Usefulness of Quality Click-through Data for Training

Craig Macdonald, Iadh Ounis
Department of Computing Science
University of Glasgow, Scotland, UK
{craigm,ounis}@dcs.gla.ac.uk

ABSTRACT

Modern Information Retrieval (IR) systems often employ document weighting models with many parameters that require to be appropriately set for effective retrieval performance. To obtain these parameter settings, quality training is usually required, where assessors have manually labelled the relevance of retrieved items for many queries. In this work, we examine the usefulness of high-quality click-through data for training an IR system, on searching the .gov vertical domain of the Web. We find that, compared to training using relevance judgements created using human assessors, the click-through trained settings are as good and occasionally better.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Experimentation, Measurement

Keywords

Training, Web, Queries & Click-through

1. INTRODUCTION

Most Information Retrieval (IR) systems have some parameters which affect the selection and ordering of results that are returned to the user, and hence affect the overall retrieval performance of the system. While these parameters can be left at their default value, there is often an improvement in terms of retrieval performance to be gained by tuning these parameters using a supervised or unsupervised training method [17]. Moreover, there has been much research done over recent years to develop new methods for training models with many parameters, for instance by attempting to directly optimise rank-based evaluation measures such as Mean Average Precision (MAP) or Mean Reciprocal Rank (MRR) [13]. In contrast, the RankNet technique described in [4] avoids generating entire rankings of documents, by using a cost function calculated on document pairs, that correlates with the effectiveness of an approach that directly maximises the normalised Discounted Cumulated Gain (nDCG) evaluation measure. This work is now seen as part of the wider Learning to Rank combined field of machine learning and information retrieval [10].

Traditionally, training to find a setting of the parameters involves maximising the retrieval performance of the retrieval system, using a suitable measure, on a set of training queries using the corresponding relevance judgements. The performance of the system can then be measured using a set of unseen queries - known as the test set. However, deriving relevance judgements is expensive, involving many man-hours by human assessors.

Instead, in this work, we examine how quality click-through data, obtained from a live Web search engine, can be used to train the parameters of another IR system designed for a vertical portion of the Web. In this way, we are utilising the data as training to improve our IR system, but no feature or ranking model is being directly learned from the query data. The use of click-through data in aggregate form means that no individual user is treated as absolutely correct, and instead, the behaviour of a larger number of users is utilised [1].

To assess the usefulness of the trained settings, we evaluate the final IR system on a selection of tasks on a test collection based on a vertical portion of the Web. In particular, we examine how queries for which users clicked into documents within the .gov Web domain can be used to train search engines on the GOV TREC Web collection [6].

The contributions of this work are twofold: we perform an analysis of the usefulness of sampling training data from a large query log. Three different sampling strategies are investigated, and results drawn across three user search tasks. The remainder of this paper is structured as follows: Section 2 discusses strategies for selecting training queries from a click-through log; Section 3 discusses a ranking strategy for Web IR settings which requires training, and describes the methodology used to obtain the parameter values from training data; Section 4 experiments with the usefulness for training of the discussed query log samples; We provide concluding remarks in Section 5.

2. SELECTING TRAINING QUERIES FROM CLICK-THROUGH LOGS

The ability to generate many queries for training purposes from a large query log permits investigations into which queries are best to sample. In [3], Broder classified Web

WSCD '09, Feb 9, 2009 Barcelona, Spain.

search queries into three broad categories: *Navigational* where the user's immediate intent is to reach a particular site. For example, the query "google" is likely to be looking for the Google home page; For *informational* queries, the user's immediate intent is to acquire some information assumed to be present on one or more Web pages, in a fashion closest to information seeking in classical IR; and finally *transactional* queries, where the user's intent is to perform some Web-mediated activity. The purpose of transactional queries is to reach a site where further interaction will happen, for example shopping. In [16], Rose & Levinson refined Broder's model by further categorising queries in the informational and transactional/resource categories. Moreover, they summarised the quantity of these query categories across several studies, as approximately 60% informational, 25-35% transactional and 15-25% navigational. However, the most frequent queries are often navigational [8].

In academic research, the Text REtrieval Conference (TREC) Web tracks investigated user retrieval tasks in the Web setting (e.g. [5, 6]). During the course of the Web tracks, three user search tasks were defined, namely the home page finding task and the named page finding task (both representing different kinds of navigational user search task: in home page finding, the user is looking for the home page of a particular Web site; in named page finding, the user is trying to find a single particular page), and also the topic distillation task (an informational task, where systems should provide good entrance pages to relevant sites). For the TREC Web track tasks, each task forms a test collection comprising of a shared corpus of Web documents (the GOV corpus in this case), a set of queries, and corresponding binary relevance judgements made by human assessors.

However, building such relevance assessments is expensive, particularly if only used for training an IR system. In this work, we wish to automatically derive data to use when training. The subsequent performance of the IR system can then be measured using test data for which relevance assessments are available. Our training queries are drawn from MSN Search Asset Data collection, a 15 million query log (7 million unique queries) with click-through documents, itself sampled from the MSN Web search engine during May 2006.

While the test data may be broken up by task, our IR system does not attempt to predict query category, and hence any training data should consist of a mixture of query types. To derive the training data, we sample training queries from the query log, and assume that the documents clicked on by the users are relevant during the training-phase evaluations, while non-clicked documents are non-relevant. Given that the most frequent queries are usually navigational, these alone could be used as training queries, and it would be likely that the search engine could learn to rank these queries well. However, these queries are merely the head of the power-law distribution of query frequency in a query log, and by focusing on sampling these queries, could lead to failures for other forms of user intents. We formulate three strategies for sampling queries to use for training. In all strategies, we use the assumption that queries where users clicked on documents which are in the GOV Web test collection depict valid user needs for that vertical of the Web. Of the 7 million unique queries, 25,375 were found to have click-through to documents contained in the the GOV Web test collection. From these 25k queries, we sample queries for training using the described sampling methods:

Statistic	Head-first	Unbiased	Biased
Num. Queries	631	975	594
Ave. Terms per Query	1.77	3.15	1.98
URLs as Query	133	40	30
Ave. Clicked Documents per Query	1.24	1.15	1.51
Max Clicked Documents per Query	4	9	11

Table 1: Statistics of the query log samples that we use in this work. Note that domain names and URLs (e.g. www.irs.gov) were counted as single terms.

- **Head-First**. Ranking the most commonly clicked query-document pairs, we select the top 1000 pairs, providing the most frequent queries with click-through in GOV Web test collection.
- **Unbiased Randomly**. Select 1000 random queries from the query list (without repetitions) that have click-through in GOV, providing a random sample of both frequent and infrequent queries.
- **Biased Randomly**. Select 1000 random queries from the query list (with repetitions) that have click-through in GOV. The queries in this sample are more likely to be frequent.

In the first strategy, we derive the most common queries with their click-through documents. These queries are of value in training, as they will allow the search engine to accurately train for the frequent queries. In the unbiased sampling approach, we are treating all queries equally, on the assumption that this will contain a mix of various information need types. However, this approach is also likely to have too many low value (long-tail) queries, of less value for navigational task training. Finally, the biased sampling takes into account the frequency with which a query appears in the query log during sampling, so more frequent queries are more likely to become part of the training sets.

Table 1 details the statistics of our training samples. Firstly, note that duplicate removal causes the number of queries in the training sets to be reduced. For the unbiased sample set, this is due to some query normalisation such as character case. For the head-first query set, 1000 document-query pairs were selected, causing the number of queries to be significantly less. For the biased sample, more frequent queries were more likely to be selected, and hence duplicate queries were likely to occur. The number of queries that consist of URLs can be seen as an indicator of the proportion of queries with homepage intent in the training set. It is notable that this is highest for the head-first query set. We note that the unbiased sample has a higher average query length, suggesting that it contains more informational queries [7]. However, the biased sample has a higher mean and maximum number of documents clicked per query.

3. SEARCH ENGINE RANKING STRATEGY

In training from the click-through data, we are not learning features from the query and relevant documents directly, as exemplified by [1, 9], and examined in the learning to rank sub-field. Instead, based on the Terrier IR platform [14], our search engine uses textual features from the documents. In particular, in Web IR, the term frequency distribution in various parts of the document can be of importance - for instance, in the title of the documents, the content of the

document or even in the anchor text of the incoming hyperlinks. A field-based weighting model takes into account separately the influence of a term in each field of a document. In this section, we describe the field-based weighting model that we apply, and the manner in which it is trained.

While we could also integrate other document features, such as URL length or link analysis, we believe that the field-based weighting model described here is sufficient for this initial study, as it integrates both textual evidence, and implicit link evidence (in the form of anchor text surrogates).

3.1 PL2F Field-based Weighting Model

In this work, we use the PL2F field-based weighting model, which is a derivative of the PL2 Divergence from Randomness weighting model [2]. In these models, the relevance score of a document d for a query Q is:

$$score(d,Q) = \sum_{t \in Q} qtw \cdot \frac{1}{tfn+1}\big(tfn \cdot \log_2 \frac{tfn}{\lambda} + (\lambda - tfn)\cdot \log_2 e + 0.5 \cdot \log_2(2\pi \cdot tfn)\big) \quad (1)$$

where λ is the mean and variance of a Poisson distribution. It is given by $\lambda = F/N$. F is the frequency of the query term in the collection and N is the number of documents in the whole collection. The query term weight qtw is given by qtf/qtf_{max}, where qtf is the query term frequency. qtf_{max} is the maximum query term frequency among the query terms.

PL2 and PL2F differ in their definition of tfn. In the PL2F model, the document length normalisation step is altered to take a more fine-grained account of the distribution of query term occurrences in different fields. The so-called Normalisation 2 used by PL2 is replaced with *Normalisation 2F* [12], so that the normalised term frequency tfn corresponds to the weighted sum of the normalised term frequencies tf_f for each used field f:

$$tfn = \sum_f \left(w_f \cdot tf_f \cdot \log_2(1 + c_f \cdot \frac{avg\_l_f}{l_f}) \right), (c_f > 0) \quad (2)$$

where c_f is a hyper-parameter for each field controlling the term frequency normalisation, and the contribution of the field is controlled by the weight w_f. Together, c_f and w_f control how much impact term occurrences in a field have on the final ranking of documents. tf_f is the term frequency of term t in field f of document d, l_f is the number of tokens in field f of the document, while avg_l_f is the average length of field f in all documents, counted in tokens.

3.2 Training PL2F

The fields we use are content, title and anchor text of incoming hyperlinks, from which standard stopwords are removed, and Porter's stemmer applied. For ranking using these three fields, the PL2F field-based weighting model has 6 parameters: a weight for each field w_{body}, w_{anchor} and w_{title}, and the field normalisation parameters, namely c_{body}, c_{anchor} and c_{title} for PL2F. This high number of parameters infers that the model is likely to require training before use. We train the parameters using simulated annealing [11] to directly optimise a given evaluation measure on a training set of queries. However, to train all 6 parameters in one simulated annealing would be very time consuming and unlikely to obtain an effective setting. Instead, we take advantage of the independence of the field normalisation parameters c_f to perform concurrent optimisations for each, as also discussed in [15, 18]: while optimising a field normalisation parameter, the weights of the other fields are set to 0. Once settings for the field normalisation parameters for each field have been found, these are fixed, and the weights (w_f) for the three fields are trained using multi-dimensional simulated annealing[1]. Moreover, as simulated annealing only offers a probabilistic guarantee that the global maxima will be found, we repeat each simulated annealing three times, so that a stable, effective setting can be found by inspecting all outcomes.

4. EXPERIMENTS IN TRAINING USING CLICK-THROUGH

We wish to ascertain the suitability of click-through training data for training an IR system. To do this, we aim to show that the retrieval effectiveness of the PL2F retrieval model using the parameter settings obtained from the click-through training is comparable to that obtained when the parameters are trained from queries that have real human relevance judgements. While the click-through data is larger and more noisy than the available human relevance judgements, we will show that it is indeed as effective. In particular, to evaluate the effectiveness of the trained settings on a test set of queries, we employ the topic distillation (denoted td), named page (np), and home page (hp) retrieval tasks from the TREC Web tracks 2003 and 2004 [5, 6]. We report the MAP measure on the test set, but note that for the known-item retrieval tasks (hp and np), with a single relevant document, this equates to MRR.

In our experiments, our baseline system is trained using a mixed set of TREC Web task queries, with human relevance judgements. As mentioned above, our system does not attempt to classify queries, and hence training should be performed by a set of queries that reflect all tasks. In particular, we train with: mq2004 which consists of all of the queries from the td2004, np2004 and hp2004 tasks, and is used as training for the 2003 tasks; mq2003', which is a subset of the first 50 queries of each of td2003, td2003 and hp2003, suggested by [15] as training for the 2004 tasks. In both cases, we train to maximise MAP on the training set - for the known-item queries (hp and np), this corresponds to using MRR - i.e. the correct measure is used for each query type.

To test the click-through training data defined in Section 2, we report the achieved performances using trainings derived from each of the three samples of the click-through data. Note that for these samples, since we do not know the correct search task for each query, and hence the correct evaluation measure, we train using both MRR and MAP on the training sets. In the case where there is more than one document marked as 'relevant' for a query, we believe that MAP is likely to be a better training measure than MRR, as MAP responds to more changes in the ranking of documents.

All results are reported in Table 2, including statistical significance from the baseline using the Wilcoxon signed-rank test. We also report the mean of the six reported MAP values in each row, such that the overall trends are easily observed. From the results, we make the following observations on the use and training of PL2F using the click-through training data: In general, training on the click-through data is comparable to the TREC mixed task training. Indeed,

[1]Unlike, [18] we do not restrict $w_{body} = 1$.

Training Measure	td2003	td2004	hp2003	hp2004	np2003	np2004	mean
Trained on TREC mq2003'/mq2004							
MAP	0.1416	0.1453	0.7025	**0.6392**	0.6837	0.6796	0.4987
Trained on head-first click-through							
MAP	**0.1535**$^{=}$	0.1408$^{=}$	0.6862$^{=}$	0.5844$^{<}$	0.5619$^{\leq}$	0.5821$^{\leq}$	0.4515
MRR	0.1498$^{=}$	0.1415$^{=}$	0.6891$^{=}$	0.5873$^{<}$	0.5677$^{\leq}$	0.5811$^{\leq}$	0.4528
Trained on unbiased random click-through							
MAP	0.1477$^{=}$	0.1389$^{>}$	**0.7425**$^{\geq}$	0.6335$^{=}$	0.7065$^{>}$	**0.6945**$^{=}$	0.5106
MRR	0.1507$^{=}$	0.1416$^{=}$	0.7373$^{\geq}$	0.6259$^{=}$	**0.7067**$^{=}$	0.6741$^{=}$	0.5060
Trained on biased random click-through							
MAP	0.1491$^{>}$	**0.1506**$^{=}$	0.7147$^{=}$	0.6267$^{=}$	0.6899$^{=}$	0.6578$^{<}$	0.4981
MRR	0.1468$^{=}$	0.1479$^{=}$	0.7342$^{=}$	0.6236$^{=}$	0.6744$^{=}$	0.6511$^{<}$	0.4963

Table 2: Test MAP on various TREC test sets, for various training data. Statistical significance using the Wilcoxon Signed Rank test from the mq2004/mq2003' trained settings of PL2F are denoted using five symbols: $\leq$ and $<$, ($>$ and $\geq$) denote a significant decrease (increase) in retrieval performance with $p < 0.05$ and $p < 0.01$ respectively. $=$ denotes no significant different in retrieval performance.

only in 8 out of 36 cases is there a significant drop in retrieval performance using the click-through training compared to training on the TREC mixed query tasks. In 5 of the 36 cases, retrieval performance is significantly better. For the remaining 23 cases, the retrieval performance of the settings trained on the click-through data show no significant differences from the settings training using human relevance judgements.

Examining each of the three query log samples in turn, we note that the random samples are, in general, more effective than the head-first sample. In particular, the random unbiased sample performs best overall, mainly because of its high performance on the home page finding and named page finding tasks. Indeed, the results on these tasks are very promising compared to the TREC runs for the corresponding years [5, 6], even without the use of document features, such as URL length or link analysis. For topic distillation, there appears to be no clear best sample, while results are comparable to training on the TREC mixed query sets. Additional document features were previously shown to assist in improving performance on this task [15].

Lastly, comparing training measures, we note from the mean values, MAP appears to be marginally better for training using click-through. This is likely due to the high number of queries which have only one clicked document in the training set, inferring that MAP is very close to MRR in general. For instance, across all trainings conducted on the unbiased random sample, MAP and MRR were highly correlated, with $\tau = 0.988$ (for 2848 evaluations).

Overall, we conclude that for field-based weighting models, training using click-through appears to be sufficient for obtaining robust parameter settings that achieve as good retrieval performance as real training data, particularly for the home page and named page finding tasks. In this case, the quantity of training data seems to provide good settings in comparison to the setting trained on the smaller but higher quality human assessed TREC datasets.

5. CONCLUSIONS

In this paper, we examined the effectiveness of click-through data for training a retrieval system with parameters. While the click-through data was for a general Web search engine, it was found to contain sufficient click-through with which to train our vertical search engine. We sampled from these queries in three different ways, and then used these samples for training our field-based IR system. In doing so, we examine how suitable the click-through data is for training. Our results show that the click-through data is usually as good as training on bona fide relevance-assessed TREC dataset, and occasionally significantly better.

A possible disadvantage of the approach discussed here is that users will click on only the few top-ranked results, and hence this can cause bias when used for training [9]. In our case, we believe that the MSN search engine (from which the click-through data was obtained) was likely trained for various representative samples of queries, and therefore the results displayed to users and clicked on are likely to be relevant. Next, because two of our samples were random, repeating the experiments with different samples would enhance the reliability of the results derived here, while investigating the effect of the sample size on the effectiveness of the trained setting would also be worthwhile.

The TREC 2009 Web track will use a considerably larger test collection formed from a general crawl of the Web. From the experimental results in this paper, it seems likely that we can effectively train our document retrieval system for retrieval from the new Web collection, using samples derived from the entire query log used in this work.

This work contrasts from [1, 9] because, at this stage, we are only concerned with the training of the document retrieval component of our system. In the future, this work could be expanded to train many more features: document features such as link analysis and URL length; as well as directly learning features from the query logs for direct integration into the ranking strategy.

Acknowledgements

We wish to thank Microsoft for providing the query and click-through logs studied in this work. We also wish to thank Ben He for his helpful comments.

6. REFERENCES

[1] E. Agichtein, E. Brill, S. Dumais, and R. Ragno. Learning user interaction models for predicting Web search result preferences. In *Proceedings of SIGIR 2006*, pages 3–10.

[2] G. Amati. *Probabilistic Models for Information Retrieval based on Divergence from Randomness*. PhD thesis, Univ. of Glasgow, 2003.

[3] A. Broder. A taxonomy of Web search. *SIGIR Forum*, 36(2):3–10, 2002.

[4] C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *Proceedings of ICML 2005*, pages 89–96.

[5] N. Craswell and D. Hawking. Overview of TREC-2004 Web track. In *Proceedings of TREC 2004*.

[6] N. Craswell, D. Hawking, R. Wilkinson, and M. Wu. Overview of the TREC-2003 Web track. In *Proceedings of TREC 2003*.

[7] B. J. Jansen, D. L. Booth, and A. Spink. Determining the informational, navigational, and transactional intent of Web queries. *Inf. Process. Manage.*, 44(3):1251–1266, 2008.

[8] B. J. Jansen, A. Spink, and J. Pedersen. A temporal comparison of AltaVista Web searching. *JASIST*, 56(6):559–570, 2005.

[9] T. Joachims. Optimizing search engines using clickthrough data. In *: Proceedings of SIGKDD 2002*, pages 133–142.

[10] T. Joachims, H. Li, T.-Y. Liu, and C. Zhai. Learning to Rank for Information Retrieval. In *SIGIR Forum*, 41(2):58–62, 2007.

[11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[12] C. Macdonald, V. Plachouras, B. He, C. Lioma, and I. Ounis. Univ. of Glasgow at WebCLEF 2005: Experiments in per-field normalisation and language specific stemming. In *Proceedings of CLEF Workshop 2005*, LNCS vol 4022, pages 468–480.

[13] D. Metzler. Direct maximization of rank-based metrics. Technical report IR-425, Univ. of Massachusetts, 2005.

[14] I. Ounis, G. Amati, V. Plachouras, B. He, C. Macdonald, and C. Lioma. Terrier: A high performance and scalable information retrieval platform. In *Proceedings of the OSIR Workshop 2006*, Seattle, USA.

[15] V. Plachouras. *Selective Web Information Retrieval*. PhD thesis, Univ of Glasgow, 2006.

[16] D. Rose and D. Levinson. Understanding user goals in Web search. In *Proceedings of WWW 2004*, pages 13–19.

[17] Y. Yue, T Finley, F. Radlinski, and T. Joachims. A Support Vector Method for Optimizing Average Precision. In *Proceedings of SIGIR 2007*, pages 271–278.

[18] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft Cambridge at TREC-13: Web and HARD tracks. In *Proceedings of TREC 2004*.