. On the other hand, a different parameter selection improves significantly over the original precision values, up to a 50% pruning level for MAP and 70% for P@10.

Overall, using a suitable combination of parameters pruning is able to improve MAP and P@10. Considering the precision values obtained with the original not pruned inverted file as our baseline, it can be retained up to a 35-50% pruning level in most cases, although the method tends to favour short queries and the P@10. In that case, MAP(P@10) is over the baseline up to a 30-50%(60-70%) pruning level. Maximum improvements go up to 12% for MAP and 10% for P@10. These values are collection-dependent and lower with long queries. The behaviour is better in web collections than in disks 4&5, where MAP improvements are very small, even though the original precision value can still be maintained up to a 40-60% pruning level.

Parameter selection is crucial, and different algorithm settings lead to totally different performances. P@10 presents a good behaviour with any query size, whereas MAP behaves better with short queries. Some parameter combinations (document length update, selective term removal) are consistently better than the rest, although the improvements are more noticeable in the web collections (WT2g especially). Finally, the curves are not always monotonically decreasing, and high pruning values may increase precision, probably due to the score function ranking high *bad* document-term pointers, which are removed at those pruning levels.

We end this section showing that the effect of pruning updating the dl and without updating the *avgdl* in BM25's score, is to alter the tf contributions in a particular fashion. The tf normalisation in BM25 is $tf_n = \frac{tf}{nf}$ where $nf = (1-b) + b\frac{dl}{avgdl}$, and $b \in [0..1]$ is a constant (typically 0.75). If α terms are removed from a document by the pruning algorithm, the new normalisation factor would be $nf' = nf - b\frac{\alpha}{avgdl}$. This has the global effect of softening the dl contribution, and it can be compared with selecting a lower b value. In this case, $b' = b - \beta$ which implies that $nf' = nf - \beta(\frac{dl}{avgdl} - 1)$, and hence both normalisation factors have the same analytical form for long documents ($dl > avgdl$), and different otherwise.

3. CONCLUSIONS

In this work we outlined and experimented with several new variants of the most well-known inverted file pruning algorithm [2], and stated how it is possible to tweak the technique so that the precision is improved when some information is removed selectively. As a main conclusion, discarding pointers from an inverted file off-line and independently of the queries, is able to devise satisfactory results, efficiency and effective-wise, for ad-hoc retrieval. As well, carefully addressing the algorithmic issues involved brings some new intuitions regarding to weighting schemes or term frequency normalisation.

Acknowledgements. The work reported here was co-funded by SEUI and FEDER under project MEC TIN2005-08521-C02 and "Xunta de Galicia" under project PGIDIT06PXIC10501PN.

4. REFERENCES

[1] D. Carmel, E. Amitay, M. Herscovici, Y. S. Maarek, Y. Petruschka, and A. Soffer. Juru at TREC 10 - Experiments with index pruning, 2001.

[2] D. Carmel, D. Cohen, R. Fagin, E. Farchi, M. Herscovici, Y. Maarek, and A. Soffer. Static index pruning for information retrieval systems. In *Proc. of SIGIR 2001*, pages 43–50.