Tasks, Queries, and Rankers in Pre-Retrieval Performance Prediction

Paul Thomas
Microsoft
Canberra, Australia

Falk Scholer
RMIT University
Melbourne, Australia

Peter Bailey
Microsoft
Canberra, Australia

Alistair Moffat
The University of Melbourne
Melbourne, Australia

ABSTRACT

Query performance prediction estimates the effectiveness of a query in advance of human judgements. Accurate prediction could be used, for example, to trigger special processing, select query variants, or choose whether to search at all.

Prediction evaluations have not distinguished effects due to *query wording* from effects due to the *underlying information need*, nor from effects due to *performance of the retrieval system* itself. Here we use five rankers, 100 tasks, and 28,869 queries to distinguish these three effects over six pre-retrieval predictors. We see that task effects dominate those due to query or ranker; that many "query performance predictors" are in fact predicting task difficulty; and that this makes it difficult to use these algorithms.

1 QUERY PERFORMANCE PREDICTION

Query performance prediction seeks to ascertain whether the results of a query would be useful, without presenting those results to a user [7, 9, 14]. For example, given the text of a query we may predict the quality of the resulting SERP, were that query to be run; or, given a set of results, we may predict an effectiveness score such as AP without ever showing the results to users or judges.

An accurate predictor would help in at least three settings. First, it could inform *triggering* of special actions—for example, if a query seems to be ineffective then we may choose to spend more effort processing that query, by considering variant terms or looking deeper in the index. An ineffective query might also trigger a different use of screen space, or a different type of response, for example by adding hints. Triggering requires accurate prediction of *absolute* performance: that is, we need to know how well (or poorly) a single query is performing on whatever metric we choose.

Second, it could inform *selection* of queries. Given many possible query variants—constructed for example by dropping, adding, or rewriting terms, or adding or removing location or time constraints—it may not be possible to run them all. Selection needs a predictor of *relative* query effectiveness.

Third, it could inform *optional search* or *search as fallback*. For example, software such as Siri or Cortana might choose to hand off to a search engine; or, having received results back, might choose whether to present them. An accurate predictor could decide when to forward a query, or having run a query, whether to use the results. Again, *absolute* performance estimates are needed here.

Past work has focussed on evaluating predictors with simple measures of correlation, or has examined the effect of query phrasing. In this work we use the UQV100 collection and its multiple queries per task [2], finding that these effects are secondary to task: that is, predictors are responding more to changes in task than to changes in query, and as a consequence their performance estimates are not especially useful.

2 RELATED WORK

A range of performance predictors have been proposed, in two broad categories: those that can produce estimates before running the query ("pre-retrieval") and those that rely on the results of a retrieval and/or ranking pass ("post-retrieval"). Carmel and Yom-Tov [7] and Hauff [14] provide comprehensive surveys.

Pre-retrieval predictors. Pre-retrieval predictors use the text of the query, as well as term statistics or other static resources, to estimate effectiveness. Importantly, they do not rely on any retrieval results and therefore can be run before ranking and retrieval, or can be run before committing to ranking at all.

Predictors make use of collection-independent features such as length [15]; morphology and syntax [18]; or the similarity of query terms, for example with relation to an external thesaurus [14, 18]. They may also draw on term occurrence data, for example by estimating the specificity of query terms [15, 20, 24], their similarity to the collection [15], or their co-occurrence in documents [14].

Post-retrieval predictors. Much work has used *clarity*, the difference between a language model of the top-ranked documents and that of the whole collection. A high-performing query will give top-ranked documents which are both coherent and different to the collection overall [10]. By drawing on the documents actually retrieved, we might expect more reliable predictions; but this requires a full retrieval run, so is relatively expensive.

Several further methods rely on perturbation, assuming that results which are robust are likely to be high quality. For example, we may perturb queries by adding or removing terms and running

ADCS 2017, December 7–8, 2017, Brisbane, QLD, Australia

ACM ISBN 978-1-4503-6391-4/17/12…$15.00
https://doi.org/10.1145/3166072.3166079

the modified versions [11, 25]; perturb documents by making small changes to the text, then re-ranking [21, 25]; or perturb rankers, by running different systems and comparing the resulting lists [1]. These perturbation-based methods are of course even more expensive, as they need a source of queries; or to re-run the ranker; or need two or more state-of-the-art rankers, to run in parallel.

Other models have made use of further post-retrieval features, including past interactions with retrieved documents [4].

Evaluations. An experiment attached to the TREC-6 conference asked nine NIST staff to predict whether topics would be "hard", "middling", or "easy", and these predictions were compared to actual AP scores [23]. Predictions from the most accurate individual only managed a Pearson correlation of 0.26 with the actual results; the best agreement between assessors was only 0.39. Even for human experts, and even at a coarse grain, prediction is clearly hard.

Participants in the TREC Robust track, 2004–5, were asked to automatically rank topics by predicted difficulty. There was a "strong positive correlation" between a custom measure of prediction success and overall system performance; rank correlation between predicted and actual performance ranged up to $\tau = 0.62$ [22].

The evaluation by Hauff [14] is the most thorough to date. It calculated the linear correlation between predictor and effectiveness measures, an approach that we also employ here. Using 400 TREC queries from three corpora, and three retrieval methods, Hauff reports the best pre-retrieval performance from MaxSCQ and MaxIDF, and the best post-retrieval outcomes from modified clarity-based predictors, with performance varying with ranker and collection.

Scholer and Garcia [19] have also considered the performance of predictors, measuring correlations as different rankers are used. Using 234 runs from two TREC tracks, they see substantial variation in τ, and significant noise in evaluations. If two predictors were compared on τ, a change in ranker would lead to a change in conclusions 14–41% of the time. This is consistent with Hauff's observations on ranker dependence.

Task and query variation. With only one query per topic, these evaluations conflate query and task (as well as corpus) effects. Since query variation leads to effectiveness variation [6, 17], it is possible for a hard topic to mask an effective query, and vice versa. A result list may be poor because the task is hard; because there isn't coverage in the corpus; or because the query itself is ineffective.

Carmel et al. [8] describe models which predict effectiveness based on query phrasing as well on as the distribution of relevant documents and the underlying corpus. The best predictions use the difference between relevant documents and the background corpus—that is, a task and corpus effect—although in most cases the relevant documents are not known ahead of time. These experiments again use TREC data, with one query per topic, so still cannot clearly distinguish between query and task effects.

3 EXPERIMENTS

Triggering could make use of pre- or post-retrieval methods; but if the follow-on process is expensive, then the cheaper pre-retrieval methods are preferable. Query selection, which runs before any queries are processed, clearly needs an accurate pre-retrieval prediction; and search as fallback or optional search could make use of either pre- or post-retrieval techniques. We therefore focus on the relatively cheap and generally applicable pre-retrieval methods.

Predictors. We have tested six pre-retrieval predictors which have performed well in past evaluations, or are otherwise interesting, and which are inexpensive to compute: query terms, AvgQL, AvgP, MaxIDF, MaxSCQ, and MaxVAR.

The number of terms in the query is a simple baseline [15]. TREC topics have consistently short titles, which has made this hard to test; but the queries in UQV100 are both relatively long (mean 5.4 terms) and variable (range 1–26, s.d. 2.5 terms). Mothe and Tanguy [18] suggest another baseline, the mean number of characters per query term (Hauff's "AvgQL"). Longer terms may suggest more technical or specific vocabulary, which in turn should suggest better discrimination and better retrieval. Mothe and Tanguy also use polysemy as a measure of term specificity; AvgP is the mean number of senses for each query term, as measured by the number of WordNet synsets in which it appears. Following Mothe and Tanguy [18], we consider only single terms; Hauff's variation matches multi-term phrases but performed poorly [14].

Other approaches use corpus statistics to predict performance. Scholer et al.'s MaxIDF measure again estimates specificity, using a conventional IR measure [20]: $\text{MaxIDF} = \max_{t \in Q} (1/\text{df}_t)$, where Q is the set of terms in the query and df_t is the document frequency of t (the number of documents in which it occurs).

Zhao et al. [24] extend this idea to use both tf and idf, and again use the contribution of the highest-weighted term:

$$\text{MaxSCQ} = \max_{t \in Q} \left((1 + \ln(\text{cf}_t)) \times \ln\left(1 + \frac{N}{\text{df}_t}\right)\right),$$

where N is the number of documents in the collection and cf_t is the collection frequency of term t (occurrences across all documents).

Finally, the MaxVAR predictor considers the variance of term weights in responsive documents [24]. The intuition is again that widely-varying weights—high variance—should help the ranker differentiate between relevant and non-relevant documents:

$$\text{MaxVAR} = \max_{t \in Q} \left(\sqrt{\frac{1}{\text{df}_t} \sum_{d \in D_t} \left(w_{d,t} - \overline{w}_t\right)^2} \right), \tag{1}$$

where D_t denotes the set of documents containing term t, and where $w_{d,t}$ is the weight of term t in document d in a tf · idf model,

$$w_{d,t} = \begin{cases} 1 + \ln(\text{tf}_{d,t}) \ln(N/\text{df}_t) & \text{if } t \text{ in vocabulary} \\ 0 & \text{otherwise,} \end{cases}$$

in which $\text{tf}_{d,t}$ is the term frequency of t in d. Finally in Equation 1, $\overline{w}_t$ is the mean weight of term t in all documents in which it occurs, $\overline{w}_t = \sum_{d \in D_t} w_{d,t} / |D_t|$.

Corpus, rankers, and implementation. The UQV100 collection has 100 tasks, with on average 58 query variations for each [2]. We obtained runs from five rankers, covering a range of algorithmic choices: Indri using a BM25 and a language model method; Atire, using a quantised-impact method; and Terrier [16], using a PL2 and a DFRFree method.[1] In total, there are 28,869 distinct system:task:query combinations, making it possible to investigate each of the three factors separately. The UQV100 resource also includes

[1] http://www.lemurproject.org/indri/; http://atire.org; http://terrier.org

Predictor	AP per query r	AP per query τ	Median AP per task r	Median AP per task τ
Query terms	-0.04^*	0.00	-0.05^*	-0.04^*
AvgQL	0.04^*	0.04^*	0.01	0.02
AvgP	-0.17^*	-0.14^*	-0.20^*	-0.18^*
MaxIDF	0.04^*	0.31^*	0.07^*	0.33^*
MaxSCQ	0.42^*	0.32^*	0.46^*	0.35^*
MaxVAR	0.27^*	0.32^*	0.31^*	0.35^*

Table 1: Performance of selected performance predictors on the UQV100 data. Each is run separately, predicting AP. All r values ± 0.01. * marks correlations significantly different to zero ($p < 0.05$).

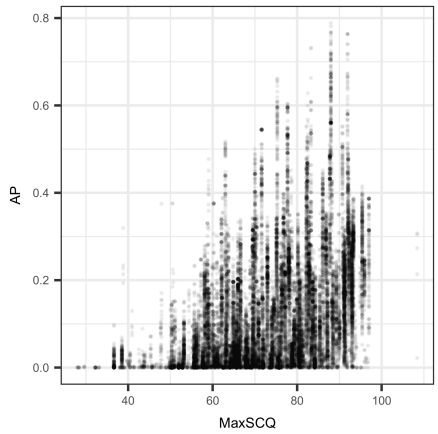

Figure 1: Relationship between MaxSCQ and AP scores, for the 28,869 query:task:ranker triples in UQV100 ($r = 0.42$).

partial relevance judgements, initially built from pools of depth 10, and then extended to reduce residuals (uncertainty) in the INST metric. There are 55,587 judgements on a five-point scale, with three-way overlap. We use the median label and compute average precision (AP) as our effectiveness metric, in line with past work.

Term statistics were based on processing with the Krovetz stemmer, and no stopwords;[2] with AvgP computed by the R `wordnet` interface [12]. Terms not in WordNet were treated as monosymous.

4 RESULTS AND DISCUSSION

Predicting absolute effectiveness. The left-hand side of Table 1 gives correlations (Pearson's r and Kendall's τ) between each of the predictors above and per-query AP scores, using the UQV100 data. These results are broadly in line with Hauff [14], although Hauff reported better correlations with MaxIDF (r between 0.21 and 0.65, vs 0.04 here). We also note that the data in Table 1 is over 28,869 pairs of predictor and metric—one for each combination of query, task, and ranker—whereas past work has tended to use TREC data with only a single query for each topic. Note that some pairs will correlate, since we have 100 tasks and five rankers underlying the 28,869 pairs. We consider tasks and rankers separately shortly.

A correlation of 0.42, for MaxSCQ, might be considered moderately useful. In practice however there is a good deal of noise and it would be brave to rely on this in a live system (Figure 1).

[2]Hauff [14] saw little difference when varying stemmers or stoppers.

Factor	Partial η^2	Coefficient	F	p
Query terms	0	−0.000 04	0	n.s.
AvgQL	0.001	0.003	29	$< .05$
AvgP	0	−0.000 01	0	n.s.
MaxIDF	0.001	−9.97	34	$< .05$
MaxSCQ	0.02	0.002	515	$< .05$
MaxVAR	0.0008	0.15	22	$< .05$
Task	0.61	−0.12–0.43	455	$< .05$
Ranker	0.01	0–0.03	99	$< .05$

Table 2: Modelling query performance on the UQV100 data. Topic effects dominate (partial $\eta^2 = 0.61$). F values on 1 d.f., except task (99 d.f.) and ranker (4 d.f.).

Controlling for task and ranker. The query and ranker variation in UQV100 let us control for task and ranker by modelling them as separate variables. Table 2 describes a linear model predicting the AP score for each query:task:ranker combination. We report the coefficient (effect) for each predictor: for example, each point of change in MaxSCQ increases the modelled AP score by 0.002. We also report the effect of changing the ranker (gains up to 0.03 points of AP) and task (ranges from losing 0.12 to gaining 0.43 points).

All effects save those of query terms and AvgP are significant at $p < 0.05$, but the effect size (partial η^2, proportion of the variance explained) varies considerably. Most of the variance in AP is explained by changes in task ($\eta^2 = 0.61$) and after allowing for this—which is to say, after allowing for task difficulty—there is very little explained by any of the putative predictors. The choice of ranker explains more of the AP score than any predictor besides MaxSCQ. This suggests that "query performance" predictors may not be predicting query effectiveness so much as task difficulty. This conclusion is supported by correlations between each predictor and task difficulty (for which we use median AP as a proxy; right-hand side of Table 1). All but one of the predictors correlate better with task difficulty than they do with per-query effectiveness.

We can also consider the correlations within a single task:ranker pair, that is, correlations between a predictor and AP when we hold the task and ranker constant and the only variation is due to query phrasing. When we control for task effects this way, the Pearson's r values are much lower: MaxSCQ, the best of the predictors, has a mean $r = 0.17$ (median 0.15). Again this is an effect of not being able to effectively predict per-query difficulty. If we consider all 28,869 runs, then we can achieve high r just by getting the tasks more or less in order; but within any single task, it is much harder to predict the effectiveness of a single query.

That is, our results make it clear that task difficulty is a major confound in query performance prediction, an observation that has been somewhat hidden to date due to the single-query TREC topics used in past evaluation. Whether this confound is an issue in practice depends on the application. If we are using prediction to trigger more processing, the confound is a problem: a poor query might benefit, but the extra effort would be wasted if the task is inherently difficult. If we are using prediction to trigger advice or hints for the searcher, we want to provide different advice for poor queries or hard tasks, so the confound is also important. For selection, the task is fixed and performance is relative, so the confound is not a problem, and we will consider this case next.

Predictor	Accuracy on all pairs	Accuracy on big diffs
Query terms	0.52	0.58
AvgQL	0.52	0.53
AvgP	0.51	0.48
MaxIDF	0.53	0.58
MaxSCQ	0.54	0.61
MaxVAR	0.54	0.63

Table 3: Performance for each predictor, on the selection task.

Selecting queries. Recall that the selection problem is to predict which of a pair of query formulations is more effective, given a fixed task and ranker. To examine this we considered all pairs of query variants for each task:ranker combination; this gave 916,371 variant pairs. We then counted the number of times each predictor correctly identified the query that gave the higher AP score.[3]

Table 3 makes it clear that none of the predictors do at all well on this task. Even the best performers can only identify the higher-performing of two queries, given a task and ranker, 54% of the time. It may be immaterial which way the prediction goes if there is no practical difference in query performance, so we also considered only those query pairs where AP scores differed by at least 0.2. On this "big difference" subset, most predictors improved, but the best performer (MaxVAR) was still correct only 63% of the time. Clearly, these pre-retrieval techniques are of only limited practical use.

Other effectiveness metrics. The results reported above are based on AP as the target metric. While not shown, similar patterns also arise with alternative metrics, both shallow (INST and RBP0.85) and deep (NDCG). In general, the overall correlations (Table 1) are higher with deeper metrics, but the task confounds continue to dominate and selection is barely better than random in any case.

5 CONCLUSIONS

Accurate pre-retrieval predictions of query effectiveness would be useful for triggering, selecting, and choosing when to search at all. Past evaluations have used multiple tasks and rankers, but a single query per task. This has conflated tasks with queries.

Scholer and Garcia [19] have used changes in rankers to argue that evaluations of query performance prediction are missing a significant source of variability. In a similar vein, our results show that properly accounting for another source of variability—task difficulty—substantially changes the picture.

Using the query variations, runs, and relevance judgements from the UQV100 collection we examined the effect of task difficulty. We conclude that—as seen in other contexts [5, 13]—task is a serious confound; further, "query performance" predictors in fact do marginally better at predicting task difficulty than anything query-related. Moreover, the moderate overall correlations sometimes seen are deceptive, and if we look inside a single task:ranker combination then correlations are poor. That is, current techniques for pre-retrieval prediction are not likely useful for triggering special processing, or extra help to searchers. Performance on the selection task—choosing a query variant, given a fixed task and ranker—is even worse, with performance barely above random.

It is possible that post-retrieval methods might perform better in this regard. However these predictors rely on running the full ranking stack, and in many cases also process the text of each retrieved document, which makes them irrelevant for the selection problem and expensive for triggering, or for optional search. It may also be possible to develop alternative pre-retrieval methods, drawing on different evidence (for example, past behaviour) or on different analyses of the query text. While such approaches might yet be developed, our primary observation here is that present methods appear inadequate for practical applications.

Acknowledgements. This work was supported by the Australian Research Council (project DP140102655). Matt Crane, Xiaolu Lu, David Maxwell, and Andrew Trotman provided system runs.

[3] This is similar to the setup used by Balasubramanian and Allan [3]; however while they choose a ranker given a single query, we assume the ranker is fixed and try to choose a query to run. This is the problem we face in the "selection" and "search as fallback" scenarios.

REFERENCES

[1] J. A. Aslam and V. Pavlu. 2007. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *Proc. ECIR*. 198–209.
[2] P. Bailey, A. Moffat, F. Scholer, and P. Thomas. 2016. UQV100: A test collection with query variability. In *Proc. SIGIR*. 725–728. Public data: http://dx.doi.org/10.4225/49/5726E597B8376.
[3] N. Balasubramanian and J. Allan. 2010. Learning to select rankers. In *Proc. SIGIR*. 855–856.
[4] N. Balasubramanian, G. Kumaran, and V. R. Carvalho. 2010. Predicting query performance on the web. In *Proc. SIGIR*. 785–786.
[5] D. Banks, P. Over, and N.-F. Zhang. 1999. Blind men and elephants: Six approaches to TREC data. *Inf. Retr.* 1, 1-2 (1999), 7–34.
[6] C. Buckley and J. Walz. 1999. The TREC-8 query track. In *Proc. TREC*.
[7] D. Carmel and E. Yom-Tov. 2010. *Estimating the query difficulty for information retrieval.* Number 15 in Synthesis Lectures on Information Concepts, Retrieval, and Services. Morgan & Claypool.
[8] D. Carmel, E. Yom-Tov, A. Darlow, and D. Pelleg. 2006. What makes a query difficult?. In *Proc. CIKM*. 390–397.
[9] D. Carmel, E. Yom-Tov, and I. Soboroff. 2005. Predicting query difficulty: Methods and applications. *SIGIR Forum* 39, 2 (2005), 25–28.
[10] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. 2002. Predicting query performance. In *Proc. SIGIR*. 299–306.
[11] F. Diaz. 2007. Performance prediction using spatial autocorrelation. In *Proc. SIGIR*. 583–590.
[12] I. Feinerer and K. Hornik. 2016. wordnet: WordNet Interface. https://CRAN.R-project.org/package=wordnet. (2016). R package version 0.1-11.
[13] N. Ferro and G. Silvello. 2016. A general linear mixed models approach to study system component effects. In *Proc. SIGIR*. 25–34.
[14] C. Hauff. 2010. *Predicting the effectiveness of queries and retrieval systems.* Ph.D. Dissertation. University of Twente.
[15] B. He and I. Ounis. 2004. Inferring query performance using pre-retrieval predictors. In *Proc. SPIRE*. 43–54.
[16] C. Macdonald, R. McCreadie, R. L. T. Santos, and I. Ounis. 2012. From puppy to maturity: Experiences in developing Terrier. In *Proc. SIGIR Wrkshp. Open Source IR*. 60–63.
[17] A. Moffat, P. Bailey, F. Scholer, and P. Thomas. 2017. Incorporating user expectations and behavior into the measurement of search effectiveness. *ACM Trans. Inf. Sys.* 35, 3 (2017), 24:1–24:38.
[18] J. Mothe and L. Tanguy. 2005. Linguistic features to predict query difficulty: A case study on previous TREC campaigns. In *Proc. SIGIR Wrkshp. Predicting Query Difficulty: Methods and Applications*. 7–10.
[19] F. Scholer and S. Garcia. 2009. A case for improved evaluation of query difficulty prediction. In *Proc. SIGIR*. 640–641.
[20] F. Scholer, H. E. Williams, and A. Turpin. 2004. Query association surrogates for web search. *JASIST* 55, 7 (2004), 637–650.
[21] V. Vinay, I. J. Cox, N. Milic-Frayling, and K. Wood. 2007. On ranking the effectiveness of searches. In *Proc. SIGIR*. 398–404.
[22] E. M. Voorhees. 2004. Overview of the TREC 2004 robust retrieval track. In *Proc. TREC*.
[23] E. M. Voorhees and D. Harman. 1997. Overview of the sixth Text REtrieval Conference. In *Proc. TREC*.
[24] Y. Zhao, F. Scholer, and J. Tsegay. 2008. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proc. ECIR*. 52–64.
[25] Y. Zhou and W. B. Croft. 2007. Query performance prediction in web search environments. In *Proc. SIGIR*. 543–550.