Predicting Query Performance on the Web

Niranjan Balasubramanian
University of Massachusetts Amherst
140 Governors Drive, Amherst MA 01003
niranjan@cs.umass.edu

Giridhar Kumaran and Vitor R. Carvalho
Microsoft Corporation
One Microsoft Way, Redmond, WA
{giridhar,vitor}@microsoft.com

Predicting the performance of web queries is useful for several applications such as automatic query reformulation and automatic spell correction. In the web environment, accurate performance prediction is challenging because measures such as clarity that work well on homogeneous TREC-like collections, are not as effective and are often expensive to compute. We present Rank-time Performance Prediction (RAPP), an effective and efficient approach for online performance prediction on the web. RAPP uses retrieval scores, and aggregates of the rank-time features used by the document-ranking algorithm to train regressors for query performance prediction. On a set of over 12,000 queries sampled from the query logs of a major search engine, RAPP achieves a linear correlation of 0.78 with DCG@5, and 0.52 with NDCG@5. Analysis of prediction accuracy shows that *hard* queries are easier to identify while *easy* queries are harder to identify.

Categories and Subject Descriptors: H.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms, Experimentation, Theory

Keywords: Performance prediction, Query difficulty, Web search

1. RANK-TIME PREDICTION

Query performance prediction is the task of estimating the quality of the results retrieved for a query, using effectiveness measures such as normalized discounted cumulative gain (NDCG). Performance prediction is useful for various applications such as detecting queries with no relevant content, performing selective query expansion, and to merge results in a distributed information retrieval system [5]. However, most post-retrieval query performance predictors are expensive to compute, and not well-suited for the web.

The key idea behind RAPP is to use retrieval scores and features that are available to the retrieval algorithm during ranking. Our choice of features is based on two observations. First, the retrieval scores of the top-ranked documents are good indicators of document relevance and therefore are good estimators of retrieval effectiveness. Retrieval score-based features have previously been shown to be effective for classifying TREC queries as *easy* or *hard* [4]. Second, Web search engines use retrieval algorithms that combine query dependent and query-independent document features that are designed to capture relevance. The performance of a query is intimately related to these feature values. Since retrieval scores for different queries may not be directly comparable, we use statistical aggregates of the scores. Moreover, statistical aggregates such as maximum, mean, and standard deviation capture different aspects of the quality of search results. For example, our initial analysis showed that retrieval scores for low-performing queries tend to have low mean and high variance.

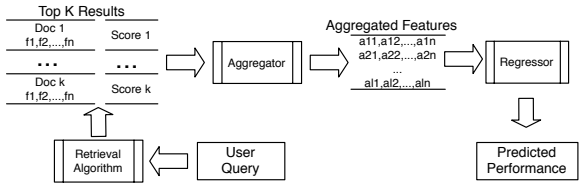


Figure 1: RAPP Overview.

Figure 1 illustrates RAPP. First, we retrieve the top k documents using the retrieval algorithm.We use the retrieval scores as well as the query-dependent and query-independent features of these top-ranking documents. We then compute statistical aggregates such as mean, maximum, standard deviation, variance, and coefficient of dispersion of these features. Finally, we use these aggregated features and the individual retrieval scores to train a regressor to predict a target performance measure.

2. EXPERIMENTS

To evaluate RAPP, we target the prediction of two performance measures commonly used in web search: DCG@5[1] and NDCG@5 (referred as DCG and NDCG henceforth). We use a set of 12,185 queries, which were obtained as a frequency-weighted random sample from the query logs of a major web search engine. For retrieval we use LambdaRank [2], an effective learning to rank algorithm for the web. For each query in our collection, we create feature vectors as follows. First, we use LambdaRank to assign scores and rank documents on the Web[2]. Our implementation uses several retrieval features such as BM25F-based features, click-based features, query length, and other query-independent features such as variants of PageRank. For each of these retrieval features we create statistical aggregates as listed in Section 1. Next, we select the top 100 aggregates (referred to as regression features henceforth) that have the highest linear correlation with the target metric on a set of training queries. Some example features include clickboost_max (maximum value of a click-based feature) and score_stdev (standard deviation of LambdaRank scores). Finally, we create a query performance prediction dataset by associating with each query, the performance metric, DCG@5 or NDCG@5 and the regression features. On this dataset, we conduct 3-fold cross-validation experiments to train linear as well as non-linear regressors based on the Random Forest algorithm [6][3].

Clarity comparison. We use Clarity [3], a competitive performance prediction technique, as an experimental baseline. To compute Clarity for a query, we use a query model built from the top

[1]Normalized by perfect DCG@5 to scale values to (0,1).
[2]LambdaRank was trained on an entirely different data set.
[3]We use the R package implementation with default parameters.

SIGIR'10, July 19–23, 2010, Geneva, Switzerland.
ACM 978-1-60558-896-4/10/07.

50 results returned by the search engine. Because Clarity computation is expensive, we calculated Clarity only for a random subset of 600 queries drawn from our original query set. Table 1 shows the results of performance prediction for DCG and NDCG using Clarity as well as selected features used in RAPP. Clarity achieves very low linear correlation with both DCG and NDCG. When compared to the performance of features used in RAPP, even the lowest performing individual feature outperforms Clarity. This suggests that while Clarity is a competitive measure in smaller TREC collections, it is not a well-suited for the Web.

Table 1: Clarity comparison: *Average* – the average correlation of RAPP features. *Best* and *Worst* – the highest and the lowest individual correlation of RAPP features on the entire set of 12,185 queries. *Clarity* correlation is measured on a subset of 600 queries.

Predicted Measure	Clarity	Average	Best	Worst
DCG	0.11	0.57	0.70	0.20
NDCG	0.10	0.27	0.50	0.17

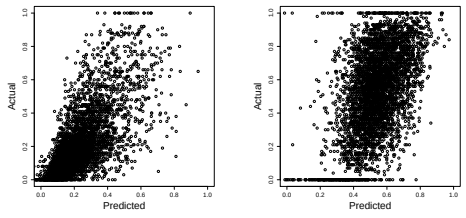
Table 2 shows the prediction accuracy for RAPP in terms of linear correlation and root mean squared error (RMSE). Both predicted DCG and NDCG values achieve a high linear correlation and low RMSE. Also, NDCG prediction is much worse as indicated by the low correlation and the higher RMSE values. This is mainly because NDCG is a non-linear metric that is calculated based on the actual number of relevant documents that exist in the collection. Thus NDCG cannot be estimated based on the features of the top-ranked documents alone. Finally, in terms of correlation and RMSE, there is little difference in prediction effectiveness between simple linear regression and the non-linear random forest based regression.

Table 2: RAPP Effectiveness: Corr. – Linear correlation measure. RMSE – root mean squared error.

Method	DCG		NDCG	
	Corr.	RMSE	Corr.	RMSE
Linear	0.78	0.13	0.50	0.23
Random Forest	0.79	0.13	0.52	0.22

The scatter plot in Figure 2(a) illustrate a strong correlation between the predicted and actual DCG values for one fold of the data. Figure 2(b) shows predicted NDCG values which are not as strongly correlated with the actual values. For DCG, when the actual values are less than 0.2, the predicted values are also less than 0.2 in most cases. On the other hand, when the actual values are greater than 0.4 the predicted values are more spread out. This suggests, DCG prediction is more precise for *hard* queries than for *average*, and *easy* queries. Our preliminary analysis suggests that feature values for hard queries are most consistent (lower values) compared to easy queries.

Similarly, NDCG prediction is highly precise when predicted values are below 0.3. However, prediction effectiveness degrades quickly when predicted values are above 0.4. Thus, for both measures, the high linear correlation and low RMSE values mask the rather poor effectiveness at the extremes.



(a) DCG (b) NDCG

Figure 2: Prediction versus Actual Metrics for Test fold 1.

Feature Importance. Next, we inspect the features used for regression. We consider three subsets: features based on 1) *LambdaRank* scores, 2) *Click*-based features, and 3) *BM25F*-based features. Table 3 shows the prediction effectiveness of the different feature groups for linear regression. For DCG, all feature groups achieve high correlation while for NDCG, click and BM25F features are substantially lower compared to the combined features. Also, relative feature importance differs for DCG and NDCG. For instance, click features are more important for predicting DCG than LambdaRank score features. The order is reversed for NDCG. Click-based features are strong predictors of user preference [1], and it is no surprise that they correlate well with DCG. However, NDCG being a non-linear metric, is harder to predict with click-based features alone. Also, we hypothesize that since LambdaRank combines several features including click features and is trained to optimize for NDCG, the LambdaRank-based features are better predictors than click-based features. Interestingly, we find that the click features for DCG and LambdaRank features for NDCG are as effective as all the features combined. This suggests that more careful feature selection can reduce run-time computations while retaining prediction effectiveness.

Table 3: Feature Groups Effectiveness: Corr. – Linear correlation measure. RMSE – root mean squared error.

Group	DCG		NDCG	
	Corr.	RMSE	Corr.	RMSE
LambdaRank	0.75	0.14	0.50	0.22
Click	0.78	0.13	0.41	0.24
BM25F	0.71	0.14	0.38	0.24
All	0.78	0.13	0.50	0.23

3. CONCLUSIONS

In this paper, we describe RAPP, an effective and efficient Web query performance prediction technique that uses retrieval scores and retrieval features. Large scale evaluation using actual web queries shows that RAPP is effective, and outperforms the state-of-the-art Clarity baseline. Moreover, experimental results suggest that Clarity is not well-suited for the web. While RAPP is a general approach that can be used for different ranking algorithms and to target different measures, the results in this paper are based only on DCG@5 and NDCG@5 prediction for LambdaRank. We leave investigation of RAPP's utility for other ranking algorithms and performance measures such as MAP as part of future work.

4. ACKNOWLEDGMENTS

This work was supported in part by the Center for Intelligent Information Retrieval and in part by NSF IIS-0910884. Any opinions, findings and conclusions or recommendations expressed here are the authors' and do not necessarily reflect those of the sponsor.

5. REFERENCES

[1] E. Agichtein, E. Brill, and S. Dumais. Improving web search ranking by incorporating user behavior information. In *SIGIR 2006*, 19-26.

[2] C. Burges, R. Ragno, and Q. Le. Learning to rank with nonsmooth cost functions. *Advances in NIPS*, 19:193, 2007.

[3] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR 2002*, pages 299-306.

[4] J. Grivolla, P. Jourlin, and R. de Mori. Automatic classification of queries by expected retrieval performance. In *SIGIR 2005 Workshop on Predicting Query Difficulty*.

[5] E. Yom-Tov, S. Fine, D. Carmel, and A. Darlow. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *SIGIR 2005*, pages 512–519.

[6] A. Liaw and M. Wiener. Classification and regression by randomforest. *R News*, 2(3):18–22, 2002.