Selection bias in the LETOR datasets

Tom Minka
Microsoft Research
7 JJ Thomson Avenue
Cambridge, U.K.
minka@microsoft.com

Stephen Robertson
Microsoft Research
7 JJ Thomson Avenue
Cambridge, U.K.
ser@microsoft.com

ABSTRACT

The LETOR datasets consist of data extracted from traditional IR test corpora. For each of a number of test topics, a set of documents has been extracted, in the form of features of each document-query pair, for use by a ranker. An examination of the ways in which documents were selected for each topic shows that the selection has (for each of the three corpora) a particular bias or skewness. This has some unexpected effects which may considerably influence any learning-to-rank exercise conducted on these datasets. The problems may be resolvable by modifying the datasets.

1. INTRODUCTION

For the Learning to Rank workshop at SIGIR 2007, a dataset (actually a group of three datasets) was released for experimental purposes [6]. The intention was to provide some set of standard benchmarks, and to encourage participants to conduct comparable experiments (comparable to the benchmarks and to each other). The benchmark results provided in the cited paper are for Ranking SVM and RankBoost.

The three LETOR datasets were extracted from two TREC datasets (TDT2003, TDT2004) and from the OHSUMED corpus. Each dataset consists of a set of topics, together with extracted features for each of a set of query-document pairs, and associated relevance judgements. A number of different features are provided, both low-level and relatively high-level (features which might themselves serve as simple ranking algorithms). For the purpose of this paper, we note that one of the provided features is the BM25 score, which is a well-established ranking algorithm in its own right [7]. Although there is clearly no guarantee to this effect, we might expect the BM25 score on its own to give at least a reasonably effective ranking.

The set of documents associated with each topic is a selection, not the whole original corpus. There are obvious practical reasons for this procedure; however, the selection methods have given rise to a skew in the judgements which calls into question the validity of at least some of the results obtained on this dataset. This selection bias is present not only in the LETOR training data but also the test data. Therefore the algorithms which give the best test results are the ones which output rankings consistent with this bias. As a result, performance on the LETOR datasets is not an accurate guide for choosing a ranking algorithm for a real-world problem.

There are two issues here. On the one hand, we are concerned with ranking algorithms, and with evaluating such algorithms. On the other, we are concerned with learning algorithms, and their use in learning ranking algorithms. We discuss the effects of the skewed selection methods on both tasks. We note also that the LETOR datasets come with pre-defined training and test splits, and evaluation scripts. These scripts evaluate a ranker (possibly but not necessarily one trained or learnt using the training data) on the test data. They encode assumptions about (for example) how to deal with unjudged documents in evaluation.

1.1 Related work

The issue raised in the present paper can be seen as relating to the issue of evaluation with incomplete judgements, which has been the subject of much recent work (e.g. [8]). The traditional way to deal with incomplete judgements has been to regard unjudged documents as not relevant; this is probably a fair assumption if the original judgements were obtained from complete assessment of large pools, obtained from a wide variety of systems/runs. Some proposals involve leaving out the unjudged documents altogether. Recent work has addressed the issue of evaluation where this assumption is not good, and also the case where documents can be selected for judgement (so the task is to select for judgement those documents that are most likely to be informative, e.g. [3]). Some work has also addressed the possible bias in judgements (e.g. [2]). The approach in the paper just cited, as in [1], is to estimate the relevance of the unjudged documents, and include them in the evaluation.

Generally this work has not yet addressed the question of learning or training with incomplete or biased judgements (a recent exception is [4]). In constructing the LETOR datasets, a prior selection of documents has been made, with the effect that some assumptions about appropriate ways to deal with incomplete or biased judgements have been built into the datasets. The particular selection methods, and therefore the built-in assumptions, differ between the different LETOR datasets.

We note also that there has been some work in the machine learning literature on learning with 'imbalanced' or 'skewed' datasets. However, this is a different problem: typically learning a binary classifier in the case where one of the two classes occurs very much more frequently than the other (again typically, both in training-and-test datasets and in the real world). The problem discussed in the present paper has to do with the selection of data, for training-and-test, from real world data with different characteristics. (One rather obvious form of solution to the problem, that may be discovered in the machine learning literature, is discussed in section 4.1.)

2. THE DATASETS

2.1 TDT

In the LETOR TDT dataset, the documents selected for each topic include (a) the top 1000 documents ranked by BM25, with relevance judgements where these are available, plus (b) any other documents judged relevant. An immediate bias is evident: documents with high BM25 scores are selected anyway, irrespective of relevance; but documents with low BM25 scores are selected *only if they are relevant*.

The effect of this selection policy on the apparent effectiveness of BM25 as a ranking algorithm/feature is dramatic. It means that within these extractions, BM25 is negatively correlated with relevance. If using BM25 on its own as a ranking algorithm, it is best to select documents with low BM25 over documents with high BM25 scores. A ranking in reverse BM25 order is not only more effective than a ranking in positive BM25 order, it is also more effective at early ranks than either of the other benchmarks (Ranking SVM or RankBoost) in [6] – see table 1 for the TDT2003 dataset. All the tables and results below are as reported by the LETOR evaluation scripts, and in particular on the test split of the datasets.

	Reverse BM25	RankBoost	RankSVM	BM25
P@1	0.52	0.26	0.42	0.12
P@2	0.40	0.27	0.35	0.13
P@3	0.35	0.24	0.34	0.16
P@5	0.27	0.22	0.26	0.15
NDCG@5	0.326	0.279	0.347	0.183
MAP	0.185	0.212	0.256	0.126

Table 1: Results for ranking in reverse BM25 order, compared to baselines, on the LETOR TDT2003 dataset

Any learning algorithm which chooses to rank in reverse order of BM25 is therefore being rewarded in the LETOR evaluation. In general, we have no way of knowing whether a learning algorithm acquired this bias from the training data or whether it is inherent to the algorithm. If the bias is inherent to the algorithm, then applying the same learning algorithm to real data may give very different results than those observed on LETOR.

Another way of reading the results in the table is as follows: if all the relevant documents are contained in the top 1000, then it is very unlikely that the 1000th is relevant; thus a topic with this condition probably contributes zero to the P@1 figure for reverse BM25. From which it follows that up to 52% of the test topics have at least one relevant outside the top 1000. The P@2 row suggests that a much smaller proportion, probably around 28% (because 40% is the average of 52% and 28%), have at least two relevant outside the top 1000. In fact, these figures are exactly correct: 26 out of 50 topics have at least 1001 documents, while 14 out of 50 have at least 1002. For NDCG@5, Reverse BM25 does less well than RankingSVM (although still better than RankBoost); this probably has to do with the distribution of total numbers of relevant documents per topic, which correlates differently with effectiveness for the different methods. Reverse BM25 does worse than either baseline on MAP, which takes account of the entire curve, although its early-rank performance is enough to keep it above BM25 itself.

In the case of TDT2004, regular BM25 does much better and Reverse BM25 not nearly so well. The incidence of extra relevant documents outside the top 1000 (by BM25) is very much lower: the proportion of test topics with at least 1001 documents is 14/75 = 19%, and for 1002 is 6/75 = 8% (average 13%). Again, these figures are reflected exactly in the early-rank results for Reverse BM25 – see table 2. The effect is still enough to give Reverse BM25 a non-negligible early-rank precision.

	Reverse BM25	BM25
P@1	0.19	0.31
P@2	0.13	0.29
P@3	0.10	0.26
P@5	0.06	0.23
NDCG@5	0.097	0.319
MAP	0.060	0.282

Table 2: Results for ranking in reverse BM25 order, compared to baselines, on the LETOR TDT2004 dataset

2.2 OHSUMED

The OHSUMED dataset presents different issues. The documents selected for each topic were just those for which relevance judgements were available. When the original OHSUMED dataset was constructed [5], expert searchers conducted the searches for each topic, using a traditional Boolean search system. Subsequently, relevance assessment was done by another set of expert physicians. The pools of documents provided for assessment were constructed from those items viewed by the expert searchers, together with those items retrieved by searchers' final refined Boolean search statements.

This selection meant that documents in the pool had a high chance of being relevant. This is evident from the proportion of non-relevants among the selected documents – nine topics have less than 40% in the non-relevant category. Furthermore, every document in the pool matched some version of the query very well. Thus these non-relevant documents are highly atypical. Examples of the wider range of non-relevant documents that clearly exist in the full collection are missing in the dataset. In particular, there are likely to be many other documents that could be scored relatively highly by a ranking algorithm (which one would particularly like the algorithm to learn to distinguish).

3. LEARNING

As indicated, the LETOR datasets contain a number of features for each topic-document pair. The objective is to allow a learning method to learn how these features should be combined in order to provide optimal ranking according to some measure of search effectiveness. Such combination might for example be a linear function with learnt weights, or some more complex combination. In this section we discuss the impact of the skewed judgements on learning.

Suppose that, in addition to the BM25 feature, we add the log of BM25 to the TDT dataset as a separate feature (this feature is already present in the OHSUMED data for example). This means that even a simple linear model can actually learn a class of non-linear functions of BM25, by combining these two features linearly. We note that BM25 itself with a positive weight, combined with log BM25 with a negative weight, can yield a U-shaped function where very low as well as high BM25 scores are rewarded. We have indeed found this kind of effect when adding log BM25 as a new feature and then fitting a linear model on the TDT datasets. It seems clear that the effect is an artefact of the dataset.

This artefact can arise even without explicitly adding non-linear functions of BM25. This is because many standard IR features are correlated with BM25. In the TDT dataset, the features "sitemap based score propagation" and "sitemap based feature propagation" have a correlation coefficient with BM25 exceeding 0.98. Like BM25, these features perform best when given negative weights on TDT2003. However, a linear combination of BM25 with these features, using weights of opposite signs, provides increased performance due to the effective nonlinearity. Language modelling functions would presumably also have a high correlation with BM25, although in the TDT dataset these were not included at the whole-document level.

It may be argued that the LETOR dataset is intended to compare learning algorithms, and that it may serve this purpose even if the resulting learnt ranker is not a useful one. In this sense, it may serve a similar purpose to a purely artificial dataset, for which one has no guarantee that it reflects any real-world data. However, this seems a very limited aspiration for LETOR, and one that would indeed be better served by generating purely artificial data from known distributions. The fact that some attempt has been made to draw LETOR data from realistic datasets should be one of its advantages. We note also that it is difficult to draw useful conclusions about the value of learning algorithms in discovering good rankers for search, if what the learning algorithm learns is so dominated by selection biases in the dataset.

4. POSSIBLE SOLUTIONS

It is clear that in order to learn how to rank for real, or even to test ideas about learning properly, we need more realistic datasets. Results on the present LETOR datasets cannot be relied upon to yield believable research conclusions.

But the challenge of designing really good datasets for the learning to rank task is not simple. We may be able to suggest some modifications to the LETOR datasets which have some chance of making them more useful, but we also suggest that some serious investigation of the validity of results from any proposed dataset is required. The danger (as revealed above) is that a learning system will succeed only in learning artefactual characteristics of the dataset.

4.1 TDT

A simple way to remove the bias in the TDT datasets is to remove the relevant documents outside the top 1000 of BM25. This redefines the learning task as 'learning to rank within the results returned by another search engine', defined for these purposes as the top 1000 retrieved by a BM25 search engine. The effect of this on reverse BM25 is dramatic: its P@5 and NDCG@5 drop to zero and its MAP is nearly zero. Regular BM25 has its NDCG and MAP slightly increased. One problem with this approach is that it still gives special status to BM25, and indeed by extension to any ranking algorithm that is highly correlated with BM25.

Rather than remove relevant documents, we could also add more non-relevants from the original TDT collection. How many such documents should be included, and where should they be sampled from? For the number, we might assume that precision declines with rank in the BM25 ranking (again, this seems to be loading too much onto BM25, but again it's hard to see an alternative). This assumption would imply that (for example) the total number of non-relevant beyond rank 1000 should be fixed to ensure that the precision of this set alone is less than (say) the precision of ranks 900-1000 alone.

In order for the declining-precision argument to apply at any rank, it would be necessary to generate a much larger ranking in BM25 order, locate the relevant documents within it, and sample non-relevants from each interval between relevant documents. Such a procedure could probably be worked out, but would have to deal with some special cases:

- two relevant documents occurring close together in the ranking, or even tied;
- relevant documents with zero BM25.

The latter case does indeed occur in the LETOR datasets.

4.2 OHSUMED

The OHSUMED dataset is somewhat more tricky, since we have very little idea (certainly no formal definition) of how the included documents were selected for judgement in the first place. It might be better to replicate the TDT procedure, and introduce an algorithmic ranking such as BM25 as the basis for selection. Once again, we would have to sample in some systematic way in the gaps between the selected documents. In this case the selection includes non-relevant documents already; we would assume that the additional random documents are also non-relevant. This is perhaps a questionable assumption in the case of the OHSUMED data.

4.3 Redefining the test set

One of the issues that led to the construction of the LETOR datasets, and in particular the selection of documents for each topic, is that training on a full-size corpus is often not feasible. Many learning algorithms from the machine learning domain would be impossible to scale to operate on complete collections of documents (even TREC collections, let alone the web). However, this constraint does not apply to testing/evaluation. Most reasonable ranking algorithms could without difficulty be applied to a full-sized TREC collection.

This suggests that we should have different kinds of training and test collection: the test collection should be the entire original document set (TDT or OHSUMED as appropriate). This would ensure that a learning algorithm would not be rewarded for learning the biases of the selection process (as is currently the case). On the contrary, there would be benefit to be gained by designing a learning algorithm which could take proper account of these biases in training, and thereby produce a ranking algorithm which would work well with unselected data. Furthermore, the tests would have similar validity to many current experiments on TREC-like test collections outside the LETOR context, which they do not currently have.

5. CONCLUSIONS

We have shown that the LETOR datasets exhibit skewed judgements which cast doubt on any results derived from them. The TDT datasets can be fixed in a simple way by excluding relevant documents outside the top 1000. It may also be possible to improve all the datasets by including some additional sampled documents, assumed non-relevant, in the per-topic extractions. A more radical suggestion is to redefine the test part of LETOR to match much more closely the way in which ranking algorithms are normally tested on TREC-like corpora, using the entire corpus.

6. REFERENCES

[1] J. A. Aslam and E. Yilmaz. Inferring document relevance from incomplete information. In *CIKM '07: Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 633–642, New York, NY, USA, 2007. ACM.

[2] S. Buttcher, C. L. A. Clarke, P. C. K. Yeung, and I. Soboroff. Reliable information retrieval evaluation with incomplete and biased judgements. In *SIGIR 2007*, pages 63–70. ACM Press, 2007.

[3] B. Carterette, J. Allan, and R. Sitaraman. Minimal test collections for retrieval evaluation. In E. N. Efthimiadis, S. T. Dumais, D. Hawking, and K. Järvelin, editors, *SIGIR 2006: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 268–275, New York, 2006. ACM Press.

[4] B. He, C. Macdonald, and I. Ounis. Retrieval sensitivity under training using different measures. 2008. To appear in SIGIR 2008.

[5] W. R. Hersh, C. Buckley, T. J. Leone, and D. H. Hickam. OHSUMED: an interactive retrieval evaluation and new large test collection for research. In W. B. Croft and C. J. van Rijsbergen, editors, *SIGIR '94: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 192–201. Springer-Verlag, 1994.

[6] T.-Y. Liu, J. Xu, T. Qin, W. Xiong, and H. Li. Letor: Benchmark dataset for research on learning to rank for information retrieval. Technical report, 2007. LR4IR 2007, in conjunction with SIGIR 2007.

[7] K. Sparck Jones, S. Walker, and S. E. Robertson. A probabilistic model of information retrieval: development and comparative experiments. *Information Processing and Management*, 36:779–808 (Part 1) and 809–840 (Part 2), 2000. `http://www.soi.city.ac.uk/~ser/blockbuster.html`.

[8] E. Yilmaz and J. A. Aslam. Estimating average precision with incomplete and imperfect judgements. In P. S. Yu, V. J. Tsotras, E. A. Fox, and B. Liu, editors, *CIKM 2006: Proceedings of the 13th ACM Conference on Information and Knowledge Management*, pages 102–111, New York, 2006. ACM Press.