Improved Query Difficulty Prediction for the Web

Claudia Hauff[*]
University of Twente
Human Media Interaction
P.O.Box 217 NL-7500 AE
Enschede, Netherlands
c.hauff@ewi.utwente.nl

Vanessa Murdock
Yahoo! Research - Barcelona
Ocata 1
08003 Barcelona Spain
vmurdock@yahoo-inc.com

Ricardo Baeza-Yates
Yahoo! Research - Barcelona
Ocata 1
08003 Barcelona Spain
rbaeza@acm.org

ABSTRACT

Query performance prediction aims to predict whether a query will have a high average precision given retrieval from a particular collection, or low average precision. An accurate estimator of the quality of search engine results can allow the search engine to decide to which queries to apply query expansion, for which queries to suggest alternative search terms, to adjust the sponsored results, or to return results from specialized collections. In this paper we present an evaluation of state of the art query prediction algorithms, both post-retrieval and pre-retrieval and we analyze their sensitivity towards the retrieval algorithm. We evaluate query difficulty predictors over three widely different collections and query sets and present an analysis of why prediction algorithms perform significantly worse on Web data. Finally we introduce *Improved Clarity*, and demonstrate that it outperforms state-of-the-art predictors on three standard collections, including two large Web collections.

Categories and Subject Descriptors

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

General Terms

Algorithms, Performance

Keywords

query clarity, query performance prediction

1. INTRODUCTION

Query performance prediction aims to predict whether a query will have a high average precision in a given document collection ("easy" queries), or low average precision ("difficult" queries). Such a prediction based on search engine results is a potentially useful tool for search engines. An accurate estimator of the quality of search engine results can allow the search engine to decide to which queries to apply query expansion, for which queries to suggest alternative search terms, to adjust the sponsored results, or to return results from specialized collections.

Accurate query prediction can help the user to better understand how to find information in large scale collections such as the Web. The search engine can adjust its results based on the performance prediction, possibly searching a second collection or adding results to the current list if necessary to better serve the user.

Query performance prediction algorithms fall into two broad categories: pre-retrieval prediction and post-retrieval prediction. In pre-retrieval prediction, the query is evaluated before the retrieval step without considering the ranked list of results. The advantage of such algorithms is that they can be computed quickly, using statistics that are available from the collection or query history, before the search engine makes the computational expense of producing the ranking. A disadvantage of such predictors is that by not taking into account the specific retrieval algorithms, the predictions may not be as accurate.

Post-retrieval prediction algorithms are more complex. They either compare the ranked list to the collection as a whole, or different rankings produced by perturbing the query or documents. The first post-retrieval algorithm proposed was *Clarity Score*, which measures a query's ambiguity towards a collection by creating a language model from the top-retrieved documents and comparing it to the collection language model. Several methods have been proposed following from Clarity, designed with the Web in mind. One of the problems of these algorithms is their sensitivity to a change in parameters as well as to the retrieval algorithm. Currently, one needs to search exhaustively through the parameter space in order to find a reasonable setting. When relevance judgments are not available, this is not possible.

While query prediction algorithms have been shown to work well on established news report test collections such as TREC Volumes 4+5, they generally fail on Web test collections such as WT10g. The reasons for this failure are not well understood. In this paper, we investigate several state-of-the-art query prediction algorithms and propose *Improved Clarity*, a query performance predictor that is less sensitive to the retrieval algorithm. The parameters for Improved Clarity are set automatically in two ways: (1) by constraining the terms included in the KL divergence calculation between the query language model and the collection

[*] Work done while the author was an Intern at Yahoo! Research - Barcelona

CIKM'08, October 26–30, 2008, Napa Valley, California, USA.
Copyright 2008 ACM 978-1-59593-991-3/08/10 ...$5.00.

language model and (2) by determining the number of feedback documents in a query-dependent manner. Improved Clarity outperforms both established and newly proposed query prediction algorithms.

To summarize, the main contributions of this work are:

- An evaluation of state of the art query prediction algorithms, both post retrieval and pre-retrieval and their sensitivity toward the particular retrieval algorithm used.
- An evaluation of query difficulty predictors over three widely different collections and query sets. In the literature typically only a subset of the collections and topics is presented, but since the performance of predictors varies widely with a change of collection and query it is important to present all three collections.
- An analysis of why prediction algorithms perform significantly worse on Web collections.
- An improved query performance prediction algorithm that outperforms state-of-the-art predictors on all collections presented.

The rest of the paper is organized as follows. Section 2 gives an overview of related work, followed by a discussion of our proposed algorithms in Section 3. Sections 4 and 5 outline the experiments and Section 6 discusses the results. The paper closes with conclusions and possibilities for future work (Section 7).

2. RELATED WORK

In this section we discuss work related to query prediction algorithms. Pre-retrieval algorithms either take into account the frequencies of the query terms in the collection, such as *Averaged IDF* or *Simplified Clarity Score*, or the co-occurrence of query terms in the collection, such as *Averaged Pointwise Mutual Information* (PMI).

Post-retrieval algorithms are more diverse. *Clarity Score* [4] relies on the difference between the language model of the collection and the language model of the top retrieved documents. *Query Feedback* [17] considers the query drift when the top ranked documents are used to create a new query. Document and query perturbation algorithms slightly change the query terms or the top retrieved documents and consider the amount of change [13]. A method proposed by Aslam et al. [2] measures whether the overlap between the top ranked documents is similar for a range of different retrieval algorithms.

2.1 Pre-retrieval Algorithms

Pre-retrieval predictors rely only on the collection statistics of the query terms. *Averaged IDF* takes the average inverse document frequency over all query terms:

$$AvIDF(Q) = \frac{1}{m}\sum_{i=1}^{m} \log \frac{|C|}{|D_{q_i}|}. \qquad (1)$$

where Q is a query composed of m terms q_i, $|C|$ is the number of documents in the collection, and $|D_{q_i}|$ is the number of documents containing term q_i. Queries with low frequency terms are predicted to achieve a better performance than queries with high frequency terms.

He et al. [6] evaluated a number of algorithms including *Query Scope* and *Simplified Clarity Score.* Query Scope bases the prediction on the number of documents in the collection that contain at least one of the query terms. Simplified Clarity Score is very similar in spirit to Averaged IDF, but instead of document frequencies it relies on term frequencies:

$$SCS(Q) = \sum_{q_i \in Q} P_{ml}(q_i|Q) \times \log_2 \frac{P_{ml}(q_i|Q)}{P_{coll}(q_i)} \qquad (2)$$

where $P_{ml}(q_i|Q)$ is the maximum likelihood estimator of q_i given Q. $P_{coll}(q_i)$ is set as the term count of q_i in the collection divided by the total number of terms in the collection.

A final pre-retrieval predictor is *Averaged PMI*, which measures the average mutual information of two query terms in the collection, averaged over all query term pairs:

$$AvPMI(Q) = \frac{1}{|(q_i,q_j)|} \sum_{(q_i,q_j)\in Q} \log_2\left(\frac{P_{coll}(q_i,q_j)}{P_{coll}(q_i)P_{coll}(q_j)}\right) \qquad (3)$$

$P_{coll}(q_i, q_j)$ is the probability that q_i and q_j occur in the same document. $AvPMI$ is zero for single term queries.

2.2 Post-retrieval Algorithms

Cronen-Townsend et al. [4] introduced Clarity Score which, as already mentioned, measures a query's ambiguity towards a collection. The approach is based on the intuition that the top ranked results of an unambiguous query will be topically cohesive and terms particular to the topic will appear with high frequency. The term distribution of an ambiguous query on the other hand is assumed to be more similar to the collection distribution, as the top ranked documents cover a variety of topics. For instance, *artists who died in the 1700's* (TREC title query 534[1]) is likely to perform poorly as keyword based retrieval approaches will find documents with the terms "artist", "die" or "1700" in them, which includes a broad set of topics. An extension of Clarity Score that takes into account the temporal profiles of the queries was proposed by Diaz et al. [5].

Yom-Tov et al. [14] compared the ranked list of the original query with the ranked lists of the query's constituent terms. The idea behind the approach is that for well performing queries the result list does not change considerably if only a subset of query terms is used. They applied machine learning approaches, exploiting several features, among others the overlap in the top ranked documents between the original query and the subqueries, the score of the top ranked document and the number of query terms. An alternative based on the same idea was proposed by Aslam et al. [2]: a query is considered to be difficult if different ranking functions retrieve diverse ranked lists. If the overlap between the top ranked documents is large across all ranked lists, the query is deemed to be easy. For evaluation purposes the prediction scores are correlated against the average and median precision created from all submitted TREC runs.

Zhou and Croft [17] investigated two approaches to estimating query difficulty in Web search environments. *Weighted Information Gain* measures "the change in information about the quality of retrieval from an imaginary state that only an

[1] The average precision for this query using query likelihood retrieval with Dirichlet smoothing is 0.0047

average document is retrieved [estimated by the collection model] to a posterior state that the actual search results are observed". *Query Feedback* frames query prediction as a communication channel problem. The input is query Q, the channel is the retrieval system and the ranked list L is the noisy output of the channel. From the ranked list L, a new query Q' is generated, a second ranking L' is retrieved with Q' as input and the overlap between L and L' is used as prediction score. The lower the overlap between the two rankings, the higher the query drift and thus the more difficult the query. Experiments on GOV2 show considerable improvements over Clarity Score. The parameters of Query Feedback are the number $t = |Q'|$ of terms Q' consists of and the number of top ranked documents s considered for overlap between L and L'.

In the remainder of this paper we investigate pre- and post-retrieval predictors in terms of their stability and their correlation with average precision.

3. IMPROVED CLARITY SCORE

In looking for a reliable predictor of query performance in a Web environment, we turned to the Clarity Score predictor [4] which has been shown to correlate well with average precision. Clarity Score's performance, as all other prediction algorithms, is dependent on the collection, the retrieval setting and the query set.

3.1 Clarity Score

To compute the Clarity Score, the ranked list of documents returned for a given query are used to create a query language model [10] where terms that often co-occur in documents with query terms receive higher probabilities

$$P_{qm}(w) = \sum_{D \in R} P(w|D)P(D|Q). \tag{4}$$

R is the set of retrieved documents, w is a term in the vocabulary, D is a document, and Q is a query. In the query model, $P(D|Q)$ is estimated using Bayesian inversion:

$$P(D|Q) = P(Q|D)P(D) \tag{5}$$

where the prior probability of a document $P(D)$ is zero for documents containing no query terms.

Typically, the probability estimations are smoothed to give non-zero probability to terms not appearing the query, by redistributing some of the collection probability mass:

$$\begin{aligned} P(D|Q) &= P(Q|D)P(D) \\ &= P(D)\prod_i P(q_i|D) \\ &\approx P(D)\prod_i \lambda P(q_i|D) + (1-\lambda)P(q_i|C) \end{aligned} \tag{6}$$

where $P(q_i|C)$ is the probability of the ith term in the query, given the collection, and λ is a smoothing parameter. The parameter λ is constant for all query terms, and is typically determined empirically on a separate test collection.

The Clarity Score is the Kullback-Leibler (KL) divergence between the query language model P_{qm} and the collection language model P_{coll}:

$$D_{KL}(P_{qm}||P_{coll}) = \sum_{w \in V} P_{qm}(w) \log \frac{P_{qm}(w)}{P_{coll}(w)} \tag{7}$$

The larger the KL score, the more distinct is the query language model from the collection language model. The only parameter of Clarity Score is the number of top ranked documents (the number of *feedback documents*) to sample the query language model from.

In the following two sections we introduce *Improved Clarity*, which differs in two key ways from Clarity Score. First the number of feedback documents is set automatically. Second the term selection is made dependent on the frequency of the terms in the collection. These changes improve results, as we will show in the Section 5.

3.2 Setting the Number of Feedback Documents Automatically

Setting the number of feedback documents to a fixed value for all queries has been the standard so far. In Cronen-Townsend et al. [4] it is suggested that the exact number of feedback documents used is of no particular importance and 500 feedback documents is proposed as sufficient. In Section 5.1 experimental results show that the prediction performance of Clarity Score indeed depends on the number of feedback documents. While the prediction quality is high in comparison with other prediction algorithms, it requires an exhaustive search through the parameter space. Without such a search, the predictor may have very low performance.

In real-world situations, such a dependence on the tuning of the parameter in order to achieve meaningful prediction can have disastrous effects if training on one query set does not translate to another query set. Preferably, it should be possible to set parameters automatically such that performance on the evaluation set is close to or better than the best performing parameter setting.

In computing Clarity Score, if the query language model is created from a mixture of topically relevant and off-topic documents, its score will be lower compared to a query language model that is made up only of topically relevant documents, due to the increase in vocabulary size of the language model and the added noise. For example, consider TREC title query 476 "*Jennifer Aniston*". If the query language model not only includes documents containing both terms, but also documents containing the term "Jennifer" but not the term "Aniston", essentially, a focused query is turned into an ambiguous one, since added to the query language model are the same documents that would have been returned for the query "Jennifer". The term "Aniston" on the other hand is an important term in the query, as it disambiguates the term "Jennifer". Thus, preferably the query language model should be created from documents containing "Jennifer Aniston".

In a retrieval setting, we assume there is a vocabulary mismatch between how users express their need, and how a relevant document expresses the same information. Thus in a retrieval setting we may choose to smooth the probability estimates for unseen terms, or to assign probabilities to terms that are not in the query, in the interest of casting a wider net in hopes of finding information to satisfy the user.

In estimating the difficulty of a given query, we are not interested in estimating the difficulty of a query the user *might* have submitted. Instead we are operating on the terms at hand; we only care about the ambiguity of *this* query, composed of these exact terms. Every term in the query is mandatory for the purpose of predicting the ambiguity of the query.

Instead of fixing λ to a single value over the entire vocabulary as is the case in Equation 6, we turn to Hiemstra's Term-Specific Smoothing [7] which applies a smoothing weight specific to each query term:

$$P(D|Q) \approx P(D) \prod_i \lambda_i P(q_i|D) + (1 - \lambda_i) P(q_i|C) \qquad (8)$$

Setting $\lambda_i = 1$ for all query terms q_i, enforces the constraint that all query terms must be present in the document, or the document will receive a score of zero. One issue with this formulation for estimating a language model is that the language model, although it reflects documents containing the mandatory terms, itself is no longer smoothed. For this reason, we add an additional smoothing parameter β which determines the amount of smoothing with the collection language model:

$$\begin{aligned} P(D|Q) \approx P(D) \prod_i \lambda_i \Big(\beta P(q_i|D) + (1-\beta) P(q_i|C) \Big) \\ + (1 - \lambda_i) P(q_i|C) \end{aligned} \qquad (9)$$

Thus, the query language model is created only from documents that contain all query terms. This effectively sets the number of feedback documents in the Clarity Score algorithm automatically: for each query, the number of feedback documents utilized in the generation of the query language model is equal to the number of documents in the collection containing all query terms.

In some instances, not a single document in the collection contains all query terms. If this is the case, the constraint $\lambda_i = 1 \ \forall i$ is relaxed and documents containing $m - 1$ query terms are included in the query language model generation.

3.3 Frequency-Dependent Term Selection

The performance of Clarity Score depends on the initial retrieval run. In the language modeling approach to information retrieval [11], Clarity Score performs better with retrieval algorithms relying on a small amount of smoothing. Since increased smoothing often increases the mean average precision [16], retrieval with more smoothing is preferred. Hence, we would like to improve Clarity Score for retrieval runs with more smoothing. Increasing smoothing also increases the influence of high frequency terms on the KL divergence calculation (Equation 7), despite the fact that terms with a high document frequency do not aid in retrieval and therefore should not have a strong influence on the prediction score.

Thus we would like to minimize the contribution of terms that have a high document frequency in the collection. The situation is similar in a retrieval setting where we estimate a query model using feedback documents. One proposed solution [15], uses expectation maximization (EM) to learn a separate weight for each of the terms in the set of feedback documents. In doing this they reduce noise from terms that are frequent in the collection, as they have less power to distinguish relevant from nonrelevant documents. A similar approach is proposed in Hiemstra et al. [8]. The effect of both approaches is to select the terms that are frequent in the set of feedback documents, but infrequent in the collection as a whole.

Web retrieval requires speed. Running EM to convergence, although principled, would be computationally impractical. To approximate the effect of selecting terms frequent in the query model, but infrequent in the collection, we select the terms from the set of feedback documents that appear in $N\%$ of the collection, where $N = \{1, 10, 100\}$. We leave the comparison of a fixed document frequency-based threshold and a variable EM-based threshold to future work.

4. EXPERIMENTAL SETUP

The quality of a query prediction algorithm depends on several factors:

- Retrieval algorithm (for post-retrieval algorithms only)
- Parameters of the prediction algorithm
- Query set
- Collection
- Correlation measure

Each of these factors has a significant influence on the performance of the query prediction algorithm. A slight change in retrieval parameters that does not change the mean average precision, can have a significant effect on the performance of post-retrieval prediction algorithms. In our experiments we relied on the language modeling approach to information retrieval [11], specifically language modeling with Dirichlet smoothing [16]. Documents are ranked according to the likelihood of generating the query $P(Q|D)$:

$$P(Q|D) = \prod_{q_i \in Q} P(q_i|D). \qquad (10)$$

Since the document language models are very sparse, they are smoothed with the collection language model. Dirichlet smoothing is document dependent, and longer documents are smoothed less than shorter documents. The larger the smoothing parameter μ the more smoothing is applied:

$$P(q_i|D) = \frac{tf(q_i, D) + \mu P_{coll}(q_i)}{\sum_w tf(w, D) + \mu} \qquad (11)$$

$tf(w, D)$ is the term frequency of term w in document D. In our experiments, μ was evaluated for the values 100, 500, 1000, 1500, 2000, 2500, 3000 and 5000.

The three test collections and query sets used in our experiments are listed in Table 1. The collections were indexed with Indri[2], and stopped and stemmed using the Krovetz stemmer [9]. TREC Volumes 4+5 (without the Congressional Records) is a collection consisting of more than half a million news reports. WT10g contains three times as many documents and was derived from a crawl of the Web. WT10g [12] is noisy, it contains spam pages and pages with just 1 or 2 terms. GOV2 [3] is by far the largest collection, containing more than 25 million documents. It was created from a crawl of the .gov domain and therefore it is likely to be less noisy than WT10g, and more topically cohesive because although government documents cover a wide range of topics, they are not as diverse as the topics covered on the Web in general. To some extent the GOV2 collection is more like an Intranet than a Web collection. We evaluated the topics in sets of 50 queries each and considered only the title part of each topic as queries containing 2-3 terms are more realistic for a Web environment than longer queries.

[2] http://www.lemurproject.org

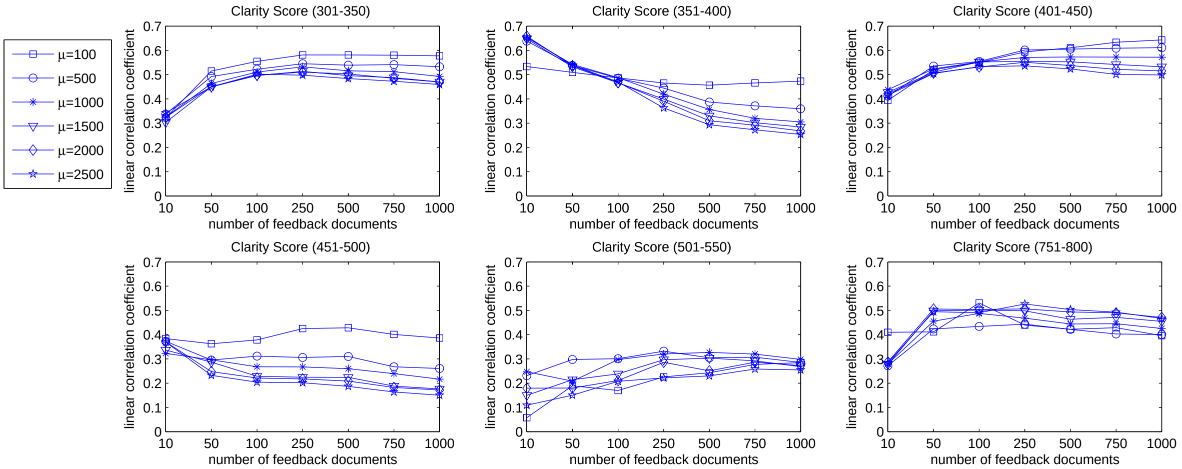

Figure 1: Sensitivity of Clarity Score towards the collection, the smoothing parameter of the retrieval approach and the number of feedback documents.

	Trec Volumes 4+5 (minus CR)	WT10g	GOV2
#documents	528155	1678849	25288903
#unique terms	764376	5506744	32933168
av. doc. length	267.848	379.111	664.904
topics	301-450	451-550	701-800
av. topic length	2.48	2.70	3.02

Table 1: Overview of test collections and topics. The corpora and topics were Krovetz stemmed and stopwords were removed.

	topics	μ	map
TREC Vol. 4+5	301-350	500	0.2274
	351-400	2000	0.1897
	401-450	100	0.2447
WT10g	451-500	1000	0.2073
	501-550	2000	0.1890
GOV2	701-750	1000	0.2687
	751-800	1000	0.3244

Table 2: Mean average precision (map) of the best performing retrieval run for language modeling with Dirichlet smoothing μ.

Query prediction algorithms are evaluated by correlating the average precision scores of the queries with their prediction scores. The *linear correlation coefficient* measures the strength and direction of a linear relationship between average precision and prediction scores. The coefficient is $+1$ (-1) in case of a perfect increasing (decreasing) linear relationship and a value in between otherwise. A coefficient of zero means that no linear relationship exists. The disadvantage of linear correlation is the assumption of a linear relationship which may not hold. Nonetheless we include this metric to make our work comparable to other work reporting this metric.

The *Kendall's tau rank correlation coefficient* is a parameter-free measure where the scores are first converted to ranks. A perfect agreement between the average precision and prediction rankings results in a coefficient of $+1$, a perfect reverse ranking in -1 and a value inbetween otherwise. The two correlation coefficients give very different results for different predictors and different collections, so to clarify this in our experiments we report both the linear correlation coefficient and Kendall's tau rank correlation coefficient.

5. RESULTS

In Section 5.1 Clarity Score and Query Feedback are investigated for their sensitivity towards the factors listed in the previous section. Section 5.2 gives an overview of the pre-retrieval and post-retrieval baselines. Section 5.3 evaluates *Improved Clarity*.

5.1 Sensitivity Analysis of Clarity Score and Query Feedback

This section emphasizes the influence of the different factors affecting prediction by giving examples of the behavior of Clarity Score and Query Feedback as their parameters vary. Figure 1 shows the linear correlation coefficients of Clarity Score when varying the number of feedback documents and the smoothing parameter. The top row of Figure 1 displays the behavior of the query sets of TREC Volumes 4+5. While topics 301-350 are relatively insensitive to the specific number of feedback documents and do not show much of a change once 250 feedback documents are reached, topics 351-400 exhibit a very different behavior. At 10 feedback documents and $\mu = 2000$ the correlation is as high as 0.66, at 1000 feedback documents the correlation has degraded to 0.27.

The bottom row of Figure 1 shows the results of the two query sets of the WT10g collection, which have a considerably lower correlation overall, and one query set of GOV2. The influence of the smoothing parameter is also visible - in general, the lower the amount of smoothing, the higher the correlation coefficient. Low smoothing however, often does not result in the best retrieval performance as measured in mean average precision as shown in Table 2. The table contains for all query sets the smoothing parameters that result in the highest mean average precision. In most

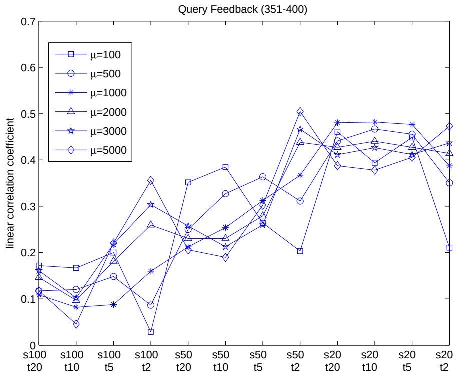

Figure 2: Sensitivity of Query Feedback towards its parameters and the smoothing parameter of the retrieval approach.

	topics	best	standard	worst
TREC Vol. 4+5	301-350	0.5451	0.5390	0.3375
	351-400	0.6587	0.3095	0.2678
	401-450	0.5727	0.5727	0.4382
WT10g	451-500	0.3227	0.2595	0.2168
	501-550	0.2866	0.2508	0.1796
GOV2	701-750	0.6351	0.6033	0.4064
	751-800	0.4877	0.4441	0.2789

Table 3: Linear correlation coefficients of the best, standard (500 feedback documents) and worst performing Clarity Score with respect to the retrieval run with the best map as given in Table 2.

cases, $\mu \geq 1000$ performs best. For the topics 451-500 for instance, the highest correlation (0.3538) was achieved for $\mu = 100$ and 500 feedback documents. However, the mean average precision of that retrieval run is only 0.1469, significantly worse than the mean average precision of the best performing run: 0.2000.

To stress the point that the standard setting of 500 feedback documents may be inadequate, in Tables 3 and 4 the linear and Kendall's correlation coefficients are presented for Clarity Score with 500 feedback documents (standard) as well as the results of the best and worst performing feedback document setting.

Figure 2 shows Query Feedback's sensitivity to changes in its parameter settings exemplary for topics 351-400. On the x-axis the parameter pairs (s, t) are given. Recall, that s is the number of top ranked documents evaluated for their overlap between the two ranked lists L and L' and t is the number of terms query Q' consists of. The effect of smoothing on the prediction quality is reversed compared to Clarity Score: the lower the smoothing, the less well the algorithm performs. Finding the right parameter pair is as important as for Clarity Score - the linear correlation coefficient can be as low as 0.1086 and as high as 0.5045.

The conclusions we draw from this analysis are that both Clarity Score and Query Feedback are highly sensitive to both the initial retrieval parameter tuning, as well as their own parameters. Furthermore, parameters tuned to one query set do not produce reliable results for other query sets. Even when the query set and the collection are fixed, the performance of the predictor varies widely depending on the setting of the parameters of the metric. This instability makes them less useful in a Web environment where we cannot assume that one query will have any relation to the next, and the results returned for a given query may or may not be the same the next time the query is posed to the system.

	topics	best	standard	worst
TREC Vol. 4+5	301-350	0.4361	0.4198	0.3022
	351-400	0.5031	0.2172	0.1552
	401-450	0.3665	0.3045	0.3045
WT10g	451-500	0.3003	0.1285	0.1182
	501-550	0.2432	0.2228	0.0527
GOV2	701-750	0.4752	0.4149	0.2571
	751-800	0.3773	0.3299	0.2434

Table 4: Kendall tau correlation coefficients of the best, standard (500 feedback documents) and worst performing Clarity Score with respect to the retrieval run with the best map as given in Table 2.

5.2 Query Prediction Baselines

In all reported experiments that follow, the smoothing parameter μ was set to the value given in Table 2 for each query set. The prediction algorithms are thus evaluated for their performance on the best performing retrieval setting. Although lower performing retrieval runs can result in higher correlation scores, a badly performing retrieval algorithm with a higher prediction accuracy is less desirable than a well performing retrieval algorithm with lower prediction performance. Furthermore, if more queries are performing badly, the prediction becomes easier because the topics that perform well regardless of smoothing are likely very specific queries, and are not difficult to predict in any setting.

Due to the influence of the retrieval run on the prediction accuracy, we reimplemented a number of query prediction algorithms instead of citing scores from other papers, as not all metrics have been reported on all three collections. The three pre-retrieval baselines are Averaged IDF (Equation 1), Simplified Clarity Score (Equation 2) and Averaged PMI (Equation 3). These three performed best from a pool of 12 pre-retrieval algorithms evaluated. Clarity Score and Query Feedback [17] as described in Section 2 were used as post-retrieval baselines. While the pre-retrieval baselines are parameter-free, the post-retrieval parameter settings are as follows: the original Clarity Score results are reported with the standard setting of 500 feedback documents; since for Query Feedback no standard parameter setting exists, the parameters were tuned on one query set and then utilized on the second (and third) query set of the collection. Thus, for example, the parameters for topics 451-500 were set using topics 501-550 and vice versa. In Tables 5 and 6 we report the linear correlation coefficient and Kendall's correlation coefficient found as well as the average correlation and standard deviation over all query sets.

5.3 Improved Clarity

Tables 5 and 6 also contain the correlation coefficients of Improved Clarity. The runs marked with *fixed* have the same fixed number of feedback documents for all queries as well as frequency-dependent term selection. To make the

		TREC Volumes 4+5			**WT10g**		**GOV2**			
approach	**N**	**301-350**	**351-400**	**401-450**	**451-500**	**501-550**	**701-750**	**751-800**	mean	std. dev.
Averaged IDF		0.5908	0.3739	0.5756	0.1425	0.2211	0.3927	0.3235	0.3743	0.1671
Simplified CS		0.5775	0.3193	0.5179	0.0813	0.1893	0.3251	0.2898	0.3286	*0.1732*
Averaged PMI		0.3158	0.3763	0.4384	0.2897	0.2346	0.4303	0.4650	0.3643	0.0865
Query Feedback		0.3179	0.4271	0.3820	0.2902	0.2160	0.6016	0.5348	0.3957	0.1369
Clarity Score	100%	0.5390	0.3095	0.5727	0.2595	0.2508	0.6033	0.4441	0.4256	0.1517
Improved Clarity	10%	0.6561	0.4092	0.5721	0.3482	0.2528	0.5269	0.4670	0.4618	0.1376
(Fixed)	1%	**0.6643**	0.4427	0.6744	0.5451	0.1990	0.5267	0.4261	0.4969	0.1630
Improved Clarity	100%	0.5490	0.4848	0.6663	0.4255	**0.3966**	**0.6192**	**0.6033**	0.5350	0.1024
(Automatic)	10%	0.6289	**0.5286**	0.6393	0.4284	0.3660	0.5765	0.6020	0.5385	0.1047
	1%	0.6330	0.5106	**0.7064**	**0.5917**	0.2806	0.5422	0.5498	**0.5524**	0.1354

Table 5: Linear correlation coefficients of the baselines with respect to the retrieval run with the best map as given in Table 2.

		TREC Volumes 4+5			**WT10g**		**GOV2**			
approach	**N**	**301-350**	**351-400**	**401-450**	**451-500**	**501-550**	**701-750**	**751-800**	mean	std. dev.
Averaged IDF		0.3136	0.2711	0.3127	0.2288	0.1871	0.2766	0.2630	0.2647	0.0451
Simplified CS		0.2679	0.2270	0.2767	0.1642	0.1361	0.2110	0.2564	0.2199	0.0543
Averaged PMI		0.1758	0.2903	0.2321	0.2110	0.2119	0.3014	0.3145	0.2481	0.0535
Query Feedback		0.2944	0.2735	0.2241	0.2374	0.1595	**0.4320**	0.4200	0.2916	0.1012
Clarity Score	100%	0.4198	0.2172	0.3045	0.1285	0.2228	0.4149	0.3299	0.2911	0.1081
Improved Clarity	10%	0.4737	0.3038	0.3976	0.2254	0.2245	0.3475	0.3593	0.3331	0.0904
(Fixed)	1%	0.4851	0.3446	0.4971	0.3445	0.1599	0.3511	0.3103	0.3561	0.1136
Improved Clarity	100%	0.4230	0.3757	0.4482	0.2169	**0.2857**	0.4202	0.4410	0.3730	0.0885
(Automatic)	10%	0.4606	0.3969	0.4645	0.2595	0.2772	0.3972	**0.4573**	0.3876	0.0865
	1%	**0.4998**	**0.4002**	**0.5624**	**0.3735**	0.1837	0.3723	0.4181	**0.4014**	*0.1190*

Table 6: Kendall tau correlation coefficients of the baselines with respect to the retrieval run with the best map as given in Table 2.

collection	**perplexity**	**# unique terms**
TREC Volumes 4+5	5622.49	764376
WT10g	10367.60	5506744
GOV2	4432.19	32933168

Table 7: Collection perplexity and vocabulary size.

results comparable with the original Clarity Score, the reported numbers are the correlation coefficients achieved with the standard setting of 500 feedback documents. The runs marked *automatic* have their number of feedback documents set automatically as described in Section 3.2. The parameter N determines the amount of frequency-dependent term selection. At $N = 100\%$, all terms independent of their document frequency are included in the KL divergence calculation, at $N = 10\%$ ($N = 1\%$) only terms occurring in less than $\frac{1}{10}$th ($\frac{1}{100}$th) of the documents in collection are included.

5.4 Perplexity

Predicting the quality of queries 451-550 has proven to be the most difficult across all predictors. In a Web environment, there are potentially millions of relevant documents for a given query. We hypothesize that the language of news articles and government websites is less varied, and the documents in these collections are more topically cohesive than Web pages. A single Web page contains a large proportion of content not related to the topic of the page itself, and furthermore even among the set of Web pages relevant to a given query, there may be a large number of different genres represented. For example in a Web setting, the set of relevant results may include pages that are largely informational (such as Wikipedia pages), pages that are largely commercial in nature, personal home pages, spam pages that provide a link farm centered around a particular topic, blogs, etc. Whereas the TREC Volumes 4+5 and GOV2 collections can be expected to be free of noisy pages such as spam, WT10g is not. Moreover, web pages are also less likely to be focused and bound to a particular topic.

Furthermore, while the style for news articles is determined by a news organization and enforced to a large extent by the editors at that organization, on the Web the content is written by members of the general public with no style guidelines in place. Thus we hypothesize that one reason for the difficulty of predicting performance on the Web is the large variance in vocabulary, even among topically related documents.

Since Clarity Score builds on the hypothesis that relevant documents have a more focused term distribution than non-relevant documents this metric correlates less well with noisy relevant documents. We use perplexity as an indicator of the topical cohesion.

The perplexity of a discrete probability distribution P is defined as two to the power of the entropy:

$$Perplexity(P) = 2^{\sum_{w \in V} P(w) \log_2 P(w)} \tag{12}$$

where $P(w)$ is the maximum likelihood of term w in P. Perplexity is a measure of the uncertainty about a word, given a language model. In designing language models, the goal is to reduce the perplexity, and in this way reduce uncertainty about the terms in the language model. We conduct two experiments: we measure the perplexity of a language model built from a collection of documents, and we compare the average perplexity of a language model created from the top N documents retrieved for each query to the language model of the collection as a whole. The intuition is that if the language model of the collection has a high perplexity,

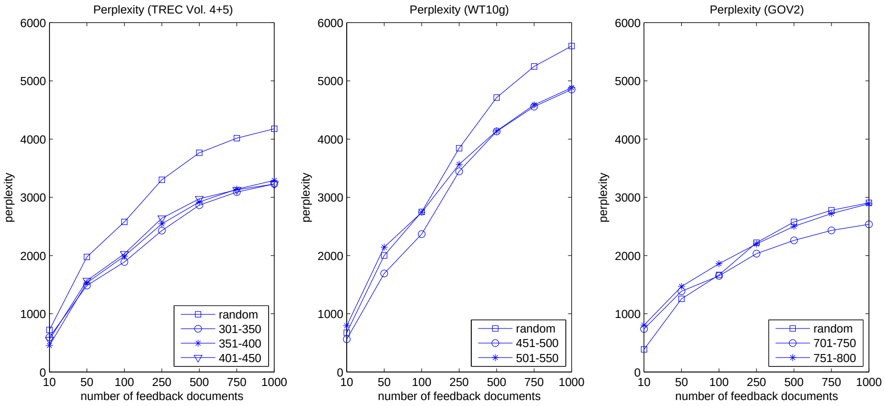

Figure 3: Perplexity results.

the collection as a whole has a less topically cohesive vocabulary. Additionally we would like for the language model of the top retrieved documents to have a lower perplexity than the language model of the collection.

The collection perplexities are given in Table 7. Although GOV2 is the largest collection, its perplexity is actually the smallest while WT10g's perplexity is the highest. This indicates that the language of GOV2 is actually much less varied than WT10g inspite of the fact that both represent Web collections.

In a second experiment, the top n ranked documents were concatenated for $n = \{10, 50, 100, 250, 500, 750, 1000\}$ and a slightly smoothed language model was created from the concatenated documents. The perplexity of this language model averaged over the 50 queries of each query set is shown in Figure 3. For comparison, the run marked *random* represents the average perplexity of 500 randomly generated ranked lists of documents. The perplexity of the top ranked documents of topics 301-451 (TREC Volumes 4+5) is distinctly lower than the perplexity of the same number of documents chosen randomly from the collection. The same cannot be said about the top ranked results of topics 451-550 (WT10g) however - their perplexity is close to *random*. We suspect that this is the reason for the poor performance of Clarity Score on WT10g. Note that eventually, as the number of feedback documents increases, the perplexity of the language models will converge to *random*, as the number of documents in the language model approaches the number of documents in the collection.

The perplexity of the language models for GOV2 also is much closer to than for topics 301-451, and the topics from Terabyte 2005 track with *random* for all settings of feedback documents, but the overall perplexity is much lower than for WT10g. More importantly, for GOV2, the perplexity seems to level off between 2000 and 3000 at 500 feedback documents, whereas for WT10g the perplexity continues to increase as the number of feedback documents increases.

These experiments indicate that the language models for TREC 4+5 and topics 301-451, and for GOV2, are much more topically unified than for WT10g. We attribute the difficulty in predicting query performance on the Web to the high degree of variability in the language, even among relevant documents on the Web.

6. DISCUSSION

On the Web, roughly 50% of queries are seen only once, thus it is virtually impossible to create a representative query sample with relevance judgements to tune parameters. Furthermore, as shown in Section 5.4, the relevant documents for a given Web query are noisy. Therefore, we require a performance predictor that is robust to differences in retrieval parameters, queries and document collections.

For short unambiguous queries, constraining the language model to documents containin all query terms adds less noise to the language model. For long queries, constraining the language model limits the noise contributed by terms that are unambiguous. For terms that are ambiguous, forcing their inclusion increases noise, but this is desirable because we are capitalizing on noise in the language model to identify ambiguous queries. In the case that a query is unambiguous, but contains non-content terms, we compensate by selecting terms from the language model that are infrequent in the collection. Thus in Improved Clarity non-content terms do not harm queries that are otherwise unambiguous.

Table 5 shows a big difference in the prediction performance over different collections. Averaged IDF for example varies between a correlation of 0.1425 and 0.5908. Furthermore, there are also fluctuations between query sets within a single collection albeit to a lesser extent. Averaged IDF and Simplified Clarity Score have very similar scores over all query sets due to their closeness in spirit. In all but one case, Averaged IDF and Simplified Clarity Score outperform Clarity Score and Query Feedback on TREC Volumes 4+5. The reverse is true for the query sets of WT10g and GOV2 where Clarity Score and Query Feedback perform better than the pre-retrieval predictors. Averaged PMI is the most stable of three pre-retrieval predictors; it performs not as well in some cases, but overall has no radical performance drop unlike Averaged IDF and Simplified Clarity Score. Query Feedback performs worse than Query Clarity overall, but it is the better predictor in some instances (e.g. query set 351-400).

Improved Clarity with frequency-dependent term selection (the runs are marked as *Fixed*) outperforms the original Clarity Score in 6 out of 7 cases for $N = 10\%$, and for more than half of the cases for $N = 1\%$. The largest improvements were achieved on TREC Volumes 4+5 whereas the performance decreases slightly on the GOV2 query sets. The mean performance across all query sets is larger for $N = 1\%$ than for $N = 10\%$, that is, the more common terms are removed from the KL divergence calculation, the better the method performs.

When reverting to the automatic setting of the number of feedback documents (the runs are marked as *Automatic*), in all but one instance Improved Clarity outperforms the original Clarity Score. Additionally relying on term-dependent term selection is mainly helpful for the query sets of TREC Volumes 4+5. While for one query set (301-350) Improved Clarity with the standard setting of the number of feedback documents shows the best performance, the best prediction performance for the remaining six query sets is achieved by Improved Clarity with the automatic setting of the number of feedback documents. Across all query sets, the automatic setting of Improved Clarity with $N = 1\%$ achieves the highest prediction performance with an average linear correlation coefficient of 0.5524. WT10g is the hardest collection to predict for all methods, possible reasons for it were discussed in the last Section.

In Table 6 the Kendall tau correlation coefficients of the experiments are presented. A general observation is that Kendall's coefficients are lower than the linear correlation coefficients presented in Table 5. The trend of the results however is similar - WT10g is still the most difficult collection to predict the performance for and Improved Clarity with an automatic setting of feedback documents and $N = 1\%$ outperforms all other tested approaches overall with a mean correlation coefficient of 0.4014. In Vinay et al. [13] a Kendall tau correlation of up to 0.5 is reported; those results however are not comparable to ours, as we restrict ourselves to the title part of the topics instead of the description.

7. CONCLUSIONS AND FUTURE WORK

This paper proposed two changes to Clarity Score, namely setting the number of feedback documents used in the estimation of the query language model individually for each query to the number of documents that contain all query terms, and ignoring high-frequency terms in the KL divergence calculation. These adaptations have been tested on three TREC collections: a corpus of news articles and two Web corpora. Apart from one set of queries, *Improved Clarity* outperforms the baselines in all cases, in some instances by a large margin. Furthermore, the gap between the highest and lowest correlation scores for different retrieval runs is decreased. While a difference remains between the performance of query prediction algorithms on WT10g and the two corpora TREC Volumes 4+5 and GOV2, we were able to improve the correlation significantly.

In the future, we plan to further investigate how to set the feedback document parameter more effectively, specifically by taking into account the dependency between the query terms. Going back to the example of TREC title query 476 (*"Jennifer Aniston"*), documents containing either both terms or only the term *Aniston* should be included in the query language model generation. Furthermore, the question of how best to set N automatically arises. Lastly, the noticeable difference in prediction performance of the two correlation measures needs to be addressed. One possible direction is to apply the query prediction algorithms to a specific problem such as selective query expansion [1].

8. REFERENCES

[1] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness and selective application of query expansion. In *Proceedings of the 25th European Conference on Information Retrieval*, pages 127–137, 2004.

[2] J. A. Aslam and V. Pavlu. Query hardness estimation using Jensen-Shannon divergence among multiple scoring functions. In *Advances in Information Retrieval: 28th European Conference on IR Research*, pages 198–209, 2007.

[3] C. Clarke, N. Craswell, and I. Soboroff. Overview of the trec 2004 terabyte track. In *Proceedings of the Thirteenth Text REtrieval Conference*, 2004.

[4] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 299–306, 2002.

[5] F. Diaz and R. Jones. Using temporal profiles of queries for precision prediction. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–24, 2004.

[6] B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *The Eleventh Symposium on String Processing and Information Retrieval (SPIRE)*, pages 43–54, 2004.

[7] D. Hiemstra. Term-specific smoothing for the language modeling approach to information retrieval: the importance of a query term. In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 35–41, 2002.

[8] D. Hiemstra, S. Robertson, and H. Zaragoza. Parsimonious language models for information retrieval. In *SIGIR '04: Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 178–185, 2004.

[9] R. Krovetz. Viewing morphology as an inference process. In *SIGIR '93: Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 191–202, 1993.

[10] V. Lavrenko and W. B. Croft. Relevance based language models. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 120–127, 2001.

[11] J. M. Ponte and W. B. Croft. A language modeling approach to information retrieval. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281, 1998.

[12] I. Soboroff. Does WT10g look like the web? In *SIGIR '02: Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 423–424, 2002.

[13] V. Vinay, I. J. Cox, N. Milic-Frayling, and K. Wood. On ranking the effectiveness of searches. In *SIGIR '06: Proceedings of the 29th annual international ACM*

SIGIR conference on Research and development in information retrieval, pages 398–404, 2006.

[14] 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 '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 512–519, 2005.

[15] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, 2001.

[16] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 334–342, 2001.

[17] Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 543–550, 2007.