Query Hardness Estimation Using Jensen-Shannon Divergence Among Multiple Scoring Functions

Javed A. Aslam* and Virgil Pavlu
{jaa,vip}@ccs.neu.edu

Northeastern University

Abstract. We consider the issue of query performance, and we propose a novel method for automatically predicting the difficulty of a query. Unlike a number of existing techniques which are based on examining the ranked lists returned in response to perturbed versions of the query with respect to the given collection or perturbed versions of the collection with respect to the given query, our technique is based on examining the ranked lists returned by multiple scoring functions (retrieval engines) with respect to the given query and collection. In essence, we propose that the results returned by multiple retrieval engines will be relatively similar for "easy" queries but more diverse for "difficult" queries. By appropriately employing Jensen-Shannon divergence to measure the "diversity" of the returned results, we demonstrate a methodology for predicting query difficulty whose performance exceeds existing state-of-the-art techniques on TREC collections, often remarkably so.

1 Introduction

The problem of *query hardness estimation* is to accurately and automatically predict the difficulty of a query, i.e., the likely quality of a ranked list of documents returned in response to that query by a retrieval engine, and to perform such predictions in the absence of relevance judgments and without user feedback. Much recent research has been devoted to the problem of query hardness estimation, and its importance has been recognized by the IR community [1–7]. An accurate procedure for estimating query hardness could potentially be used in many ways, including the following:

- Users, alerted to the likelihood of poor results, could be prompted to reformulate their query.
- Systems, alerted to the difficult query, could automatically employ enhanced or alternate search strategies tailored to such difficult queries.
- Distributed retrieval systems could more accurately combine their input results if alerted to the difficulty of the query for each underlying (system, collection) pair [2].

* We gratefully acknowledge the support provided by NSF grants CCF-0418390 and IIS-0534482.

In this work, we propose a new method for automatically predicting the difficulty of a given query. Our method is based on the premise that different retrieval engines or scoring functions will retrieve relatively similar ranked lists in response to "easy" queries but more diverse ranked lists in response to "hard" queries. As such, one could automatically predict the difficulty of a given query by simultaneously submitting the query to multiple retrieval engines and appropriately measuring the "diversity" of the ranked list responses obtained. In order to measure the diversity of set of ranked lists of documents, we map these rankings to distributions over the document collection, where documents ranked nearer the top of a returned list are naturally associated with higher distribution weights (befitting their importance in the list) and vice versa. Given a set of distributions thus obtained, we employ the well known Jensen-Shannon divergence [8] to measure the diversity of the distributions corresponding to these ranked lists.

We extensively tested our methodology using the benchmark TREC collections [9–15]. To simulate the ranked lists returned by multiple retrieval strategies in response to a given (TREC) query, we chose subsets of the retrieval runs submitted in response to that query in a given TREC. We then predicted query difficulty using the methodology described and compared our estimated query difficulty to the difficulty of that query in that TREC, as measured in a number of standard and new ways. Finally, we compared the quality of our query difficulty estimates to state-of-the-art techniques [1, 6, 7], demonstrating significant, often remarkable, improvements.

The remainder of this paper is organized as follows. We begin by more extensively discussing relevant related work, followed by a presentation of our methodology in detail. We then discuss the results of extensive experiments using the TREC collections. Finally, we conclude with a summary and discussion of future work.

2 Background and Related Work

Existing work on query hardness estimation can be categorized along at least three axes: (1) How is query hardness defined?, (2) How is query hardness predicted?, and (3) How is the quality of the prediction evaluated? In what follows, we describe our work and related work along these dimensions.

2.1 Defining query hardness

One can define query hardness in many ways; for example, queries can be inherently difficult (e.g., ambiguous queries), difficult for a particular collection, or difficult for a particular retrieval engine run over a particular collection. Other notions of query difficulty exist as well. In what follows, we discuss two notions of query harness, which we shall refer to as *system query hardness* and *collection query hardness*.

System query hardness captures the difficulty of a query for a given retrieval system run over a given collection. Here the notion of query hardness is system-specific; it is meant to capture the difficulty of the query for a *particular* system, run over a given collection. System query hardness is typically measured by the average precision of the ranked list of documents returned by the retrieval system when run over the collection using the query in question.

Examples of work considering system query hardness include (1) Carmel et al. [3] and Yom-Tov et al. [1] who investigate methods for predicting query hardness, testing against the Juru retrieval system, (2) Cronen-Townsend et al. [6] and Zhou and Croft [7] who investigate methods for predicting query hardness, testing against various language modeling systems, and (3) the Robust track at TREC [13] wherein each system attempted to predict its own performance on each given query.

Collection query hardness captures the difficulty of a query with respect to a given collection. Here the notion of query hardness is meant to be largely *independent* of any specific retrieval system, capturing the inherent difficulty of the query (for the collection) and perhaps applicable to a wide variety of typical systems. Collection query hardness can be measured by some statistic taken over the performance of a wide variety of retrieval systems run over the given collection using the query in question. For example, Carmel et al. [3] consider collection query hardness by comparing the query difficulty predicted by their method to the median average precision taken over all runs submitted in the Terabtye tracks at TREC for a given query.

Our work: We consider both system query hardness and collection query hardness and demonstrate that our proposed methodology is useful in predicting either. In order to test the quality of our methodology for predicting a given system's performance (system query harness), one must fix a retrieval system. In this work, we simply choose the system (retrieval run) whose mean average precision was the median among all those submitted to a particular TREC; thus, we consider a "typical" system, one whose performance was neither extremely high or low. We refer to this measure as the *median system AP* (med-sys AP).

In order to test the quality of our methodology for predicting collection query hardness, one must fix a measure for assessing the hardness of a query for a given collection. In this work, we consider two statistics taken over all runs submitted to a particular TREC with respect to a given query: (1) the *average* of the average precisions for all runs submitted in response to a given query and (2) the *median* of the average precisions for all runs submitted in response to a given query. We refer to the former as *query average AP* (avgAP) and the latter as *query median AP* (medAP).

2.2 Predicting query hardness

Cronen-Townsend et al. [6] introduced the *clarity score* which effectively measures the ambiguity of the query with respect to a collection, and they show that clarity scores are correlated with query difficulty. Clarity scores are computed by assessing the information-theoretic distance between a language model associated with the query and a language model associated with the collection. Subsequently, Zhou and Croft [7] introduced *ranking robustness* as a measure of query hardness, where ranking robustness effectively measures the stability in ranked results with respect to perturbations in the collection.

Carmel et al. [3] proposed the use of pairwise information-theoretic distances between distributions associated with the collection, the set of relevant documents, and the query as predictors for query hardness. Yom-Tov et al. [1] proposed a method for predicting query hardness by assessing the stability of ranked results with respect to perturbations in the query; subsequently, they showed how to apply these results to the problem of metasearch [2].

In other related work, Amati et al. [4] studied query hardness and robustness in the context of query expansion, Kwok [5] proposed a strategy for selecting the scoring function based on certain properties of a query, and Macdonald et al. [16] investigated query hardness prediction in an intranet environment.

Our work: While fundamentally different from existing techniques, our work is related to the methodologies described above in a number of ways. A number of existing techniques predict query hardness by measuring the stability of ranked results in the presence of perturbations of the *query* [1] or perturbations of the *collection* [7]. In a similar spirit, our proposed technique is based on measuring the stability of ranked results in the presence of perturbations of the *scoring function*, i.e., the retrieval engine itself. We measure the "stability" of the ranked results by mapping each ranked list of documents returned by a different scoring function to a probability distribution and then measuring the diversity among these distributions using the information-theoretic Jensen-Shannon divergence [8]. In a similar spirit, Cronen-Townsend et al. [6] use the related Kullback-Leibler divergence [17] to compute clarity scores, and Carmel et al. [3] use the Jensen-Shannon divergence to compute their query hardness predictor.

2.3 Evaluating the quality of query hardness predictions

In order to evaluate the quality of a query hardness prediction methodology, test collections such as the TREC collections are typically used. The system and/or collection hardnesses of a set of queries are measured, and they are compared to predicted values of query hardness. These actual and predicted values are real-valued, and they are typically compared using various parametric and non-parametric statistics. Zhou and Croft [7] and Carmel et al. [3] compute the *linear correlation coefficient* ρ between the actual and predicted hardness values; ρ is a parametric statistic which measures how well the actual and predicted hardness values fit to a straight line. If the queries are *ranked* according to the actual and predicted hardness values, then various non-parametric statistics can be computed with respect to these rankings. Cronen-Townsend et al. [6] and Carmel et al. [3] compute the Spearman rank correlation coefficient. Zhou and

Croft [7], Yom-Tov et. al [1], and the TREC Robust track [13] all compute and report the Kendall's τ statistic.

Our work: In the results that follow, we assess the quality of our query hardness predictions using both the linear correlation coefficient ρ and Kendall's τ.

3 Methodology

Our hypothesis is that disparate retrieval engines will return "similar" results with respect to "easy" queries and "dissimilar" results with respect to "hard" queries. As such, for a given query, our methodology essentially consists of three steps: (1) submit the query to multiple scoring functions (retrieval engines), each returning a ranked list of documents, (2) map each ranked list to a distribution over the document collection, where higher weights are naturally associated with top ranked documents and vice versa, and (3) assess the "disparity" (collective distance) among these distributions. We discuss (2) and (3) in the sections that follow, reserving our discussion of (1) for a later section.

3.1 From ranked lists to distributions

Many measures exist for assessing the "distance" between two ranked lists, such as the Kendall's τ and Spearman rank correlation coefficients mentioned earlier. However, these measures do not distinguish between differences in the "top" of the lists from equivalent differences in the "bottom" of the lists; however, in the context of information retrieval, two ranked lists would be considered much more dissimilar if their differences occurred at the "top" rather than the "bottom" of the lists.

To capture this notion, one can focus on the top retrieved documents only. For example, Yom-Tov et al. [1] compute the *overlap* (size of intersection) among the top N documents in each of two lists. Effectively, the overlap statistic places a uniform $1/N$ "importance" to each of the top N documents and a zero importance to all other documents. More natural still, in the context of information retrieval, would be weights which are higher at top ranks and smoothly lower at lesser ranks. Recently, we proposed such weights [18, 19] which correspond to the implicit weights which the average precision measure places on each rank, and we use these distribution weights in our present work as well. Over the top c documents of a list, the distribution weight associated with any rank r, $1 \leq r \leq c$, is given below; all other ranks have distribution weight zero.

$$\text{weight}(r) = \frac{1}{2c}\left(1 + \frac{1}{r} + \frac{1}{r+1} + \cdots + \frac{1}{c}\right). \tag{1}$$

3.2 The Jensen-Shannon divergence among distributions

Using the above distribution weight function, one can map ranked lists to distributions over documents. In order to measure the "disparity" among these lists, we measure the disparity or divergence among the distributions associated with these lists. For two distributions $\boldsymbol{a} = (a_1, \ldots, a_n)$ and $\boldsymbol{b} = (b_1, \ldots, b_n)$, a natural and well studied "distance" between these distributions is the Kullback-Leibler divergence [17]:

$$KL(\boldsymbol{p}||\boldsymbol{q}) = \sum_i p_i \log \frac{p_i}{q_i}$$

However, the KL-divergence suffers two drawbacks: (1) it is not symmetric in its arguments and (2) it does not naturally generalize to measuring the divergence among more than two distributions. We instead employ the related Jensen-Shannon divergence [8]. Given a set of distributions $\{\boldsymbol{p_1}, \ldots, \boldsymbol{p_m}\}$, let $\overline{\boldsymbol{p}}$ be the average (centroid) of these distributions. The Jensen-Shannon divergence among these distributions is then defined as the average of the KL-divergences of each distribution to this average distribution:

$$JS(\boldsymbol{p_1}, \ldots, \boldsymbol{p_m}) = \frac{1}{m} \sum_j KL(\boldsymbol{p_j}||\overline{\boldsymbol{p}})$$

An equivalent and somewhat simpler formulation defined in terms of entropies also exists [8]. In this work, we employ the Jensen-Shannon divergence among the distributions associated with the ranked lists of documents returned by multiple retrieval engines in response to a give query as an estimate of query hardness.

4 Experimental Setup and Results

We tested our methodology extensively on multiple TREC datasets: TREC5, TREC6, TREC7, TREC8, Robust04, Terabyte04, and Terabyte05. The performance of our proposed Jensen-Shannon query hardness estimator is measured against three benchmark query hardness statistics: query average AP (avgAP) and query median AP (medAP), both measures of *collection query hardness*, and median-system AP (med-sys AP), a measure of *system query hardness*. When predicting the difficulties of multiple queries in any given TREC, the strength of correlation of our predicted difficulties with actual query difficulties is measured by both Kendall's τ and linear correlation coefficient ρ. We conclude that even when using few input systems, our method consistently outperforms existing approaches [1, 6, 7], sometimes remarkably so.

The ad hoc tracks in TRECs 5–8 and the Robust track in 2004 each employ a standard 1,000 documents retrieved per system per query on collections of size in the range of hundreds of thousand of documents. For these collections, the weight cutoff was fixed at $c = 20$ in Equation 1; in other words, only the top 20 documents retrieved by each system received a non-zero weight in the distribution corresponding to the retrieved list, as used in the Jensen-Shannon divergence computation. The Terabyte tracks use the GOV2 collection of about 25 million documents, and ranked result lists consist of 10,000 documents each; for this larger collection and these longer lists, the weight cutoff was set at

$c = 100$ in Equation 1. This work leaves open the question of how to optimally set the weight cutoff per system, query, and/or collection.

The baseline statistics query avgAP, query medAP, and the fixed system medsys AP are computed among *all* retrieval runs available. The Jensen-Shannon divergence is computed among 2, 5, 10, 20, or all retrieval runs available. When less than all of the available runs are used, the actual runs selected are chosen at random, and the entire experiment is repeated 10 times; scatter plots show a typical result among these 10 repetitions, and tables report the average performance over all 10 repetitions. We note that in general, the quality of our query hardness predictions increases rapidly as more system runs are used, with improvements tailing off after the inclusion of approximately 10 systems.

We compare our Jensen-Shannon query hardness predictions with all three baseline statistics, for all queries and all collections; in two isolated cases we excluded queries with zero relevant documents. Figures 1 and 2 show a selection of the results as scatter plots, separately for JS estimation using five system runs and for JS estimation using 10 system runs.

Prediction Method	TREC5	TREC6	TREC7	TREC8	Robust04	TB04	TB05
JS (2 systems)	0.334	0.353	0.436	0.443	0.393	0.339	0.288
JS (5 systems)	0.420	0.443	0.468	0.551	0.497	0.426	0.376
JS (10 systems)	0.468	0.444	0.544	0.602	0.502	0.482	0.406
JS (20 systems)	0.465	0.479	0.591	0.613	0.518	0.480	0.423
JS (all systems)	0.469	0.491	0.623	0.615	0.530	0.502	0.440

Table 1. Kendall's τ (JS vs. query average AP) for all collections using 2, 5, 10, 20, and all input systems. All but the last row report 10-run average performance.

Kendall's τ measures the similarity of two rankings, in our case, the rankings of the queries in terms of a baseline measure (query average AP, query median AP, or median-system AP) and the rankings of the queries in terms of our Jensen-Shannon estimate. Prediction performance as measured by Kendall's τ is given in Tables 1, 2, and 3, and for visual purposes, we graph Tables 2 and 3 in Figure 3. Note that while our scatter plots seem to indicate negative correlation (high Jensen-Shannon divergence implies low query performance), this indicates positive correlation with the problem as defined (high Jensen-Shannon divergence implies high query difficulty). As such, we report the corresponding "positive" correlations in all tables, and we note the equivalent negative correlations in all scatter plots.

Where we could make a direct comparison with prior results (TREC5, TREC8, Robust04, Terabyte04, and Terabyte05), we indicate the performance reported in prior work along with references. For past results measuring *system query hardness* (i.e., correlations between predicted query hardness and the hardness of the query for a *specific* system), we compare these prior results against our correlations with the median-system AP, as that would be closest to a fair com-

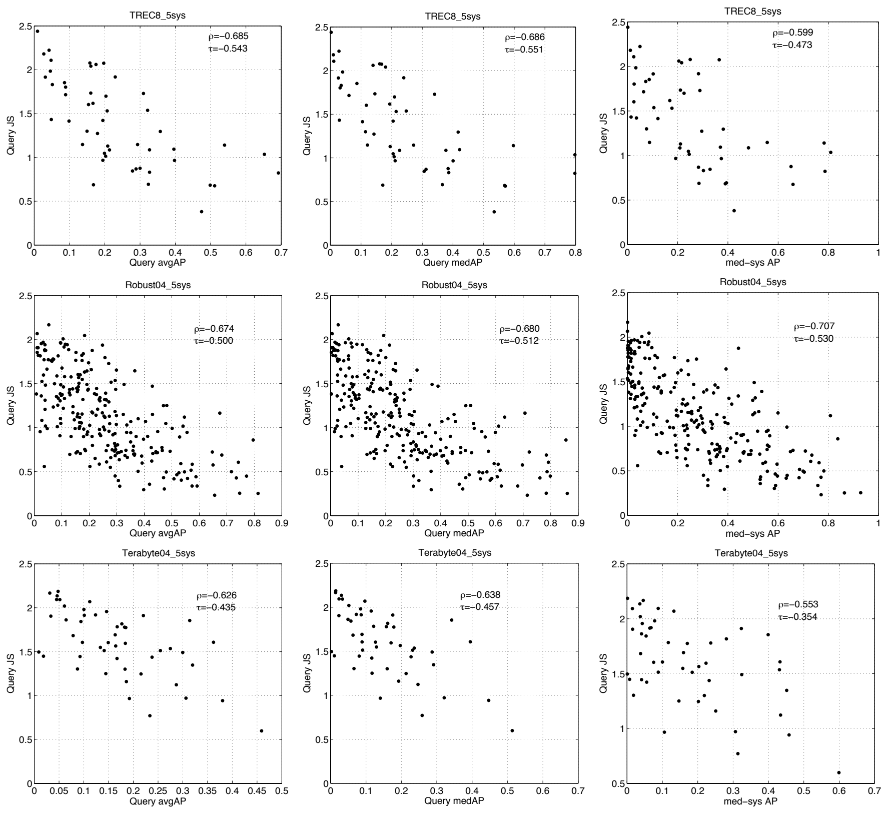

Fig. 1. Query hardness prediction results using five input systems for (top to bottom) TREC8, Robust04 and Terabyte04. Each dot in these scatter plots corresponds to a query. The x-axis is actual query hardness as measured by query average AP (left), query median AP (center), and median-system AP (right). The y-axis is the Jensen-Shannon divergence computed over the ranked results returned by five randomly chosen systems for that query.

Prediction Method	TREC5	TREC6	TREC7	TREC8	Robust04	TB04	TB05
JS (2 systems)	0.341	0.385	0.452	0.442	0.401	0.338	0.260
JS (5 systems)	0.448	0.475	0.483	0.547	0.510	0.435	0.340
JS (10 systems)	0.483	0.464	0.556	0.585	0.515	0.485	0.366
JS (20 systems)	0.488	0.503	0.610	0.599	0.533	0.496	0.382
JS (all systems)	0.510	0.530	0.634	0.597	0.544	0.520	0.391

Table 2. Kendall's τ (JS vs. query median AP) for all collections using 2, 5, 10, 20, and all input systems. All but the last row report 10-run average performance.

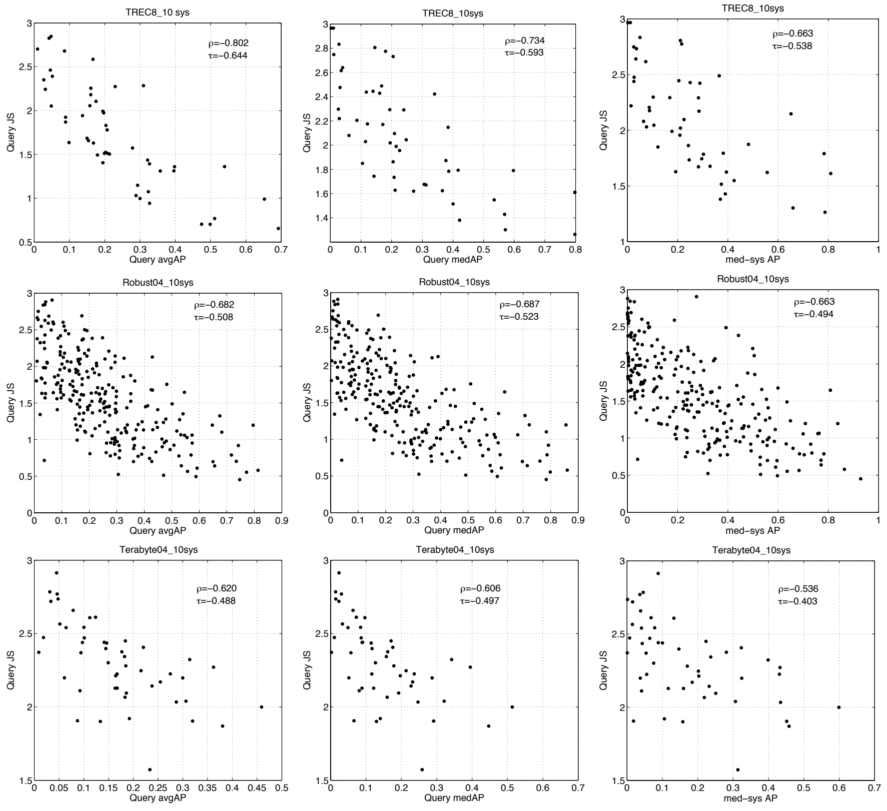

Fig. 2. Query hardness prediction results using 10 input systems for (top to bottom) TREC8, Robust04 and Terabyte04. Each dot in these scatter plots corresponds to a query. The x-axis is actual query hardness as measured by query average AP (left), query median AP (center), and median-system AP (right). The y-axis is the Jensen-Shannon divergence computed over the ranked results returned by 10 randomly chosen systems for that query.

parison. Using 10 input system runs for the Jensen-Shannon computation yield improvements over best previous results of on average approximately 40% to 50%; the complete Tables 3, 5, and 6 show improvements ranging from 7% to 80%.

The linear correlation coefficient ρ effectively measures how well actual and predicted values fit to a straight line; in our case, these actual and predicted values are the hardness of queries in terms of a baseline measure (query average AP, query median AP, or median-system AP) and the hardness of these same queries in terms of our Jensen-Shannon estimate. Prediction performance as measured by linear correlation coefficient is presented in Tables 4, 5, and 6. Note the substantial improvements over prior results as shown in Tables 5 and 6.

Prediction Method	TREC5	TREC6	TREC7	TREC8	Robust04	TB04	TB05
[7] clarity	0.311				0.412	0.134	0.171
[7] robust+clarity	0.345				0.460	0.226	0.252
[1] hist. boost. class.				.439			
JS (2 systems)	0.260	0.276	0.384	0.452	0.387	0.298	0.241
JS (5 systems)	0.350	0.370	0.427	0.525	0.472	0.349	0.318
JS (10 systems)	0.355	0.334	0.476	0.552	0.490	0.408	0.359
JS (20 systems)	0.339	0.365	0.516	0.577	0.498	0.403	0.380
JS (all systems)	0.363	0.355	0.509	0.561	0.512	0.427	0.381

Table 3. Kendall's τ (JS vs. median-system AP) for all collections using 2, 5, 10, 20, and all input systems. All but the last row report 10-run average performance.

Prediction method	TREC5	TREC6	TREC7	TREC8	Robust04	TB04	TB05
JS (2 systems)	0.498	0.456	0.601	0.627	0.555	0.466	0.388
JS (5 systems)	0.586	0.611	0.637	0.736	0.673	0.576	0.516
JS (10 systems)	0.632	0.651	0.698	0.778	0.672	0.642	0.564
JS (20 systems)	0.645	0.677	0.731	0.784	0.688	0.666	0.577
JS (all systems)	0.623	0.698	0.722	0.770	0.695	0.682	0.581

Table 4. Correlation coefficient ρ (JS vs. query average AP) for all collections using 2, 5, 10, 20, and all systems. All but the last row report 10-run average performance.

Prediction method	TREC5	TREC6	TREC7	TREC8	Robust04	TB04	TB05
[3] Juru(TB04+05)						.476	
JS (2 systems)	0.495	0.467	0.612	0.622	0.557	0.466	0.366
JS (5 systems)	0.592	0.631	0.654	0.727	0.677	0.586	0.477
JS (10 systems)	0.622	0.678	0.715	0.762	0.676	0.646	0.524
JS (20 systems)	0.630	0.703	0.752	0.769	0.694	0.674	0.541
JS (all systems)	0.600	0.727	0.743	0.755	0.701	0.687	0.543

Table 5. Correlation coefficient ρ (JS vs. query median AP) for all collections using 2, 5, 10, 20, and all systems. All but the last row report 10-run average performance.

Prediction method	TREC5	TREC6	TREC7	TREC8	Robust04	TB04	TB05
[7] clarity	0.366				0.507	0.305	0.206
[7] robust+clarity	0.469				0.613	0.374	0.362
JS (2 systems)	0.425	0.294	0.553	0.595	0.542	0.435	0.338
JS (5 systems)	0.537	0.459	0.609	0.676	0.645	0.490	0.467
JS (10 systems)	0.556	0.469	0.639	0.707	0.659	0.566	0.524
JS (20 systems)	0.562	0.479	0.679	0.724	0.665	0.585	0.545
JS (all systems)	0.567	0.497	0.657	0.702	0.677	0.603	0.541

Table 6. Correlation coefficient ρ (JS vs. median-system AP) for all collections using 2, 5, 10, 20, and all systems. All but the last row report 10-run average performance.

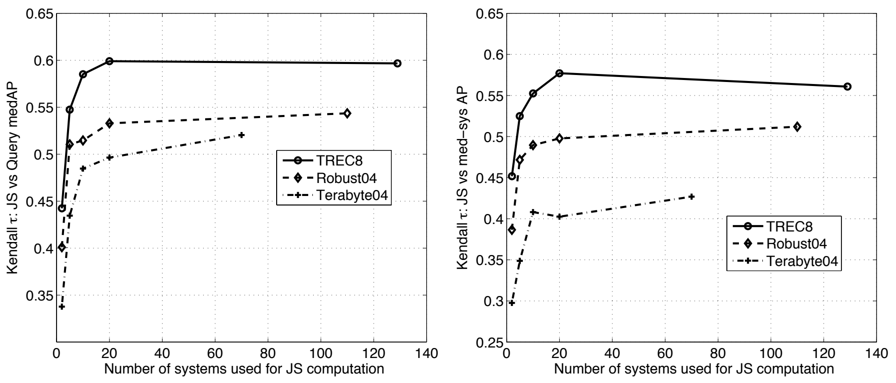

Fig. 3. Kendall's τ for JS vs. query median AP (left) and Kendall's τ for JS vs. median-system AP (right).

5 Conclusion and Future Work

Previous work on query hardness has demonstrated that a measure of the stability of ranked results returned in response to perturbed versions of the query with respect to the given collection or perturbed versions of the collection with respect to the given query are both correlated with query difficulty, both in general and for specific systems. In this work, we further demonstrate that a measure of the stability of ranked results returned in response to perturbed versions of the *scoring function* is also correlated with query hardness, often at a level significantly exceeding that of prior techniques. Zhou and Croft [7] and Carmel et al. [3] demonstrate that *combining* multiple methods for predicting query difficulty yields improvements in the predicted results, and we hypothesize that appropriately combining our proposed method with other query difficulty prediction methods would yield further improvements as well. Finally, this work leaves open the question of how to optimally pick the number and type of scoring functions (retrieval engines) to run in order to most efficiently and effectively predict query hardness.

References

1. Yom-Tov, E., Fine, S., Carmel, D., Darlow, A.: Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In: SIGIR. (2005) 512–519
2. Yom-Tov, E., Fine, S., Carmel, D., Darlow: Metasearch and Federation using Query Difficulty Prediction. In: Predicting Query Difficulty - Methods and Applications. (August 2005)
3. Carmel, D., Yom-Tov, E., Darlow, A., Pelleg, D.: What makes a query difficult? In: SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference

on Research and development in information retrieval, New York, NY, USA, ACM Press (2006) 390–397

4. Amati, G., Carpineto, C., Romano, G.: Query difficulty, robustness and selective application of query expansion. In: Proceedings of the 25th European Conference on Information Retrieval ECIR 2004. (2004)
5. Kwok, K.: An attempt to identify weakest and strongest queries. In: ACM SIGIR'05 Query Prediction Workshop. (2005)
6. Cronen-Townsend, S., Zhou, Y., Croft, W.: Predicting query performance. In: In Proceedings of the ACM Conference on Research in Information Retrieval (SIGIR). (2002)
7. Zhou, Y., Croft, W.B.: Ranking robustness: A novel framework to predict query performance. Technical Report IR-532, Center for Intelligent Information Retrieval, University of Massachusetts, Amherst (2006) URL: `http://maroo.cs.umass.edu/pub/web/674`.
8. Lin, J.: Divergence measures based on the shannon entropy. IEEE Trans. Infor. Theory **37** (1991) 145–151
9. Voorhees, E.M., Harman, D.: Overview of the Fifth Text REtrieval Conference (TREC-5). In: TREC. (1996)
10. Voorhees, E.M., Harman, D.: Overview of the Sixth Text REtrieval Conference (TREC-6). In: TREC. (1997) 1–24
11. Voorhees, E.M., Harman, D.: Overview of the Seventh Text REtrieval Conference (TREC-7). In: Proceedings of the Seventh Text REtrieval Conference (TREC-7). (1999) 1–24
12. Voorhees, E.M., Harman, D.: Overview of the Eighth Text REtrieval Conference (TREC-8). In: Proceedings of the Eighth Text REtrieval Conference (TREC-8). (2000) 1–24
13. Voorhees, E.M.: The TREC robust retrieval track. SIGIR Forum **39**(1) (2005) 11–20
14. Clarke, C., Craswell, N., Soboroff, I.: The TREC terabyte retrieval track. (2004)
15. Clarke, C.L.A., Scholer, F., Soboroff, I.: The TREC 2005 terabyte track. In: Proceedings of the Fourteenth Text REtrieval Conference (TREC 2005). (2005)
16. Macdonald, C., He, B., Ounis, I.: Predicting query performance in intranet search. In: ACM SIGIR'05 Query Prediction Workshop. (2005)
17. Cover, T.M., Thomas, J.A.: Elements of Information Theory. John Wiley & Sons (1991)
18. Aslam, J.A., Pavlu, V., Yilmaz, E.: A statistical method for system evaluation using incomplete judgments. In Dumais, S., Efthimiadis, E.N., Hawking, D., Jarvelin, K., eds.: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press (August 2006) 541–548
19. Aslam, J.A., Pavlu, V., Yilmaz, E.: Measure-based metasearch. In Marchionini, G., Moffat, A., Tait, J., Baeza-Yates, R., Ziviani, N., eds.: Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press (August 2005) 571–572