The Combination and Evaluation of Query Performance Prediction Methods

Claudia Hauff[1], Leif Azzopardi[2], and Djoerd Hiemstra[1]

[1] University of Twente, The Netherlands
{c.hauff,d.hiemstra}@utwente.nl
[2] University of Glasgow, United Kingdom
leif@dcs.gla.ac.uk

Abstract. In this paper, we examine a number of newly applied methods for combining pre-retrieval query performance predictors in order to obtain a better prediction of the query's performance. However, in order to adequately and appropriately compare such techniques, we critically examine the current evaluation methodology and show how using linear correlation coefficients (i) do not provide an intuitive measure indicative of a method's quality, (ii) can provide a misleading indication of performance, and (iii) overstate the performance of combined methods. To address this, we extend the current evaluation methodology to include cross validation, report a more intuitive and descriptive statistic, and apply statistical testing to determine significant differences. During the course of a comprehensive empirical study over several TREC collections, we evaluate nineteen pre-retrieval predictors and three combination methods.

1 Introduction

Predicting the retrieval performance or determining the degree of difficulty of a query is a challenging research area which has received a lot attention recently [13, 20, 19, 8, 16, 5, 7]. The aim is to create better methods (predictors) for the task, as a reliable and accurate prediction mechanism would enable the creation of more adaptive and intelligent retrieval systems. For instance, if the performance of the query is considered to be poor, remedial action can be taken by the system to try and ensure that the user's information needs are satisfied. This may be done through asking for refinement of the query, or some automatic disambiguation process. On the other hand, if the performance of a query appears sufficiently good, the query can be improved by some affirmative action such as automatic query expansion with pseudo relevance feedback.

Despite the numerous methods proposed, little research has been performed on combining the different predictors in a principled way. One of the questions explored in this paper is whether or not such a combination leads to improved *prediction accuracy* (w.r.t. the effectiveness measure used). In Section 2 we describe in more detail how the particular task should influence the choice of evaluation measure. In the case of distinguishing easy from difficult queries for

instance, rank correlation coefficients [1] are utilized to evaluate and compare different predictors. A higher correlation coefficient is generally deemed sufficient evidence to infer that one predictor outperforms another. However, due to the small query set size in current evaluations, this is generally incorrect as significance tests of the difference in correlation will show. Additionally, the currently employed evaluation measure for query performance prediction, namely the linear correlation coefficient r, is shown to be prone to failure. r is an indicator of the strength of the linear relationship between prediction values and retrieval performance values (e.g. Average Precision) for a set of queries. While this is reasonable when evaluating parameter-free methods such as the pre-retrieval predictors introduced later, a problem arises when combining different predictors: combination methods have a higher degree of freedom and can thus fit the set of predictor/retrieval performance values very well. While such an overfitting leads to a high r value, it generally lowers the prediction accuracy, that is the quality of the method when predicting values of unseen queries. Furthermore, what does r actually mean? The value itself, unless close to one or zero, is difficult to interpret and does not provide an intuitive indication of the quality of the prediction methods. A more intuitive measure would report the error on the prediction of the effectiveness values. This informs the researcher and practitioner of the utility of the method in terms of actual performance. Since the prediction accuracy is a much more meaningful evaluation measure of query performance prediction (QPP) than r, we adopt the methodology applied in machine learning and report the *root mean squared error* (RMSE) derived from training a linear model on a training set and evaluating it on a separate test set.

This paper critically examines the evaluation of QPP methods and addresses the aforementioned problems within the current evaluation methodology. Specifically, we focus on the evaluation and combination of pre-retrieval QPP methods, where we compare nineteen predictors and three combination methods that are novel to QPP. Our findings show, that while under the previous evaluation methodology the combined predictors considerably outperform single predictor methods (given r), this study reveals that the combined methods are only slightly better than the best single predictor (in terms of RMSE). After outlining the different types of tasks and the different types of QPP algorithms in Section 2, we consider the problems with the current evaluation methodology in Section 3. We propose how these issues can be resolved by using standard techniques employed in machine learning to combat effects of combining predictors and evaluating and comparing predictor performance. In Section 4 several penalized regression models are introduced as potential combination methods, which have shown to perform well in analogous prediction scenarios in microarray data analysis [14]. We performed an evaluation on three different TREC collections (Section 5), discuss the results (Section 6) and conclude the paper in Section 7.

2 Related Work and Background

A considerable number of methods have been proposed in recent years that attempt to provide an indication of a query's quality. There are two ways in which these methods can be considered: (i) time of estimate (pre/post-retrieval) and (ii) type of task (difficulty, rank, retrieval performance).
Pre-retrieval algorithms *predict* the quality of a query without considering the ranked list of results, whereas *post-retrieval* algorithms are employed after the retrieval stage. They *estimate* the quality of the given output of a ranking function. How "quality" is defined depends on the particular task at hand. Let $\mathbf{q}$ be a query, C be a corpus of documents, E be an external source[3] and let R be a ranking function. Then, query *difficulty* estimation can formally be defined as a classification task: $f_{diff}(\mathbf{q}, C, E, R) \rightarrow \{0, 1\}$, where 0 (1) indicates a poor (good) query. In the case of $R = \emptyset$ (pre-retrieval), we speak of prediction instead of estimation[4]. If we are interested in the particular ranking of a set of queries, for example to determine which of a pool of queries is the best representation of an information need, we estimate the queries' *performance*: $f_{perf}(\mathbf{q}, C, E, R) \rightarrow \mathbb{R}$, and the query with the highest score is considered to be the best. While f_{perf} can produce a ranking of queries, it does not directly estimate the Average Precision of a query, which is required for instance when comparing the performance of queries across different collections. In such cases, we rely on $f_{norm}(\mathbf{q}, C, E, R) \rightarrow [0, 1]$, which provides comparable *normalized* scores such as Average Precision.
In this paper, we concentrate on the evaluation of f_{norm}, but note that f_{perf} and f_{diff} can be obtained from f_{norm}. None of the predictors examined in this paper provides *predicted Average Precision* scores but unbounded scores in $\mathbb{R}$, so that linear regression (whose by-product is r) needs to be applied to obtain f_{norm}.
As the evaluation is based on pre-retrieval predictors' normalized performances and to avoid over complications, we will continue with query predictors instead of estimators and normalized performance instead of rank or difficulty, although the latter could be substituted in most cases in the context. In the following subsection, we provide an overview of the 19 pre-retrieval methods proposed in the literature which we utilized in our experiments.

2.1 Overview of Pre-Retrieval Predictors

QPP methods can be divided into four different groups according to the heuristic they exploit in making their prediction: specificity, ambiguity, term relatedness and ranking sensitivity.

Specificity The specificity based predictors predict a query to perform better with increased specificity. How the specificity is determined, further divides these

[3] External sources such as Wikipedia and WordNet are currently utilized in few algorithms.
[4] This distinction holds for all definitions that follow.

predictors into collection statistics based and query based predictors. **Average Query Length (*AvQL*)** [12] relies solely on the query terms. It is based on the assumption that longer terms are more specific. A number of collection statistics based specificity predictors have been proposed, which exploit the inverse term or inverse document frequencies. **Averaged Inverse Document Frequency (*AvIDF*)** [5] assumes the more discriminative the query terms on average, the better the query will perform. **Maximum Inverse Document Frequency (*MaxIDF*)** [13] on the other hand bases its prediction on the most discriminative term of all query terms only. A number of slight variations on *AvIDF* have been proposed such as those in [7]. They include:**Averaged Inverse Collection Term Frequency (*AvICTF*)** , **Query Scope (*QS*)**, **Simplified Clarity Score (*SCS*)** and **Standard Deviation of IDF (*DevIDF*)**. In [19], three predictors were proposed that combine the collection frequency and inverse document frequency, where the assumption is that a topic which is well covered in the collection is likely to perform well: **Summed Collection Query Similarity (*SumSCQ*)** , **Averaged Collection Query Similarity (*AvSCQ*)** and **Maximum Collection Query Similarity (*MaxSCQ*)**.

Ambiguity If a term always appears in the same or similar contexts across all documents, the term is considered to be unambiguous. Low ambiguity indicates an easy query. Instead of basing the predictors on the collection, ambiguity can also be calculated with an external source such as WordNet. Predictors in this category include: **Average Polysemy (*AvP*)** and **Average Noun Polysemy (*AvNP*)** [12], which assume that the higher the average number of WordNet [2] senses, the more ambiguous and the worse the query is likely to perform. **Average Set Coherence (*AvQC*)** and **Global Average Set Coherence (*AvQCG*)** [8] are collection based and cluster the documents containing the query terms to determine the number of possible senses (clusters) associated with a term, where the assumption engaged is more clusters will be indicative of poorer performance. It should be noted, that *AvQC* and *AvQCG* are the most computationally intensive pre-retrieval predictors considered here, as they rely on a document by document similarity calculation which for large collections needs to be approximated by sampling.

Term Relatedness The disadvantage of predictors in the first two categories stems from their lack of consideration for the relationship between query terms; the query *political field* for example is unambiguous due to the relationship between the two terms, but a specificity or ambiguity based predictor is likely to predict a poor performance. A strong relationship between query terms is assumed to be indicative of a well formed query which is likely to be successful. Predictors of this kind include: **Average Pointwise Mutual Information(*AvPMI*)** and **Maximum Pointwise Mutual Information (*MaxPMI*)** which capture the dependency between query terms, such that a high score is assumed to be correlated with better performance. The exploitation of the semantic relationships identified in WordNet [4] showed no correlation with retrieval performance and are thus omitted here.

Ranking Sensitivity Predictors in this category exploit the potential sensitivity of the result ranking by predicting how easy it will be for the retrieval method to rank the documents containing the query terms. If all documents "look the same" to the retrieval method, it is difficult to rank them and the query is deemed difficult. In [19], three predictors of this nature are proposed, they are: **Summed Term Weight Variability (*SumVAR*)**, **Averaged Term Weight Variability (*AvVAR*)** and **Maximum Term Weight Variability (*MaxVAR*)**. The predictors based on the distribution of term weights across the collection, *SumVAR*, *AvVAR* and *MaxVAR* require less intensive processing than *AvQC(G)*, although the precomputation of *tf.idf* weights for all collection terms can also considered to be computationally more intensive than for instance retrieving the inverse document frequencies.

Combinations Despite the considerable number of pre-retrieval predictors proposed, few attempts have been made to combine them. In [19] the proposed predictors are linearly combined, and the best performing combination is reported. As already pointed out, in the case of the linear correlation coefficient, a combination of predictors usually results in a higher correlation coefficient and is thus not necessarily indicative of improved query performance prediction abilities.

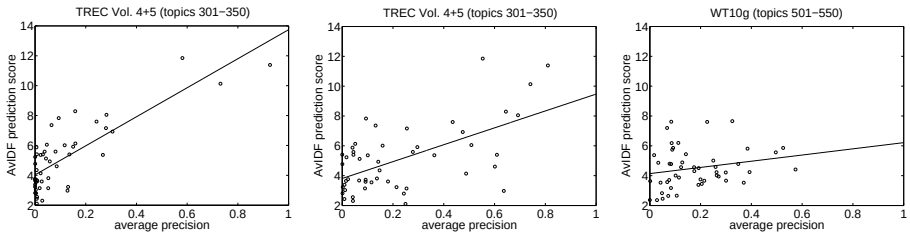
3 Evaluation Methodology

The standard correlation based approach to evaluation as performed in [8, 7, 19, 10, 12] is as follows. Let Q be the set of queries $\mathbf{q}$ and let $R_{\mathbf{q}}$ be the ranked list returned by the ranking function R for $\mathbf{q}$. For each $\mathbf{q} \in Q$, the predicted score v is obtained from a given predictor and the average precision p of $R_{\mathbf{q}}$ is determined. Given all pairs (v, p), the correlation coefficient is calculated and reported. This can either be the rank correlation coefficients Spearman's Rho or Kendall's Tau [1] which are applicable in the context of f_{diff} and f_{perf} or the linear correlation coefficient r which has been used to evaluate f_{norm}. Note that in [17, 16] two correlation-independent evaluations for f_{diff} have been proposed. Since the focus of this paper is on f_{norm}, we consider the linear correlation coefficient r. Using two examples, we show why it is difficult to interpret the r value and why it is prone to providing misleading conclusions, before addressing the shortcomings of the current methodology and conducting a comprehensive evaluation of QPP methods assuming the task defined by f_{norm}.

Linear Correlation Coefficient Ranking based approaches are not suitable to evaluate the scenario f_{norm}, due to their indifference to the particular predicted and actual scores. The linear correlation coefficient r is the used alternative. It is defined as the covariance $Cov(X, Y)$, normalized by the product of the standard deviations $\sigma_X \sigma_Y$ of the predicted scores X and the actual scores Y: $r = \frac{Cov(X,Y)}{\sigma_X \sigma_Y}$. X and Y are perfectly positively linearly related if $r = 1$, while $r = 0$ indicates the lack of a linear relationship. As pointed out in the introduction,

r, although difficult to interpret, can be employed when the QPP method is not prone to overfitting. What is not sufficient however, is to compare the point estimate of r to a basline predictor and to view a higher value as proof of a better method. Instead, the confidence interval of r should be reported and it should be established (and that is usually neglected) if the difference in correlation is statistically significant[5]. In our evaluation, we employed the significance test proposed in [11]. It will be evident that due to the small query set sizes, a number of predictors show no significant difference and thus no conclusion can be drawn about the best. While we restrict ourselves to significance tests on r due to space constraints, the results are similar when evaluating Kendall's Tau and Spearman's Rho in this manner.

Drawbacks of r To exemplify the interpretation problem of r, scatter plots are presented in Figure 1 for high, moderate and low correlation along with the best linear fit. Each point represents a query with its corresponding Average Precision on the x-axis and prediction score on the y-axis. In case of 1(a) the *AvIDF* scores of queries 301-350 were plotted against a *tf.idf* based retrieval run with a low mean average precision (*map*) of 0.11. The high linear correlation of $r = 0.81$ highlights another possible issue: the correlation coefficient of a predictor can be improved by correlating the prediction scores with the "right" retrieval method instead of improving the predictor method itself. Figures 1(b) and 1(c) were generated from the predictor *AvIDF* and a Dirichlet smoothed ($\mu = 1000$) retrieval run. They show the difference between a medium and a low correlation. Intuitively, one might expect a noticeable difference in linearity between the cases of $r = 0.59$ and $r = 0.22$. In the two scatter plots however, the difference appears minor. For comparison, Kendall's Tau τ is also reported in Figure 1: its value is generally lower, but the trend is the same.



(a) $map = 0.11, r = 0.81, \tau = 0.48$ (b) $map = 0.23, r = 0.59, \tau = 0.32$ (c) $map = 0.18, r = 0.22, \tau = 0.18$

Fig. 1. Correlation examples.

Another drawback is the increase in correlation if multiple predictors are linearly combined (multiple linear regression). Independent of the quality of the predictors, r increases as more predictors are added to the model. An extreme example

[5] Currently, predictors are only tested for their significance against a correlation of 0.

is given in Figure 2 where the Average Precision scores of queries 451-550 were correlated with between 1 and 75 randomly generated predictors. At 75 predictors, $r > 0.9$. Figure 2 also contains the trend of the so-called *adjusted* r which takes the number of predictors in the model into account, but still $r_{adj} > 0.6$.

Extending the current methodology As our focus is on predicting the retrieval performance, the evaluation measure should reflect the emphasis on the *predictive* capabilities of a predictor and should provide us with a number that can be interpreted as how far the predictions deviate on average from the true value. Let $\hat{Y}$ be the predictions and Y be the true values, then the root mean squared error *RMSE* is given by $RMSE = \sqrt{\frac{1}{n}\sum_i (y_i - \hat{y_i})^2}$. Since $RMSE^2$ is the function minimized in linear regression, in effect, the pre-retrieval predictor with the highest linear correlation coefficient will have the lowest *RMSE*. This approach however mixes training and test data - what we are evaluating is the fit of the predictor with the training data, while we are interested in the evaluation of the predictor given novel queries. Ideally, we perform regression on the training data to determine the model parameters and then use the model to predict the query performance on separate test queries. Due to the very limited query set size, this is not feasible, and cross-validation is utilized instead: the query set is split into k partitions, the model is tuned on $k-1$ partitions and the kth partition functions as test set. This process is repeated for all k partitions and the overall *RMSE* is reported.

4 Penalized Regression Approaches

Modeling a continuous dependent variable $\mathbf{y}$, which in our case is a vector of Average Precision values, as a function of p independent predictor variables $\mathbf{x}_i$ is referred to as multiple *regression*. If we assume a linear relationship between the variables, we speak of multiple linear regression. Given the data $(\mathbf{x}^i, y_i)$, $i = 1, 2, .., n$ and $\mathbf{x}^i = (x_{i1}, ..., x_{ip})^T$, the parameters $\beta = (\beta_1, ..., \beta_p)$ of the model $\mathbf{y} = \mathbf{X}\beta + \epsilon$ are to be estimated. $\mathbf{X}$ is the $n \times p$ matrix of predictors and ϵ is the vector of errors, which are assumed to be normally distributed. The ordinary least squares (OLS) estimates of β are derived by minimizing the squared error of the residuals. Apart from overfitting, the lack of model interpretation is an issue. All predictors remain in the model and very similar predictors might occur with very different coefficients. If we have a large number of predictors, methods are preferred that perform automatic model selection, only introducing the most important subset of predictors into the model. While this has not yet been explored in the context of QPP, it received considerable attention among others in microarray data analysis [14] where good results were reported with penalized regression approaches. As the problems in both areas are similar (very small data sets, possibly many predictors) it appears sensible to attempt to apply those methods to query performance prediction. Due to space constraints we only briefly introduce the four variations tested, namely LARS-Traps, LARS-CV [6], bootstrapped LASSO (BOLASSO) [3] and the Elastic Net [21]. Penalized

regression approaches place penalties on the regression coefficients β to keep the coefficients small or exactly zero which essentially removes a number of predictors from the model. The least absolute shrinkage and selection operator (LASSO) [15] is such a method:

$$LASSO(\hat{\beta}) = \arg\min\Big\{\sum_{i=1}^{n}\big(y_i - \sum_{j=1}^{p}\beta_j x_{ij}\big)^2\Big\} \text{ subject to} \sum_{j=1}^{p}|\beta_j| \leq t. \quad (1)$$

The total weight of the coefficients is restricted by tuning parameter $t \geq 0$. If a number of predictors are very similar, LASSO tends to include only one of them in the final model whereas the Elastic Net [21] has a grouping effect such that highly correlated predictors acquire similar coefficients. It relies on a penalty combination of the squared and absolute sum of beta coefficients. LASSO is a special case of the later developed least angle regression (LARS) [6]. LARS determines the full regularization path: in each step, LAR selects the predictor that is most highly correlated with the residuals $\mathbf{y} - \hat{\mathbf{y}}$ of the current model, resulting in a $p \times p$ matrix of beta coefficients. In our experiments, such regularization paths were derived for LASSO, LARS and the Elastic Net. The question remains, which vector of beta coefficients from the matrix to choose as model coefficients. Several stopping criteria exist. Traps are randomly generated predictors that are added to the set of predictors. The regularization is stopped, as soon as the one of the random predictors is picked to enter the model. An alternative is cross-validation (CV): the beta coefficients are learned from $k-1$ partitions of the training data and the kth partition is used to calculate the error; the vector of beta coefficients with the smallest error is then chosen. A third possibility is the recently proposed bootstrapped LASSO [3], where a number of bootstrap samples are generated from the training data, the matrix of beta coefficients of LASSO are determined for each sample and in the end, only those predictors with non-zero coefficients in all bootstrap samples are utilized in the final model.

5 Experiments and Results

5.1 Experimental Setup

Data The adhoc retrieval task was evaluated on three collections and their respective title topics: TREC Volumes 4+5 (minus CR), WT10g and GOV2. The corpora were stemmed [9] and stopwords were removed. The basic statistics are shown in Figure 3. TREC Volumes 4+5 consists of news reports and is the smallest of the three corpora. WT10g was extracted from a crawl of the Web; it is rather noisy and contains numerous empty pages, pages with 1-2 terms only, copyright notices etc. The largest corpus is GOV2. It was derived from a crawl of the .gov domain and resembles somewhat an intranet structure. For the analysis of pre-retrieval predictors we fixed the retrieval approach to the language modeling framework with Dirichlet smoothing [18], with smoothing parameter $\mu = 1000$.

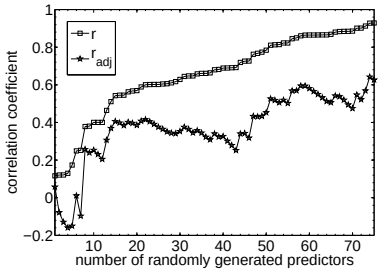


Fig. 2. Development of r and r_{adj} with increasing number of random predictors.

	TREC Vol. 4+5	WT10g	GOV2
#documents	528155	1692095	25199132
#unique terms	764376	7081712	32933168
av doc length	266.4	377.6	665.3
$r(cf, df)$	0.95	0.88	0.67
topics	301-450	451-550	701-850
av topic length	2.48	2.63	2.97

Fig. 3. Basic corpora statistics. $r(cf, df)$ is the linear correlation coefficient between collection term frequencies and document frequencies.

Predictor Settings All predictors described in Section 2 were evaluated. Most are parameter-free, only the cluster based predictors *AvQC* and *AvQCG* require a manually set sampling level. As it is infeasible to cluster all documents containing a particular term, a maximum of 10000 (TREC Vol. 4+5), 20000 (WT10g) and 50000 (GOV2) documents were sampled respectively. The parameter settings of the Elastic Net were taken from [21]. LARS-Traps was tested with 6 randomly generated traps while LARS-CV was set up with 10-fold CV. BOLASSO was used with 10 bootstrapped samples and each sample was cross-validated to retrieve the best beta coefficient vector. For evaluation purposes, the *RMSE* of all methods was determined by leave-one-out cross validation, where each query is once assigned as test set and the model is trained on all other queries. This setting is sensible due to the small query set size (maximum 150). To emphasize the cross-validation *RMSE* approach being different from r/CI established on the training set only, we write r_{train} and CI_{train}.

5.2 Results

Statistical Significance In Figure 4 the performance of all pre-retrieval predictors is given in terms of their linear correlation coefficient and the corresponding 95% confidence interval (*CI*). If the *CI* contains 0, the predictor is not significantly different from 0 correlation. Additionally, all predictors not significantly different [11] from the best performing predictor for each collection are underlined. While not shown, a similar analysis was performed for Kendall's Tau, whose results were comparable.

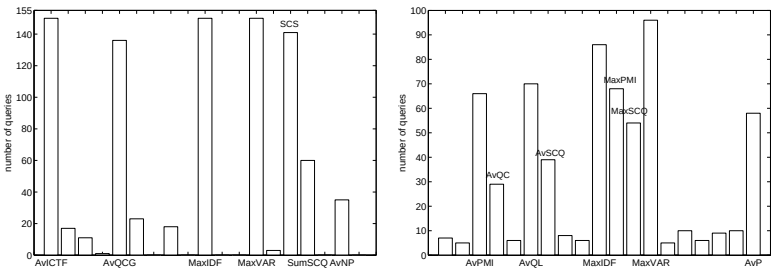
Root Mean Squared Error The *RMSE* of the single predictors are also presented in Figure 4. Since linear regression minimizes the mean squared error, the best predictors in terms of r also have the lowest *RMSE*. The penalized regression results are reported in Figure 5 along with r and *CI* and exemplary the predictors selected for LARS-Traps and LARS-CV are shown in histogram form in Figure 6. The bars indicate in how many of the n times the algorithm run, each predictor was selected to be in the model.

	TREC Vol. 4+5			WT10g			GOV2		
Predictor	r_{train}	CI_{train}	$RMSE$	r_{train}	CI_{train}	$RMSE$	r_{train}	CI_{train}	$RMSE$
AvICTF	0.49	[0.36, 0.60]	0.190	0.17	[-0.03, 0.35]	0.195	0.28	[0.13, 0.42]	0.184
AvIDF	0.52	[0.39, 0.62]	0.188	0.18	[-0.02, 0.37]	0.192	0.31	[0.16, 0.45]	0.182
AvPMI	0.35	[0.21, 0.48]	0.207	0.28	[0.09, 0.46]	0.191	0.28	[0.12, 0.42]	0.187
AvQC	0.46	[0.32, 0.58]	0.191	0.16	[-0.04, 0.35]	0.196	0.30	[0.14, 0.44]	0.184
AvQCG	0.33	[0.18, 0.47]	0.206	-0.02	[-0.22, 0.18]	0.201	0.05	[-0.11, 0.21]	0.194
AvQL	0.13	[-0.03, 0.29]	0.215	-0.14	[-0.32, 0.07]	0.197	0.01	[-0.15, 0.17]	0.194
AvSCQ	0.26	[0.10, 0.40]	0.210	0.31	[0.12, 0.48]	0.188	0.35	[0.20, 0.52]	0.180
AvVAR	0.51	[0.38, 0.62]	0.185	0.29	[0.10, 0.46]	0.188	0.39	[0.25, 0.52]	0.177
DevIDF	0.24	[0.08, 0.38]	0.212	0.23	[0.04, 0.41]	0.192	0.18	[0.02, 0.34]	0.189
MaxIDF	**0.53**	[0.41, 0.64]	0.186	0.29	[0.10, 0.46]	0.187	0.33	[0.18, 0.47]	0.181
MaxPMI	0.30	[0.14, 0.43]	0.210	0.27	[0.07, 0.44]	0.191	0.30	[0.15, 0.44]	0.186
MaxSCQ	0.34	[0.19, 0.47]	0.205	0.40	[0.22, 0.55]	**0.184**	0.40	[0.26, 0.53]	0.178
MaxVAR	0.51	[0.38, 0.62]	**0.182**	**0.41**	[0.23, 0.56]	**0.184**	**0.41**	[0.27, 0.54]	**0.176**
QS	0.41	[0.26, 0.53]	0.201	0.06	[-0.14, 0.26]	0.197	0.18	[0.02, 0.33]	0.189
SCS	0.48	[0.35, 0.59]	0.191	0.13	[-0.07, 0.32]	0.195	0.25	[0.09, 0.39]	0.186
SumSCQ	0.00	[-0.16, 0.16]	0.217	0.18	[-0.02, 0.37]	0.194	0.23	[0.08, 0.38]	0.187
SumVAR	0.30	[0.14, 0.44]	0.206	0.30	[0.11, 0.47]	0.189	0.34	[0.19, 0.47]	0.182
AvNP	-0.22	[-0.37, -0.06]	0.210	-0.11	[-0.30, 0.09]	0.198	-0.03	[-0.19, 0.13]	0.194
AvP	-0.12	[-0.28, 0.04]	0.214	-0.18	[-0.37, 0.02]	0.195	0.02	[-0.14, 0.18]	0.194

Fig. 4. Performance of pre-retrieval predictors given in r, the 95% confidence interval CI of r and the $RMSE$. In bold is the best performing predictor for each collection. According to the significance test in [11], all underlined predictors per column are not significantly different from the best performing predictor.

	TREC Vol. 4+5			WT10g			GOV2		
Predictor	r_{train}	CI_{train}	$RMSE$	r_{train}	CI_{train}	$RMSE$	r_{train}	CI_{train}	$RMSE$
OLS	**0.69**	[0.60, 0.77]	0.188	**0.64**	[0.51, 0.74]	0.208	**0.52**	[0.39, 0.63]	0.190
LARS-Traps	**0.59**	[0.47, 0.68]	**0.179**	**0.52**	[0.36, 0.65]	0.187	**0.44**	[0.30, 0.56]	0.178
LARS-CV	**0.68**	[0.59, 0.76]	0.183	**0.53**	[0.38, 0.66]	**0.178**	**0.46**	[0.33, 0.58]	0.184
BOLASSO	**0.59**	[0.47, 0.68]	**0.181**	**0.43**	[0.25, 0.58]	0.198	**0.43**	[0.28, 0.55]	0.180
Elastic Net	**0.69**	[0.60, 0.77]	0.182	**0.52**	[0.35, 0.65]	**0.182**	**0.46**	[0.32, 0.57]	0.178

Fig. 5. Results of the penalized regression approaches. In bold, improvements over the best single predictor per collection are shown. Notice how the r_{train} values provide a misleading indication of the system's actual performance.



(a) TREC Vol. 4+5, LARS-Traps (b) WT10g, LARS-CV

Fig. 6. Predictor selection.

6 Discussion

The results show that the predictor performance is collection dependent, varying according to the quality of the corpus documents. It is evident, that the single predictors perform better on the high quality news article collection compared to the web collections. The specificity based predictors achieve the highest correlation at TREC Vol. 4+5, whereas the ranking sensitivity based predictors appear the most stable, performing similarly across all three corpora. The CIs of r are wide due to the small number of queries and the lower r, the wider the interval. For $r = 0.31$ with $CI = [0.12, 0.48]$ for instance, one cannot say more than with 95% confidence that r is below 0.5. Significance testing reveals that a variety of single predictors are not significantly different from the predictor with the highest r value. This result indicates, that any one of the top predictors could be used. *MaxIDF* and the more computationally intensive *MaxVAR*, consistently provided the best performance.
While the correlation coefficient r suggests that the combined methods perform better than the single predictors, when we examine the results of the stronger *RMSE* based evaluation methodology, a different picture presents itself. Although the penalized regression approaches have a lower error than the OLS baseline as expected, the decrease in error compared to the single predictors is smaller than one might expect. In fact, on the GOV2 collection the error increased. In Figure 6(a) the histogram shows which predictors contribute the most to the combined estimate. Notably, in most instances the same five predictors are used in the final models. The remaining predictors appear to fail to capture any more of the variance within the data and remain unused. Such a behaviour is desired, however, the performance is less than expected in terms of *RMSE*. This is due to two reasons, (a) the quality of some predictors is poor and might not be better than random and (b) as evident from the scatter plots (Figure 3), the training data is overrepresented at the lower end of the Average Precision scores (0.0-0.2) while very few queries exist in the middle and high range. The problems caused by poor predictors is further exemplified in Figure 6(b) where apart from *MaxVAR* a large variety of predictors are added to the model.

7 Conclusions

In this paper we presented a taxonomy of pre-retrieval predictors and clarified the different tasks that lead to different evaluation measures. We focused on f_{norm} and its currently employed measure, the linear correlation coefficient. We showed that it is difficult to interpret, without a significance test no conclusion can be drawn about the relative improvement of one method over another and it is not usable when combining predictors. Reporting the cross-validated *RMSE* as a measure of the prediction accuracy gives a more reliable and interpretable indicator of a method's quality. Finally, we performed first experiments on combining predictors in a principled way through penalized regression which has the advantages of sparseness and interpretability. We showed that under the previous

evaluation methodology the combination methods would be considered better in terms of r, though they are in fact comparable to the best single predictors. In future work we will address the problem of limited training data by employing simulated queries to create more training data, investigate the use of non-linear methods for combination and employ regression trees.

References

1. *Rank Correlation Methods*. New York: Hafner Publishing Co., 1955.
2. *WordNet - An Electronic Lexical Database*. The MIT Press, 1998.
3. F. R. Bach. Bolasso: Model consistent lasso estimation through the bootstrap. In *ICML*, 2008.
4. S. Banerjee and T. Pedersen. Extended gloss overlaps as a measure of semantic relatedness. In *IJCAI-03*, pages 805–810, 2003.
5. S. Cronen-Townsend, Y. Zhou, and W. B. Croft. Predicting query performance. In *SIGIR '02*, pages 299–306, 2002.
6. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *Ann. Statist.*, 32(2):407–499, 2004.
7. B. He and I. Ounis. Inferring query performance using pre-retrieval predictors. In *SPIRE'04*, pages 43–54, 2004.
8. J. He, M. Larson, and M. de Rijke. Using coherence-based measures to predict query difficulty. In *ECIR'08*, pages 689–694, 2008.
9. R. Krovetz. Viewing morphology as an inference process. In *SIGIR '93*, pages 191–202, 1993.
10. C. Macdonald, B. He, and I. Ounis. Predicting query performance in intranet search. In *SIGIR'05 Query Prediction Workshop*, 2005.
11. X. Meng, R. Rosenthal, and D. Rubin. Comparing correlated correlation coefficients. *Psych. Bull.*, 111:172–175, 1992.
12. J. Mothe and L. Tanguy. Linguistic features to predict query difficulty - a case study on previous trec campaigns. In *SIGIR'05 Query Prediction Workshop*, 2005.
13. F. Scholer, H. Williams, and A. Turpin. Query association surrogates for web search. *Journal of the American Society for Information Science and Technology*, 55(7):637–650, 2004.
14. M. R. Segal, K. D. Dahlquist, and B. R. Conklin. Regression approaches for microarray data analysis. *J. Comput. Biol.*, 10(6):961–980, 2003.
15. R. Tibshirani. Regression shrinkage and selection via the lasso. *J. R. Statist. Soc. B*, 58(1):267–288, 1996.
16. V. Vinay, I. J. Cox, N. Milic-Frayling, and K. Wood. On ranking the effectiveness of searches. In *SIGIR '06*, pages 398–404, 2006.
17. E. Voorhees. Overview of the trec 2003 robust retrieval track. In *Proceedings of the Twelfth Text REtrieval Conference*, 2003.
18. C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *SIGIR '01*, pages 334–342, 2001.
19. Y. Zhao, F. Scholer, and Y. Tsegay. Effective pre-retrieval query performance prediction using similarity and variability evidence. In *ECIR'08*, pages 52–64, 2008.
20. Y. Zhou and W. B. Croft. Query performance prediction in web search environments. In *SIGIR '07*, pages 543–550, 2007.
21. H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *J. R. Statist. Soc. B*, 67(2):301–320, 2005.