

# Predicting the Effectiveness of Queries and Retrieval Systems

CLAUDIA HAUFF

# Chapter 2

## Pre-Retrieval Predictors

### 2.1 Introduction

Pre-retrieval prediction algorithms predict the effectiveness of a query before the retrieval stage is reached and are, thus, independent of the ranked list of results; essentially, they are search-independent. Such methods base their predictions solely on query terms, the collection statistics and possibly an external source such as WordNet [57], which provides information on the query terms' semantic relationships. Since pre-retrieval predictors rely on information that is available at indexing time, they can be calculated more efficiently than methods relying on the result list, causing less overhead to the search system. In this chapter we provide a comprehensive overview of pre-retrieval query performance prediction methods.

Specifically, this chapter contains the following contributions:

- the introduction of a predictor taxonomy and a clarification of evaluation goals and evaluation measures,
- an analytical and empirical evaluation of a wide range of prediction methods over a range of corpora and retrieval approaches, and,
- an investigation into the utility of combining different prediction methods in a principled way.

The organization of the chapter is set up accordingly. First, in Section 2.2 we present our predictor taxonomy, then in Section 2.3 we discuss the goals of query effectiveness prediction and subsequently lay out what evaluations exist and when they are applicable. A brief overview of the notation and the data sets used in the evaluations (Sections 2.4 and 2.5) follows. In Sections 2.6, 2.7, 2.8 and 2.9 we cover the four different classes of predictor heuristics. While these sections give a very detailed view on each method, in Section 2.10 we discuss the results of an evaluation that has so far been neglected in query performance prediction: the evaluation whether two predictors perform differently from each other in a statistically significant way. How diverse retrieval approaches influence the quality of various prediction methods is evaluated in Section 2.11. A final matter of investigation is the utility of combining prediction methods in a principled way, which is described in Section 2.12. The conclusions in Section 2.13 round off the chapter.

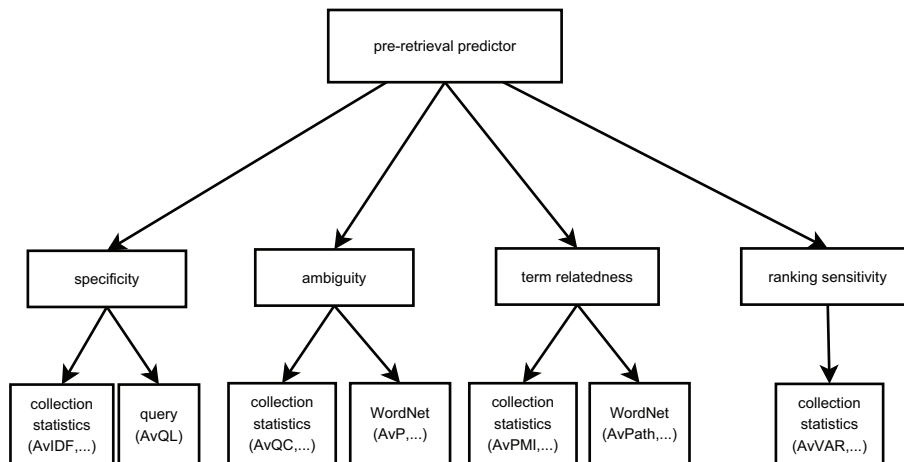


Figure 2.1: Categories of pre-retrieval predictors.

## 2.2 A Pre-Retrieval Predictor Taxonomy

In general, pre-retrieval predictors can be divided into four different groups according to the heuristics they exploit (Figure 2.1). First, *specificity* based predictors predict a query to perform better with increased specificity. How the specificity is determined further divides these predictors into collection statistics based and query based predictors.

Other predictors exploit the query terms' *ambiguity* to predict the query's quality; in those cases, high ambiguity is likely to result in poor performance. In such a scheme, if a term always appears in the same or similar contexts, the term is considered to be unambiguous. However, if the term appears in many different contexts it is considered to be ambiguous. For instance, consider that the term "tennis" will mainly appear in the context of sports and will rarely be mentioned in documents discussing finances or politics. The term "field", however, is more ambiguous and can easily occur in sports articles, agriculture articles or even politics (e.g. "field of Democratic candidates"). Intuitively, ambiguity is somewhat related to specificity, as an ambiguous term can have a high document frequency, but there are exceptions - consider that the term "tennis" might not be specific in a corpus containing many sports-related documents, but it is unambiguous and while specificity based predictors would predict it to be a poor query, ambiguity based predictors would not. The ambiguity of a term may be derived from collection statistics, additionally it can also be determined by relying on an external source such as WordNet.

The drawback of predictors in the first two categories (specificity and ambiguity) stems from their lack of consideration of the relationship between terms. To illustrate this point, consider that the query "political field" is actually unambiguous due to the relationship between the two terms, but an ambiguity based predictor is likely to predict a poor effectiveness, since "field" can appear in many contexts. Similarly for a specificity based predictor, the term "field" will likely occur often in a general corpus. To offset this weakness, a third category of predictors makes use of *term relatedness* in an attempt to exploit the relationship between query terms.

Specifically, if there is a strong relationship between terms, the query is predicted to be of good quality.

Finally, the *ranking sensitivity* can also be utilized as source of information for a query’s effectiveness. In such a case, a query is predicted to be ineffective, if the documents containing the query terms appear similar to each other, making them indistinguishable for a retrieval system and thus difficult to rank. In contrast to post-retrieval methods, which work directly on the rankings produced by the retrieval algorithms, these predictors attempt to predict how easy it is to return a stable ranking. Moreover, they rely exclusively on collection statistics, and more specifically the distribution of query terms within the corpus.

## 2.3 Evaluation Framework

Query effectiveness prediction methods are usually evaluated by reporting the correlation they achieve with the ground truth, which is the effectiveness of queries derived for a retrieval approach with the help of relevance judgments. The commonly reported correlations coefficients are Kendall’s Tau  $\tau$  [84], Spearman’s Rho  $\rho$ , and the linear correlation coefficient  $r$  (also known as Pearson’s  $r$ ). In general, the choice of correlation coefficient should depend on the goals of the prediction algorithm. As often prediction algorithms are evaluated but not applied in practice, a mix of correlation coefficients is usually reported as will become evident in Chapter 3 (in particular in Table 3.1).

### 2.3.1 Evaluation Goals

Evaluation goals can be for instance the determination whether a query can be answered by a corpus or an estimation of the retrieval effectiveness of a query. We now present three categories of evaluation goals which apply both to pre- and post-retrieval algorithms.

#### Query Difficulty

The query difficulty criterion can be defined as follows: given a query  $\mathbf{q}$ , a corpus of documents  $C$ , external knowledge sources  $E$  and a ranking function  $R$  (which returns a ranked list of documents), we can estimate whether  $\mathbf{q}$  is difficult as follows:

$$f_{diff}(\mathbf{q}, C, E, R) \rightarrow \{0, 1\}. \quad (2.1)$$

Here,  $f_{diff} = 0$  is an indicator of the class of difficult queries which exhibit unsatisfactory retrieval effectiveness and  $f_{diff} = 1$  represents the class of well performing queries. When  $R = \emptyset$  we are dealing with pre-retrieval prediction methods. A number of algorithms involve external sources  $E$  such as Wikipedia or WordNet. The majority of methods however, rely on  $C$  and  $R$  only. Evaluation measures that are

in particular applicable to  $f_{diff}$  emphasize the correct identification of the worst performing queries and largely ignore the particular performance ranking and the best performing queries [145, 151].

### Query Performance

Determining whether a query will perform well or poorly is not always sufficient. Consider for example a number of alternative query formulations for an information need. In order to select the best performing query, a more general approach is needed; such an approach is *query performance prediction*. Using the notation above, we express this as follows:

$$f_{perf}(\mathbf{q}, C, E, R) \rightarrow \mathbb{R} \quad (2.2)$$

The query with the largest score according to  $f_{perf}$  is deemed to be the best formulation of the information need. In this scenario, we are not interested in the particular scores, but in correctly ranking the queries according to their predicted effectiveness. In such a setup, evaluating the agreement between the predicted query ranking and the actual query effectiveness ranking is a sound evaluation strategy. The alignment of these two rankings is usually reported in terms of rank correlation coefficients such as Kendall's  $\tau$  and Spearman's  $\rho$ .

### Normalized Query Performance

In a number of instances, absolute estimation scores as returned by  $f_{perf}$  cannot be utilized to locate the best query from a pool of queries. Consider a query being submitted to a number of collections and the ranked list that is estimated to best fit the query is to be selected, or alternatively the ranked lists are to be merged with weights according to the estimated query quality. Absolute scores as given by  $f_{perf}$  will fail, as they usually depend on collection statistics and are, thus, not comparable across corpora. The evaluation should thus emphasize, how well the algorithms estimate the effectiveness of a query according to a particular effectiveness measure such as average precision. Again, using the usual notation:

$$f_{norm}(\mathbf{q}, C, E, R) \rightarrow [0, 1]. \quad (2.3)$$

By estimating a *normalized* score, scores can be compared across different collections. The standard evaluation measure in this setting is the linear correlation coefficient  $r$ .

## 2.3.2 Evaluation Measures

As described in the previous section, different evaluation methodologies are applicable to different application scenarios. The standard correlation based approach to evaluation is as follows. Let  $Q$  be the set of queries  $\{\mathbf{q}_i\}^i$  and let  $R_{\mathbf{q}_i}$  be the ranked list returned by the ranking function  $R$  for  $\mathbf{q}_i$ . For each  $\mathbf{q}_i \in Q$ , the predicted score  $s_i$  is obtained from a given predictor; additionally the retrieval effectiveness of  $R$  is

determined (based on the relevance judgments). Commonly, the average precision  $ap_i$  of  $R_{q_i}$  is calculated as ground truth effectiveness. Then, given all pairs  $(s_i, ap_i)$ , the correlation coefficient is determined.

## Ranking Based Approaches

Rank correlations make no assumptions about the type of relationship between the two lists of scores (predictor scores and retrieval effectiveness scores). Both score lists are converted to lists of ranks where the highest score is assigned rank 1 and so on. Then, the correlation of the ranks is measured. In this case, the ranks give an indication of each query's effectiveness relative to the other queries in the list but no quantitative prediction is made about the retrieval score of the query.

The TREC Robust Retrieval track [151, 152], where query effectiveness prediction was first proposed as part of the adhoc retrieval task, aimed at distinguishing the poorly performing queries from the successful ones. The participants were asked to rank the given set of queries according to their estimated performance. As measure of agreement between the predicted ranking and the actual ranking, Kendall's  $\tau$  was proposed.

A common approach to comparing two predictors is to compare their point estimates and to view a higher correlation coefficient as proof of a better predictor method. However, to be able to say with confidence that one predictor outperforms another, it is necessary to perform a test of statistical significance of the difference between the two [39]. Additionally, we can give an indication of how confident we are in the result by providing the confidence interval (CI) of the correlation coefficient. Currently, predictors are only tested for their significance against a correlation of zero.

While Kendall's  $\tau$  is suitable for the setup given by  $f_{pref}$ , it is sensitive to all differences in ranking. If we are only interested in identifying the poorly performing queries ( $f_{diff}$ ), ranking differences at the top of the ranking are of no importance and can be ignored. The *area between the MAP curves*, proposed by Voorhees [151], is an evaluation measure for this scenario. The mean average precision (MAP) is computed over the best performing  $b$  queries and  $b$  ranges from the full query set to successively fewer queries, leading to a MAP curve. Two such MAP curves are generated: one based on the actual ranking of queries according to retrieval effectiveness and one based on the predicted ranking of queries. If the predicted ranking conforms to the actual ranking, the two curves are identical and the area between the curves is zero. The more the predicted ranking deviates from the actual ranking, the more the two curves will diverge and, thus, the larger the area between them. It follows, that the larger the area between the curves, the worse the accuracy of the predictor. A simpler evaluation measure that is also geared towards query difficulty, was proposed by Vinay et al. [145]. Here, the bottom ranked 10% or 20% of predicted and actual queries are compared and the overlap is computed; the larger the overlap, the better the predictor.

## Linear Correlation Coefficient

Ranking based approaches are not suitable to evaluate the scenario  $f_{norm}$ , as they disregard differences between the particular predicted and actual scores. In such a case, the linear correlation coefficient  $r$  can be used instead. This coefficient is defined as the covariance, normalized by the product of the standard deviations of the predicted scores and the actual scores.

The value of  $r^2$  is known as the *coefficient of determination*. A number of tests for the significance of difference between overlapping correlations have been proposed in the literature [77, 103, 158]. In our evaluation, we employed the test proposed by Meng et al. [103].

In the case of multiple linear regression,  $r$  increases due to the increase in regressors. To account for that, the value of the *adjusted*  $r^2$  can be reported, which takes the number  $p$  of regressors into account ( $n$  is the sample size):

$$r_{adj}^2 = 1 - (1 - r^2) \frac{n - 1}{n - p - 1}. \quad (2.4)$$

## Limitations of Correlation Coefficients

Correlation coefficients compress a considerable amount of information into a single number, which can lead to problems of interpretation. To illustrate this point, consider the cases depicted in Figure 2.2 for the linear correlation coefficient  $r$  and Kendall's  $\tau$ . Each point represents a query with its corresponding retrieval effectiveness value (given in average precision) on the x-axis and its predicted score on the y-axis. The three plots are examples of high, moderate and low correlation coefficients; for the sake of  $r$ , the best linear fit is also shown. Further, the MAP as average measure of retrieval effectiveness is provided as well. These plots are derived from data that reflects existing predictors and retrieval approaches. In the case of Figure 2.2a, the predictor scores are plotted against a very basic retrieval approach with a low retrieval effectiveness (MAP of 0.11). The high correlations of  $r = 0.81$  and  $\tau = 0.48$  respectively highlight a possible problem: the correlation coefficient of a predictor can be improved by correlating the prediction scores with the “right” retrieval method instead of improving the quality of the prediction method itself.

To aid understanding, consider Figures 2.2b and 2.2c, which were generated from the same predictor for different query sets and a better performing retrieval approach. They show the difference between a medium and a low correlation. Note that, in general, the value of Kendall's  $\tau$  is lower than  $r$ , but the trend is similar.

In Section 2.12, we evaluate the utility of combining predictors in a principled way. The evaluation is performed according to  $f_{norm}$ , which is usually reported in the literature in terms of  $r$ . However, when combining predictors, a drawback of  $r$  is the increase in correlation if multiple predictors are linearly combined. Independent of the quality of the predictors,  $r$  increases as more predictors are added to the model. An extreme example of this, is shown in Figure 2.3 where the average precision scores of a query set were correlated with randomly generated predictors numbering between 1 and 75. Note that at 75 predictors,  $r > 0.9$ . Figure 2.3 also contains

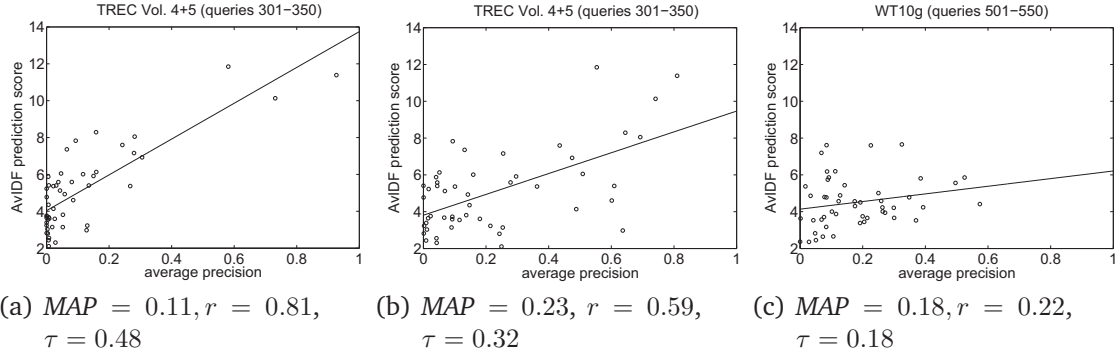


Figure 2.2: Scatter plots of retrieval effectiveness scores versus predicted scores.

the trend of  $r_{adj}$ , which takes the number of predictors in the model into account, but despite this adaptation we observe  $r_{adj} > 0.6$  when 75 random predictors are combined.

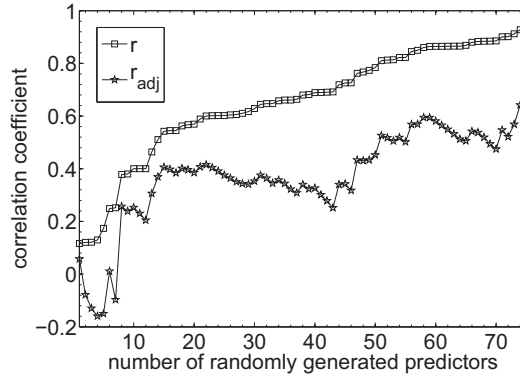


Figure 2.3: Development of  $r$  and  $r_{adj}$  with increasing number of random predictors.

## 2.4 Notation

We now briefly present definitions and notations as used for the remainder of this chapter. A query  $\mathbf{q} = \{q_1, q_2, \dots, q_m\}$  is composed of query terms  $q_i$  and has length  $|\mathbf{q}| = m$ . A term  $t_i$  occurs  $tf(t_i, d_j)$  times in document  $d_j$ . Further, a term  $t_i$  occurs  $tf(t_i) = \sum_j tf(t_i, d_j)$  times in the collection and in  $df(t_i)$  documents. The document length  $|d_j|$  is equal to the number of terms in the document. The total number of terms in the collection is denoted by *termcount* and *doccoun*t marks the total number of documents.  $N_{\mathbf{q}}$  is the set of all documents containing at least one of the query terms in  $\mathbf{q}$ .

The maximum likelihood estimate of term  $t_i$  in document  $d_j$  is given by

$$P_{ml}(t_i|d_j) = \frac{tf(t_i, d_j)}{|d_j|}. \quad (2.5)$$

The probability of term  $t_i$  occurring in the collection is  $P_{ml}(t_i) = \frac{tf(t_i)}{termcount}$ . Finally,  $P_{ml}(t_i, t_j)$  is the maximum likelihood probability of  $t_i$  and  $t_j$  occurring in the same document.

## 2.5 Materials and Methods

The evaluations of the methods outlined in the current and the following chapters are performed on a range of query sets and corpora. In this section, we briefly describe the corpora, the query sets and the retrieval approaches utilized. A more comprehensive overview of the data sets can be found in Appendix B.

### 2.5.1 Test Corpora

To perform the experiments, the adhoc retrieval task is evaluated on three different TREC corpora, namely, TREC Volumes 4 and 5 minus the Congressional Records [148] (TREC Vol. 4+5), WT10g [132] and GOV2 [38]. The corpora differ in size as well as content. TREC Vol. 4+5 is the smallest, containing newswire articles, WT10g is derived from a crawl of the Web and GOV2, the largest corpus with more than 25 million documents, was created from a crawl of the .gov domain. The corpora were stemmed with the Krovetz stemmer [90] and stopwords were removed<sup>1</sup>. All experiments in this thesis are performed with the Lemur Toolkit for Language Modeling and Information Retrieval<sup>2</sup>, version 4.3.2.

The queries are derived from the TREC title topics of the adhoc tasks, available for each corpus. We focus on title topics as we consider them to be more realistic than the longer description and narrative components of a TREC topic. Please note again, that we distinguish the concepts of *topic* and *query*: whereas a topic is a textual expression of an information need, we consider a query to be the string of characters that is submitted to the retrieval system. In our experiments we turn a TREC title topic into a query by removing stopwords and applying the Krovetz stemmer.

Table 2.1 contains the list of query sets under consideration, the corpus they belong to and the average number of query terms. In query set 451-500, we manually identified and corrected three spelling errors. Our focus is on investigating query effectiveness predictors and we assume the ideal case of error-free queries. In practical applications, spelling error correction would be a preprocessing step.

### 2.5.2 Retrieval Approaches

The goal of prediction algorithms is to predict the (relative) retrieval effectiveness of a query as well as possible. Since there are many retrieval approaches with various degrees of retrieval effectiveness, an immediate concern lies in determining

---

<sup>1</sup>stopword list: [http://ir.dcs.gla.ac.uk/resources/linguistic\\_utils/stop\\_words](http://ir.dcs.gla.ac.uk/resources/linguistic_utils/stop_words)

<sup>2</sup><http://www.lemurproject.org/>

Corpus	Queries	Av. Query Length
TREC Vol. 4+5	301-350	2.54
	351-400	2.50
	401-450	2.40
WT10g	451-500	2.43
	501-550	2.84
GOV2	701-750	3.10
	751-800	2.94
	801-850	2.86

Table 2.1: Overview of query sets.

for which retrieval approach and for which effectiveness measure the prediction method should be evaluated. In this chapter, we rely on average precision as the effectiveness measure as it is most widely used in the literature and available for all retrieval experiments we perform.

We chose to address the question of which retrieval approach to utilize in two ways. First, we investigate three common retrieval approaches, namely Language Modeling with Dirichlet Smoothing [170], Okapi [125] with its default parameter settings, and TF.IDF [13], the most basic retrieval approach based on term and document frequencies. Although this setup allows us to investigate the influence of a change in parameter setting for one particular retrieval approach, the results cannot be further generalized. In order to gain an understanding of predictor performances over a wider variety of retrieval approaches, we also rely on the retrieval runs submitted to TREC for each title topic set and their achieved retrieval effectiveness as ground truth.

Table 2.2 lists the retrieval effectiveness of the three retrieval approaches in MAP over all query sets. The level of smoothing in the Language Modeling approach is varied between  $\mu = \{100, 500, 1000, 1500, 2000, 2500\}$ . Larger values of  $\mu$  show no further improvements in retrieval effectiveness (see Appendix B, Figure B.2). As expected, TF.IDF performs poorly, consistently degrading in performance as the collection size increases. Notably, while it reaches a MAP up to 0.11 on the query sets of TREC Vol. 4+5, for the query sets of the GOV2 collection, the MAP degrades to 0.04 at best. In contrast, Okapi outperforms the Language Modeling approach with  $\mu = 100$  for all but one query set (401-450). In all other settings of  $\mu$ , Okapi performs (slightly) worse. The highest effectiveness in the Language Modeling approach is achieved for a smoothing level  $\mu$  between 500 and 2000, depending on the individual query set.

## 2.6 Specificity

Query performance predictors in this category estimate the effectiveness of a query by the query terms' specificity. Consequently, a query consisting of common (collection) terms is deemed hard to answer as the retrieval algorithm is unable to

Corpus	Queries	TFIDF	Okapi	Language Modeling with Dirichlet Smoothing					
				$\mu = 100$	$\mu = 500$	$\mu = 1000$	$\mu = 1500$	$\mu = 2000$	$\mu = 2500$
TREC Vol. 4+5	301-350	0.109	0.218	0.216	<b>0.227</b>	0.226	0.224	0.220	0.218
	351-400	0.073	0.176	0.169	0.182	0.187	0.189	<b>0.190</b>	0.189
	401-450	0.088	0.223	0.229	0.242	<b>0.245</b>	0.244	0.241	0.239
WT10g	451-500	0.055	0.183	0.154	0.195	<b>0.207</b>	0.206	0.201	0.203
	501-550	0.061	0.163	0.137	0.168	0.180	0.185	<b>0.189</b>	0.189
GOV2	701-750	0.029	0.230	0.212	0.262	<b>0.269</b>	0.266	0.261	0.256
	751-800	0.036	0.296	0.279	0.317	<b>0.324</b>	0.324	0.321	0.318
	801-850	0.023	0.250	0.247	0.293	<b>0.297</b>	0.292	0.284	0.275

Table 2.2: Overview of mean average precision over different retrieval approaches. Shown in bold is the most effective retrieval approach for each query set.

distinguish relevant and non-relevant documents based on term frequencies. The following is a list of predictors in the literature that exploit the specificity heuristic:

- Averaged Query Length (*AvQL*) [111],
- Averaged Inverse Document Frequency (*AvIDF*) [45],
- Maximum Inverse Document Frequency (*MaxIDF*) [128],
- Standard Deviation of IDF (*DevIDF*) [71],
- Averaged Inverse Collection Term Frequency (*AvICTF*) [71],
- Simplified Clarity Score (*SCS*) [71],
- Summed Collection Query Similarity (*SumSCQ*) [174],
- Averaged Collection Query Similarity *AvSCQ* [174],
- Maximum Collection Query Similarity *MaxSCQ* [174], and,
- Query Scope (*QS*) [71].

### 2.6.1 Query Based Specificity

The specificity of a query can be estimated to some extent without considering any other sources apart from the query itself. The average number *AvQL* of characters in the query terms is such a predictor: the higher the average length of a query, the more specific the query is assumed to be. For instance, TREC title topic 348 “Agoraphobia” has an average query length of 11, whilst TREC title topic 344 “Abuses of E-Mail” has an average length of  $AvQL = 4.67$ . Hence, “Agoraphobia” is considered to be more specific and therefore would be predicted to perform better than “Abuses of E-Mail”.

Intuitively, making a prediction without taking the collection into account will often go wrong. Consider, for instance, that “BM25”, which would be a very specific term in a corpus of newswire articles, contains few characters and hence, is erroneously considered to be non-specific according to the previous scheme. The success of predictors of this type also depends on the language of the collection. Text collections in languages that allow compounding such as Dutch and German might benefit more from predictors of this type than corpora consisting of English documents.

An alternative interpretation of query length is to consider the number of query terms in the search request as an indicator of specificity. We do not cover this interpretation here, as TREC title topics have very little variation in the number of terms, while TREC description and narrative topics on the other hand, often do not resemble realistic search requests. We note though, that Phan et al. [119] performed a user study where participants were asked to judge search requests of differing length, on a four point scale according to how narrow or broad they judge the underlying information need to be. A significant correlation was found between the number of terms in the search requests and the information need’s specificity.

## 2.6.2 Collection Based Specificity

The specificity of a term  $q_i$  can be approximated by either the document frequency  $df(q_i)$  or the term frequency  $tf(q_i)$ . Both measures are closely related as a term that occurs in many documents can be expected to have a high term frequency in the collection. The opposite is also normally true: when a term occurs in very few documents then its term frequency will be low, if we assume that all documents in the collection are reasonable and no corner cases exist.

The most basic predictor in this context is *AvIDF* which determines the specificity of a query by relying on the average of the inverse document frequency (*idf*) of the query terms:

$$\begin{aligned} AvIDF &= \frac{1}{m} \sum_{i=1}^m \left[ \log \frac{doccount}{df(q_i)} \right] \\ &= \frac{1}{m} \sum_{i=1}^m [\log(doccount) - \log(df(q_i))] \end{aligned} \quad (2.6)$$

$$= \log(doccount) - \frac{1}{m} \log \left[ \prod_{i=1}^m df(q_i) \right] \quad (2.7)$$

*MaxIDF* is the maximum *idf* value over all query terms. As an alternative metric, instead of averaging or maximizing the *idf* values of all query terms, the predictor *DevIDF* relies on the standard deviation of the *idf* values:

$$DevIDF = \sqrt{\frac{1}{m} \sum_{i=1}^m \left( \log \frac{doccount}{df(q_i)} - AvIDF \right)^2} \quad (2.8)$$

Note that a query with a high *DevIDF* score has at least one specific term and one general term, otherwise the standard deviation would be small. A shortcoming of this predictor lies in the fact that single term queries or queries containing only specific terms are assigned a score of 0 and a low prediction score respectively. Thus, *DevIDF* can be expected to perform worse as predictor, on average, than *AvIDF* or *MaxIDF*.

In previous work [71], INQUERY’s *idf* formulation has been used in the predictors. In contrast to *AvIDF*, this approach normalizes the values to the  $[0, 1]$  interval.

Since such normalization makes no difference to the predictor performance it is not considered here. Among the pre-retrieval predictors proposed by He and Ounis [71] are the *AvICTF* and *SCS* predictors. *AvICTF* is defined as follows:

$$\begin{aligned}
 AvICTF &= \frac{\log_2 \prod_{i=1}^m \left[ \frac{termcount}{tf(q_i)} \right]}{m} \\
 &= \frac{1}{m} \sum_{i=1}^m \log_2 \left[ \frac{termcount}{tf(q_i)} \right] \\
 &= \frac{1}{m} \sum_{i=1}^m [\log_2(termcount) - \log_2(tf(q_i))] \tag{2.9}
 \end{aligned}$$

When comparing Equations 2.6 and 2.9, the similarity between *AvICTF* and *AvIDF* becomes clear; instead of document frequencies, *AvICTF* relies on term frequencies.

The Simplified Clarity Score is, as the name implies, a simplification of the post-retrieval method *Clarity Score* which will be introduced in detail in Chapter 3. Instead of applying Clarity Score to the ranked list of results however, it is applied to the query itself, as follows:

$$\begin{aligned}
 SCS &= \sum_{i=1}^m P_{ml}(q_i|\mathbf{q}) \log_2 \frac{P_{ml}(q_i|\mathbf{q})}{P(q_i)} \\
 &\approx \sum_{i=1}^m \frac{1}{m} \log_2 \frac{\frac{1}{m}}{\frac{tf(q_i)}{termcount}} \\
 &\approx \log_2 \frac{1}{m} + \frac{1}{m} \sum_{i=1}^m [\log_2(termcount) - \log_2(tf(q_i))]. \tag{2.10}
 \end{aligned}$$

$P_{ml}(q_i|\mathbf{q})$  is the maximum likelihood estimate of  $q_i$  occurring in query  $\mathbf{q}$ . If we assume that each query term occurs exactly once in a query, then  $P_{ml}(q_i|\mathbf{q}) = \frac{1}{m}$  and  $SCS = \log_2 \frac{1}{m} + AvICTF$  (consider the similarity of Equations 2.9 and 2.10).

Importantly, if two queries have the same *AvICTF* score, *SCS* will give the query containing fewer query terms a higher score. The assumption of each term occurring only once in the query is a reasonable one, when one considers short queries such as those derived from TREC title topics. In the case of short queries, we can expect that the *SCS* and *AvICTF* scores for a set of queries will have a correlation close to 1, as the query length does not vary significantly. Longer queries such as those derived from TREC description topics that often include repetitions of terms will result in a larger margin.

Combining the collection term frequency and inverse document frequency was proposed by Zhao et al. [174]. The collection query similarity summed over all query terms is defined as:

$$SumSCQ = \sum_{i=1}^m (1 + \ln(cf(q_i))) \times \ln \left( 1 + \frac{doccount}{df(q_i)} \right). \tag{2.11}$$

*AvSCQ* is the average similarity over all query terms:  $AvSCQ = \frac{1}{m} \times SumSCQ$ , whereas the maximum query collection similarity *MaxSCQ* relies on the maximum collection query similarity score over all query terms. The authors argue that a query, which is similar to the collection as a whole is easier to retrieve documents for, since the similarity is an indicator of whether documents answering the information need are contained in the collection. As the score increases with increased collection term frequency and increased inverse document frequency, terms that appear in few documents many times are favored. Those terms can be seen as highly specific, as they occur in relatively few documents, while at the same time they occur often enough to be important to the query.

Query Scope is a measure that makes use of the document frequencies. In this instance, the number of documents containing at least one of the query terms is used as an indicator of query quality; the more documents contained in this set, the lower the predicted effectiveness of the query:

$$QS = -\log \frac{N_q}{doccount}. \quad (2.12)$$

Finally, we observe that for queries consisting of a single term, the predictors *QS*, *MaxIDF* and *AvIDF* will return exactly the same score.

### 2.6.3 Experimental Evaluation

The evaluation of the introduced prediction methods is performed in two steps. First, to support the mathematical derivation, we present the correlations, as given by Kendall's  $\tau$ , between the different predictors. A high correlation coefficient indicates a strong relationship. Then, we evaluate the predictors according to their ability to predict the performance of different query sets across different corpora and retrieval approaches. This evaluation is presented in terms of Kendall's  $\tau$  and the linear correlation coefficient.

#### Predictor-Predictor Correlations

The correlations between the predictor scores are shown in Table 2.3 aggregated over the query sets of TREC Vol. 4+5 and over the query sets of GOV2. Let us first consider the results over the queries 301-450. The three predictors *AvIDF*, *SCS* and *AvICTF* are highly correlated, with a minimum  $\tau = 0.88$  (the same evaluation with the linear correlation coefficient yields  $r = 0.98$ ). The predictors *QS* and *MaxIDF* can also be considered in this group to some extent as they correlate with all three predictors with  $\tau \geq 0.65$  ( $r \geq 0.75$ ). Most predictors have a moderate to strong relationship to each other. Only *AvQL*, *DevIDF* and *SumSCQ* consistently behave differently.

The similarity between the prediction methods is different for the queries of the GOV2 corpus. While *AvICTF*, *AvIDF*, *MaxIDF*, *SCS*, *QS* and *DevIDF* exhibit similar though somewhat lower correlations to each other, *AvQL* and the query collection similarity based predictors, on the other hand, behave differently. The query length based predictor is now consistently uncorrelated to any of the other predictors.

	<i>AvIDF</i>	<i>MaxIDF</i>	<i>DevIDF</i>	<i>SCS</i>	<i>QS</i>	<i>AvICTF</i>	<i>AvQL</i>	<i>SumSCQ</i>	<i>AvSCQ</i>	<i>MaxSCQ</i>
<i>AvIDF</i>		0.721	0.164	0.875	0.683	0.933	0.292	0.155	0.710	0.517
<i>MaxIDF</i>			0.429	0.651	0.417	0.694	0.249	0.165	0.453	0.625
<i>DevIDF</i>				0.119	-0.137	0.142	-0.002	0.203	0.019	0.397
<i>SCS</i>					0.723	0.915	0.310	0.053	0.662	0.439
<i>QS</i>						0.693	0.268	0.076	0.722	0.290
<i>AvICTF</i>							0.295	0.138	0.683	0.469
<i>AvQL</i>								-0.063	0.196	0.111
<i>SumSCQ</i>									0.297	0.236
<i>AvSCQ</i>										0.524

(a) Queries 301-450 (TREC Vol. 4+5)

	<i>AvIDF</i>	<i>MaxIDF</i>	<i>DevIDF</i>	<i>SCS</i>	<i>QS</i>	<i>AvICTF</i>	<i>AvQL</i>	<i>SumSCQ</i>	<i>AvSCQ</i>	<i>MaxSCQ</i>
<i>AvIDF</i>		0.598	0.139	0.833	0.615	0.894	0.052	0.127	0.835	0.515
<i>MaxIDF</i>			0.513	0.521	0.238	0.585	0.032	0.192	0.450	0.777
<i>DevIDF</i>				0.095	-0.220	0.133	0.036	0.157	-0.010	0.445
<i>SCS</i>					0.665	0.895	0.086	-0.023	0.742	0.419
<i>QS</i>						0.613	0.073	-0.043	0.714	0.201
<i>AvICTF</i>							0.068	0.083	0.767	0.476
<i>AvQL</i>								-0.085	0.037	0.148
<i>SumSCQ</i>									0.184	0.253
<i>AvSCQ</i>										0.435

(b) Queries 701-850 (GOV2)

Table 2.3: Kendall’s  $\tau$  between scores of specificity based predictors.

## Predictor Evaluation

While the relationship between the predictors is certainly important (for instance, it is not necessary to report both *AvICTF* and *AvIDF*), the more important question that arises is how well the predictors perform in predicting the retrieval effectiveness of queries. The retrieval effectiveness can be measured in various ways, including average precision, precision at 10 documents, reciprocal rank, and other measures. In this frame of inquiry, average precision is utilized as the measure of true retrieval performance of each query. The predictors were evaluated for their prediction capabilities of TFIDF, Okapi and Language Modeling with Dirichlet smoothing. For the latter, the level of smoothing  $\mu$  was fixed to the best performing retrieval setting as observed in Table 2.2. In Table 2.4 the linear correlation coefficient  $r$  is reported, in Table 2.5 the results of Kendall’s  $\tau$  are listed. The query sets are evaluated individually, as well as combined for all query sets of a particular corpus.

We observe that the predictor performance is influenced considerably by the particular query set under consideration. This observation holds even within the scope of a single collection. *MaxSCQ* for instance can be considered as the best predictor overall, but for one particular query set, 301-350, it breaks down completely, achieving no significant correlation. A contrasting example is *DevIDF*, which generally does not result in meaningful correlations, however for two query sets (401-450, 501-550) it is among the best performing predictors with respect to  $r$ . The group of *AvIDF*, *AvICTF*, *SCS*, *QS* and *MaxIDF* predictors achieve their highest correlations in the TFIDF setting for TREC Vol. 4+5 and the WT10g collection.

When comparing the results of the Okapi and Language Modeling approach across all predictors, considerable differences in predictor performances are only visible for a single query set (701-750). In most other instances the predictors can

Queries		<i>AvIDF</i>	<i>MaxIDF</i>	<i>DevIDF</i>	<i>SCS</i>	<i>QS</i>	<i>AvICTF</i>	<i>AvQL</i>	<i>SumSCQ</i>	<i>AvSCQ</i>	<i>MaxSCQ</i>
301-350	TFIDF	0.809	0.687	-0.068	<b>0.822</b>	0.796	0.813	0.458	-0.340	-0.033	-0.085
	Okapi	<b>0.625</b>	0.609	0.127	0.611	0.557	0.619	0.326	-0.126	0.040	0.110
	$\mu = 500$	<b>0.591</b>	0.574	0.119	0.578	0.531	0.582	0.310	-0.123	0.074	0.122
351-400	TFIDF	<b>0.604</b>	0.422	-0.068	0.584	0.603	0.578	0.014	-0.150	0.442	0.350
	Okapi	0.330	0.346	0.133	0.265	0.252	0.301	-0.210	0.189	0.360	<b>0.465</b>
	$\mu = 2000$	0.374	0.383	0.166	0.319	0.284	0.348	-0.172	0.123	0.412	<b>0.507</b>
401-450	TFIDF	<b>0.541</b>	0.492	0.176	0.540	0.493	0.494	0.465	-0.188	0.333	0.403
	Okapi	0.502	<b>0.587</b>	0.448	0.444	0.302	0.444	0.177	0.039	0.347	0.507
	$\mu = 1000$	0.576	<b>0.649</b>	0.450	0.518	0.381	0.516	0.193	0.046	0.408	0.524
301-450	TFIDF	0.693	0.565	-0.001	<b>0.696</b>	0.673	0.680	0.345	-0.244	0.176	0.150
	Okapi	0.508	<b>0.523</b>	0.226	0.469	0.400	0.483	0.129	0.009	0.214	0.322
	$\mu = 1000$	0.516	<b>0.532</b>	0.239	0.480	0.407	0.490	0.133	-0.002	0.256	0.341
451-500	TFIDF	0.641	0.408	-0.369	0.658	<b>0.699</b>	0.634	0.130	-0.391	0.332	0.092
	Okapi	0.204	0.280	0.158	0.146	0.134	0.193	-0.280	0.105	0.242	<b>0.284</b>
	$\mu = 1000$	0.153	0.214	0.139	0.087	0.092	0.141	-0.262	0.176	0.384	<b>0.429</b>
501-550	TFIDF	0.441	0.398	0.146	0.400	0.318	0.415	0.122	-0.346	0.345	<b>0.442</b>
	Okapi	0.143	0.383	<b>0.415</b>	0.168	-0.092	0.111	0.068	0.160	0.089	0.373
	$\mu = 2000$	0.221	0.469	<b>0.450</b>	0.189	-0.061	0.200	0.052	0.192	0.154	0.393
451-550	TFIDF	<b>0.525</b>	0.386	-0.108	0.523	0.513	0.511	0.127	-0.365	0.332	0.260
	Okapi	0.195	0.315	0.245	0.160	0.075	0.179	-0.147	0.116	0.191	<b>0.309</b>
	$\mu = 1000$	0.182	0.292	0.233	0.126	0.062	0.167	-0.135	0.183	0.307	<b>0.400</b>
701-750	TFIDF	0.247	0.312	0.290	0.207	0.146	0.191	-0.134	-0.041	0.282	<b>0.388</b>
	Okapi	0.202	0.263	0.121	0.128	0.150	0.154	-0.202	0.199	0.290	<b>0.382</b>
	$\mu = 1000$	0.393	0.425	0.160	0.325	0.334	0.354	-0.150	0.151	0.444	<b>0.473</b>
751-800	TFIDF	0.008	0.017	0.019	0.035	0.146	0.031	0.149	-0.125	0.073	<b>0.253</b>
	Okapi	0.304	0.244	0.052	0.274	0.267	0.297	0.049	0.200	<b>0.332</b>	0.283
	$\mu = 1000$	0.315	0.232	0.061	0.278	0.252	0.304	0.122	0.258	<b>0.393</b>	0.371
801-850	TFIDF	<b>0.581</b>	0.435	-0.076	0.534	0.533	0.567	0.042	0.213	0.359	0.225
	Okapi	0.309	0.309	0.147	0.220	0.162	0.263	0.019	0.333	<b>0.368</b>	0.345
	$\mu = 1000$	0.223	0.337	0.317	0.137	0.009	0.185	0.043	0.323	0.248	<b>0.362</b>
701-850	TFIDF	0.270	0.228	0.052	0.250	0.247	0.251	0.045	0.017	0.220	<b>0.272</b>
	Okapi	0.278	0.283	0.121	0.215	0.187	0.245	-0.040	0.229	0.324	<b>0.341</b>
	$\mu = 1000$	0.309	0.331	0.185	0.248	0.179	0.281	0.007	0.235	0.352	<b>0.403</b>

Table 2.4: Linear correlation coefficients  $r$  of specificity-based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

be considered to predict equally well for both retrieval approaches, only *MaxSCQ* performs consequently worse on Okapi. We pointed out earlier, that the only difference between *AvIDF* and *AvICTF* is the reliance on *doccount* versus *termcount*. Across all collections, *AvIDF* is slightly better than *AvICTF*, hence we can conclude that *doccount* is somewhat more reliable. The performance of *SCS* is comparable to *AvICTF*, but always slightly worse than *AvIDF*. The predictors *AvQL*, *DevIDF* and *SumSCQ* consistently perform poorly, at best they result in moderate correlations for one or two query sets. In the case of *AvQL* the reasons for failure are the lack of term length distribution. For instance, consider query set 701-750, where 31 out of 50 queries have an average term length between 5 and 6, rendering the predictor unusable. Note that the spread is considerably larger for queries 301-350, where *AvQL* results in a small positive correlation.

Overall, the predictor *MaxSCQ* performs best, however due to its drastic failure on query set 301-350, a safer choice would be the slightly worse performing *MaxIDF*. If we focus on the corpora, we observe that TREC Vol. 4+5 is easiest to predict for, whereas the WT10g and GOV2 corpora pose significant difficulties to the predictors. Although our observations hold for both the linear correlation coeffi-

Queries		<i>AvIDF</i>	<i>MaxIDF</i>	<i>DevIDF</i>	<i>SCS</i>	<i>QS</i>	<i>AvICTF</i>	<i>AvQL</i>	<i>SumSCQ</i>	<i>AvSCQ</i>	<i>MaxSCQ</i>
301- 350	<b>TEIDF</b>	<b>0.480</b>	0.474	0.093	0.439	0.356	0.465	0.281	0.045	0.225	0.286
	<b>Okapi</b>	0.348	<b>0.409</b>	0.115	0.304	0.220	0.327	0.171	0.067	0.093	0.162
	$\mu = 500$	0.314	<b>0.368</b>	0.086	0.286	0.219	0.289	0.165	0.087	0.095	0.181
351- 400	<b>TEIDF</b>	0.355	0.336	0.045	0.328	0.333	0.336	-0.043	0.042	0.368	<b>0.413</b>
	<b>Okapi</b>	0.244	0.287	0.116	0.180	0.200	0.202	-0.097	0.146	0.275	<b>0.398</b>
	$\mu = 2000$	0.271	0.307	0.153	0.227	0.227	0.238	-0.095	0.126	0.315	<b>0.422</b>
401- 450	<b>TEIDF</b>	0.310	0.320	0.146	0.300	0.197	0.293	0.250	-0.009	0.275	<b>0.439</b>
	<b>Okapi</b>	0.275	0.354	0.276	0.252	0.146	0.249	0.046	0.033	0.228	<b>0.424</b>
	$\mu = 1000$	0.313	0.402	0.314	0.277	0.161	0.273	0.048	0.058	0.265	<b>0.474</b>
301- 450	<b>TEIDF</b>	<b>0.400</b>	0.390	0.113	0.375	0.299	0.383	0.169	0.019	0.292	0.373
	<b>Okapi</b>	0.287	<b>0.340</b>	0.177	0.248	0.188	0.265	0.042	0.089	0.195	0.330
	$\mu = 1000$	0.290	<b>0.340</b>	0.180	0.251	0.190	0.266	0.039	0.080	0.204	0.332
451- 500	<b>TEIDF</b>	0.480	0.316	-0.182	0.480	<b>0.494</b>	0.470	0.100	-0.169	0.448	0.364
	<b>Okapi</b>	0.261	<b>0.361</b>	0.144	0.203	0.151	0.254	-0.115	0.079	0.188	0.336
	$\mu = 1000$	0.249	0.281	0.137	0.174	0.135	0.236	-0.076	0.147	0.321	<b>0.435</b>
501- 550	<b>TEIDF</b>	0.364	0.355	0.017	0.349	0.236	0.338	0.202	-0.246	0.337	<b>0.391</b>
	<b>Okapi</b>	0.139	0.233	0.184	0.156	0.005	0.099	0.109	0.087	0.102	<b>0.240</b>
	$\mu = 2000$	0.187	<b>0.277</b>	0.174	0.136	0.046	0.143	0.087	0.111	0.160	0.270
451- 550	<b>TEIDF</b>	<b>0.403</b>	0.319	-0.085	0.401	0.355	0.393	0.155	-0.210	0.371	0.354
	<b>Okapi</b>	0.192	<b>0.274</b>	0.165	0.175	0.069	0.177	-0.019	0.081	0.132	0.262
	$\mu = 1000$	0.213	0.266	0.157	0.163	0.079	0.192	-0.005	0.138	0.227	<b>0.322</b>
701- 750	<b>TEIDF</b>	0.186	0.258	0.257	0.186	0.084	0.173	0.017	-0.045	0.188	<b>0.297</b>
	<b>Okapi</b>	0.151	0.189	0.050	0.099	0.124	0.112	-0.111	0.160	0.184	<b>0.247</b>
	$\mu = 1000$	0.277	0.304	0.108	0.211	0.218	0.248	-0.065	0.161	0.300	<b>0.331</b>
751- 800	<b>TEIDF</b>	-0.016	0.034	0.021	-0.006	0.084	-0.034	0.082	-0.002	0.012	<b>0.173</b>
	<b>Okapi</b>	0.207	0.169	0.011	0.192	0.193	0.205	0.041	0.151	<b>0.224</b>	0.174
	$\mu = 1000$	0.253	0.204	0.059	0.240	0.217	0.260	0.117	0.165	0.274	<b>0.291</b>
801- 850	<b>TEIDF</b>	0.255	<b>0.267</b>	0.144	0.193	0.094	0.232	0.104	0.219	0.221	0.250
	<b>Okapi</b>	0.246	0.218	0.144	0.166	0.118	0.205	0.045	<b>0.277</b>	0.256	0.241
	$\mu = 1000$	0.193	0.228	<b>0.255</b>	0.130	0.004	0.166	0.057	0.238	0.171	0.241
701- 850	<b>TEIDF</b>	0.120	0.176	0.127	0.096	0.040	0.103	0.077	0.045	0.122	<b>0.229</b>
	<b>Okapi</b>	0.199	0.195	0.076	0.151	0.142	0.172	-0.011	0.182	0.216	<b>0.221</b>
	$\mu = 1000$	0.229	0.243	0.143	0.186	0.137	0.209	0.028	0.179	0.234	<b>0.274</b>

Table 2.5: Kendall’s  $\tau$  coefficients of specificity-based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

cient  $r$  and Kendall’s  $\tau$ , there are also differences visible when comparing Tables 2.4 and 2.5. Comparing the performance of *MaxIDF* and *MaxSCQ* for queries 301-450 yields hardly any differences in performance when reporting  $\tau$  ( $\tau_{MaxIDF} = 0.34$ ,  $\tau_{MaxSCQ} = 0.33$ ); the linear correlation coefficient on the other hand indicates a considerable performance gap, namely,  $r_{MaxIDF} = 0.52$  versus  $r_{MaxSCQ} = 0.34$ . Thus, if query performance prediction should be applied in a practical setup, where the average precision score is of importance, *MaxIDF* is a better predictor than *MaxSCQ*, while the reverse is true if the application relies on the effectiveness ranking of the queries.

Due to the nature of most specificity based prediction methods, it is expected that the amount of smoothing in the Language Modeling approach will have a considerable influence on their quality as increased smoothing results in an increasing influence of collection statistics. To investigate the influence of high levels of smoothing,  $\mu$  is evaluated for levels ranging from  $\mu = 5 \times 10^3$  to  $\mu = 3.5 \times 10^5$  (more specifics are given in Appendix B, Figure B.2). We report the prediction accuracy of *AvIDF*, *MaxIDF*, *SCS*, *AvSCQ* and *MaxSCQ*, the remaining predictors were excluded either due to poor performance or their similarity to one of the reported predictors.

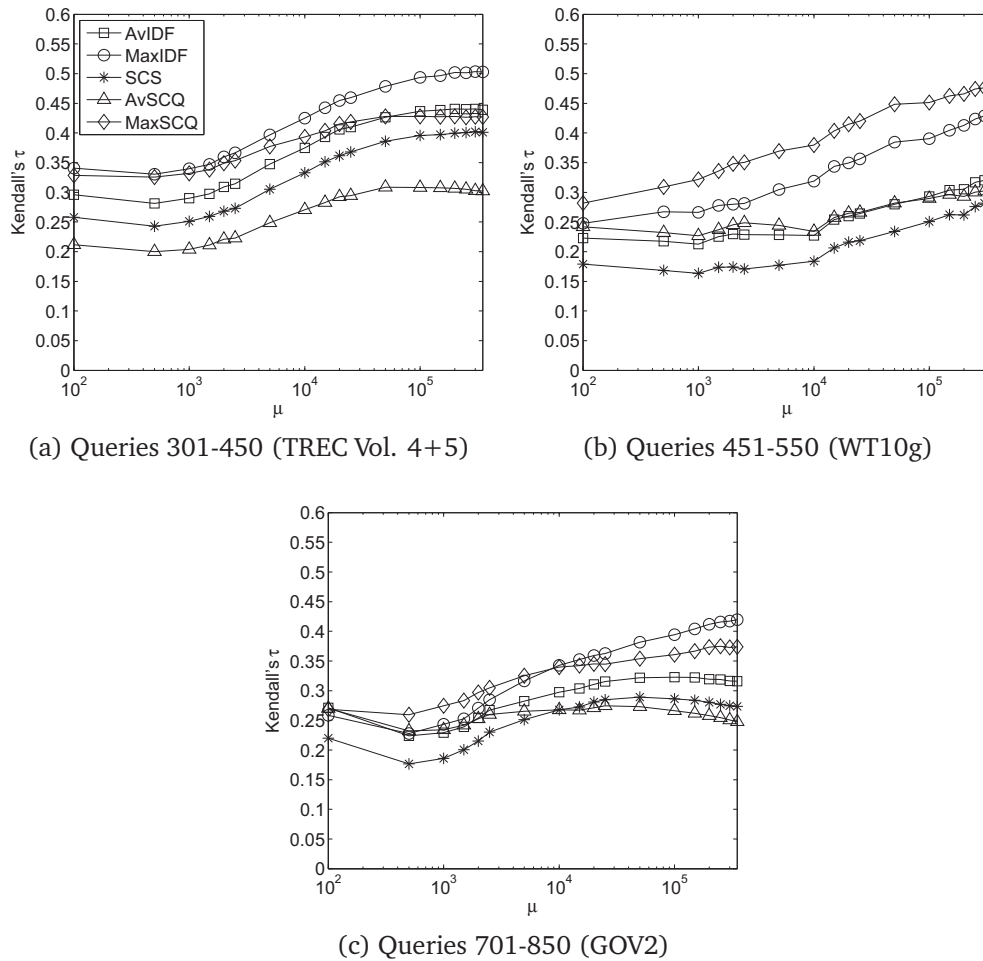


Figure 2.4: The influence of the level  $\mu$  of smoothing on the accuracy of various predictors.

The results shown in Figure 2.4 are reported in terms of Kendall's  $\tau$  (the results were similar for the linear correlation coefficient). They confirm the hypothesis, that increasing levels of  $\mu$  generally lead to a positive change in correlation for the specificity-based predictors. The relative predictor performance remains largely the same, the correlation increases occur to similar degrees. Depending on the corpus and the predictor, the performance difference can be large, for instance at low levels of smoothing *MaxIDF* has a correlation of  $\tau_{\mu=500} = 0.33$ , whereas it reaches  $\tau_{\mu=3 \times 10^5} = 0.5$  when the amount of smoothing is increased. Changing  $\mu$  has the least effect on *AvSCQ*; although its correlation also rises with the rise of  $\mu$ , the improvements are small and they trail off after  $\tau$  reaches  $5 \times 10^4$  for TREC Vol. 4+5 and GOV2.

## 2.7 Ranking Sensitivity

Although pre-retrieval predictors do not consider the ranked list of results returned by the retrieval system for a given query, they can still rely on collection statistics to infer how difficult it will be for the system to rank the documents according to the query. The three predictors in this category are all variations of the same principle, and are presented below:

- Summed Term Weight Variability (*SumVAR*) [174],
- Averaged Term Weight Variability (*AvVAR*) [174], and,
- Maximum Term Weight Variability (*MaxVAR*) [174].

### 2.7.1 Collection Based Sensitivity

This family of predictors exploits the distribution of term weights across the collection. If the term weights across all documents containing query term  $q_i$  are similar, there is little evidence for a retrieval system on how to rank those documents given  $q_i$ , and thus different retrieval algorithms are likely to produce widely different rankings. Conversely, if the term weights differ widely across the collection, ranking becomes easier and different retrieval algorithms are expected to produce similar rankings. Here we assume that the retrieval system relies solely on collection statistics, without considering external sources or additional information.

In [174], the term weight  $w(q_i, d)$  is based on TF.IDF, the average term weight  $\bar{w}_{q_i}$  is the average weight over all documents containing  $q_i$ . *SumVAR* is the sum of the query term weight deviations:

$$SumVAR = \sum_{i=1}^m \sqrt{\frac{1}{df(q_i)} \sum_{d \in N_{q_i}} (w(q_i, d) - \bar{w}_{q_i})^2}. \quad (2.13)$$

In contrast to *SumVAR* which is not normalized according to the query length,  $AvVAR = \frac{1}{m} \times SumVAR$  is normalized. Finally, the maximum variability score over all query terms is used as prediction score for the predictor *MaxVAR*. Note, that the three predictors in this category are more complex than for example *MaxIDF*, as they rely on TF.IDF weights and require additional pre-processing.

### 2.7.2 Experimental Evaluation

Analogous to the specificity based predictors, the algorithms in this category are first evaluated with respect to their similarity to each other. Then, their ability to predict retrieval effectiveness will be evaluated.

#### Predictor-Predictor Correlations

In Table 2.6 the correlations between the predictor scores are shown. While the results of the query sets of TREC Vol. 4+5 and GOV2 are similar, with *AvVAR* and

*MaxVAR* being more closely related to each other than to *SumVAR*, in the WT10g collection, *SumVAR* is hardly related to the other two predictor variations. Since *SumVAR* is not normalized with regard to query length, we expect it to perform rather poorly as predictor.

	<i>SumVAR</i>	<i>AvVAR</i>	<i>MaxVAR</i>
<i>SumVAR</i>		0.546	0.561
<i>AvVAR</i>			0.721

(a) Queries 301-450 (TREC Vol. 4+5)

	<i>SumVAR</i>	<i>AvVAR</i>	<i>MaxVAR</i>
<i>SumVAR</i>		0.075	0.210
<i>AvVAR</i>			0.669

(b) Queries 451-550 (WT10g)

	<i>SumVAR</i>	<i>AvVAR</i>	<i>MaxVAR</i>
<i>SumVAR</i>		0.397	0.478
<i>AvVAR</i>			0.616

(c) Queries 701-850 (GOV2)

Table 2.6: Kendall’s  $\tau$  between scores of ranking sensitivity based predictors.

## Predictor Evaluation

Table 2.7 contains the correlation coefficients the predictors achieve across all query sets and across the standard retrieval approaches. Of the three predictor variations, *SumVAR* is the most erratic. This is not surprising, as it is not normalized with respect to the number of terms in the queries. *MaxVAR* is the best predictor of this category, with a surprisingly good performance on query set 501-550 of the WT10g collection, which provided the most difficulties to the specificity based predictors. *AvVAR*’s overall performance is slightly worse than *MaxVAR*’s. There are two query sets which yield somewhat unexpected results: for one, query set 301-350, which has shown to be the easiest for *AvIDF* and related predictors (leading to the highest observed correlation), is the most difficult for the ranking sensitivity based predictors. Secondly, query set 801-850 shows hardly any variation for the performance of the three predictors, unlike the other query sets.

Similar to the observations made for the specificity based predictors, increasing the level of smoothing in the Language Modeling approach increases the correlation coefficients of *AvVAR* and *MaxVAR* across the three corpora. The largest improvements are recorded for the query sets of the WT10g corpus; the correlation of *MaxVAR* ranges from  $\tau_{\mu=100} = 0.29$  to  $\tau_{\mu=3.5 \times 10^5} = 0.44$  at the highest level of smoothing. Smaller improvements up to  $\tau = 0.1$  are also achieved for TREC Vol. 4+5 and *MaxVAR*, where  $\tau$  peaks at  $\mu = 2.5 \times 10^4$ . Relatively unaffected is the GOV2 corpus, where the trend is positive, but the changes in correlation are minor. The *SumVAR* predictor, on the other hand, continuously degrades when the level of smoothing is improved; it achieves its highest correlation at  $\mu = 100$ .

		SumVAR	AvVAR	MaxVAR			SumVAR	AvVAR	MaxVAR
301-350	TEIDF	-0.035	<b>0.306</b>	0.203	301-350	TEIDF	0.166	0.383	<b>0.390</b>
	Okapi	0.151	<b>0.371</b>	0.359		Okapi	0.201	0.302	<b>0.367</b>
	$\mu = 500$	0.163	<b>0.403</b>	0.369		$\mu = 500$	0.203	0.291	<b>0.353</b>
351-400	TEIDF	0.149	<b>0.583</b>	0.455	351-400	TEIDF	0.218	<b>0.434</b>	0.410
	Okapi	0.318	0.400	<b>0.426</b>		Okapi	0.334	0.339	<b>0.415</b>
	$\mu = 2000$	0.288	0.431	<b>0.445</b>		$\mu = 2000$	0.317	0.382	<b>0.434</b>
401-450	TEIDF	0.262	<b>0.706</b>	0.631	401-450	TEIDF	0.252	0.413	<b>0.437</b>
	Okapi	0.517	0.699	<b>0.723</b>		Okapi	0.304	0.432	<b>0.443</b>
	$\mu = 1000$	0.552	0.758	<b>0.764</b>		$\mu = 1000$	0.352	0.460	<b>0.494</b>
301-450	TEIDF	0.089	<b>0.487</b>	0.388	301-450	TEIDF	0.220	0.403	<b>0.417</b>
	Okapi	0.293	0.476	<b>0.491</b>		Okapi	0.285	0.356	<b>0.407</b>
	$\mu = 1000$	0.297	0.510	<b>0.513</b>		$\mu = 1000$	0.283	0.356	<b>0.411</b>
451-500	TEIDF	-0.266	<b>0.336</b>	0.181	451-500	TEIDF	-0.078	<b>0.424</b>	0.330
	Okapi	0.173	0.197	<b>0.253</b>		Okapi	0.118	0.188	<b>0.241</b>
	$\mu = 1000$	0.259	0.324	<b>0.381</b>		$\mu = 1000$	0.203	0.300	<b>0.339</b>
501-550	TEIDF	-0.168	0.489	<b>0.566</b>	501-550	TEIDF	-0.154	0.400	<b>0.451</b>
	Okapi	0.336	0.201	<b>0.513</b>		Okapi	0.189	0.189	<b>0.323</b>
	$\mu = 2000$	0.366	0.233	<b>0.533</b>		$\mu = 2000$	0.189	0.233	<b>0.327</b>
451-550	TEIDF	-0.219	<b>0.401</b>	0.366	451-550	TEIDF	-0.121	<b>0.394</b>	0.385
	Okapi	0.221	0.198	<b>0.337</b>		Okapi	0.145	0.168	<b>0.262</b>
	$\mu = 1000$	0.300	0.291	<b>0.411</b>		$\mu = 1000$	0.213	0.249	<b>0.321</b>
701-750	TEIDF	0.160	0.442	<b>0.479</b>	701-750	TEIDF	0.093	0.287	<b>0.336</b>
	Okapi	0.360	0.392	<b>0.437</b>		Okapi	0.245	0.261	<b>0.276</b>
	$\mu = 1000$	0.293	<b>0.464</b>	0.435		$\mu = 1000$	0.250	<b>0.330</b>	0.288
751-800	TEIDF	-0.062	0.119	<b>0.167</b>	751-800	TEIDF	0.012	0.089	<b>0.172</b>
	Okapi	0.295	<b>0.406</b>	0.371		Okapi	0.197	<b>0.259</b>	0.247
	$\mu = 1000$	0.363	<b>0.438</b>	0.434		$\mu = 1000$	0.230	0.292	<b>0.318</b>
801-850	TEIDF	0.357	<b>0.495</b>	0.355	801-850	TEIDF	0.204	0.242	<b>0.272</b>
	Okapi	0.401	<b>0.430</b>	0.420		Okapi	0.303	<b>0.314</b>	0.306
	$\mu = 1000$	0.380	0.314	<b>0.389</b>		$\mu = 1000$	<b>0.280</b>	0.233	0.274
701-850	TEIDF	0.143	<b>0.323</b>	0.300	701-850	TEIDF	0.107	0.198	<b>0.241</b>
	Okapi	0.336	0.397	<b>0.402</b>		Okapi	0.237	<b>0.268</b>	0.267
	$\mu = 1000$	0.337	0.392	<b>0.412</b>		$\mu = 1000$	0.241	0.269	<b>0.280</b>

(a) Linear correlation coefficient  $r$ (b) Kendall's  $\tau$ 

Table 2.7: Correlation coefficients of ranking sensitivity based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

## 2.8 Ambiguity

Predictors that infer the quality of a query from the ambiguity of the query terms include the following:

- Averaged Query Term Coherence (AvQC) [73],
- Averaged Query Term Coherence with Global Constraint (AvQCG)[73],
- Averaged Polysemy (AvP) [111], and,
- Averaged Noun Polysemy (AvNP).

The first two predictors rely on the collection and, specifically, on all documents containing any of the query terms, to determine the amount of ambiguity. The latter two predictors exploit WordNet, an external source which provides the number of senses a term has, thus making further calculations on the corpus unnecessary.

### 2.8.1 Collection Based Ambiguity

He et al. [73] derive the ambiguity of a query term  $q_i$  by calculating the similarity between all documents that contain  $q_i$ . The set of all those documents is  $N_{q_i}$ , with  $|N_{q_i}| = n$ . The *set coherence* of  $N_{q_i}$  is then defined as:

$$\text{SetCoherence}(N_{q_i}) = \frac{\sum_{i \neq j \in \{1, \dots, n\}} \sigma(d_i, d_j)}{n(n-1)}$$

where  $\sigma(d_i, d_j)$  is a similarity function that returns 1 if the similarity between  $d_i$  and  $d_j$  exceeds a threshold  $\theta$ ; otherwise  $\sigma = 0$ . The *SetCoherence* is defined in the interval  $[0, 1]$ : *SetCoherence* = 1 if all documents in  $N_{q_i}$  are similar to each other, and *SetCoherence* = 0 if none are.

When viewed as a clustering task, the documents in  $N_{q_i}$  are clustered agglomeratively. Initially, each document is assigned its own cluster and iteratively the two closest clusters are merged. The distance between two clusters is given by the distance of the two farthest points in the clusters (complete linkage clustering). The merging process stops, if the merged clusters have a similarity less than  $\theta$ . *SetCoherence* is then the number of links between nodes within a cluster, divided by the number of links between all nodes independent of the cluster. In the ideal case, all documents are clustered into a single cluster. The *SetCoherence* score is mainly influenced by the size of the largest cluster: the larger the dominant cluster, the larger the score. Equally sized clusters receive a lower *SetCoherence* score.

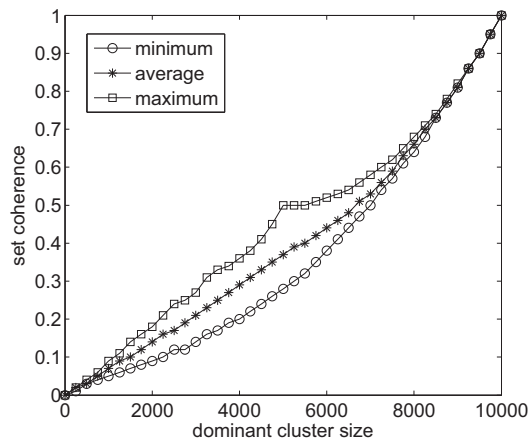


Figure 2.5: The development of the *SetCoherence* score with increased size of the dominant cluster.

We investigated the influence of the dominant cluster size with a small simulation experiment. The size of the document set to cluster was fixed to 10000 documents, while the size of the dominant cluster was varied between 1 and 10000 with a step size of 250. Once the dominant cluster is fixed, cluster sizes, with the restriction of being smaller than the dominant cluster, are randomly generated until the number of 10000 documents is reached. This process is repeated 10000 times for each dominant cluster size. Figure 2.5 contains the minimum, average and maximum

*SetCoherence* for each dominant cluster size. When 50% of all documents belong to the dominant cluster, *SetCoherence* = 0.37 on average, although one might expect a higher score as half of all possible documents belong to a single cluster.

In the work by He et al. [73], the documents are vectors and  $\sigma$  is the cosine similarity. The similarity threshold  $\theta$  is set heuristically by averaging the top 5% of similarity scores from randomly sampled sets of documents. *AvQC* is the average set coherence over all query terms.

Additionally, the following constraint is added to *AvQCG*:

$$AvQCG = SetCoherence(N_q) \times AvQC.$$

Here, *AvQC* is multiplied by the global set coherence, that is the coherence of the set of documents that contains any of the query terms:  $N_q = \cup_{i=1}^m N_{q_i}$ . If  $N_q$  is large (in [73] the limit of 10000 documents is given for the AP88 & 89 corpus), the global set coherence is approximated by the threshold  $\theta$ . In particular, for longer queries with a high number of general terms it can be expected that the global set coherence is close to constant for a set of queries, as in almost all cases  $\theta$  will be used as global set coherence. A similar result is expected for a query set from a large corpus. The GOV2 corpus contains 25 million documents and even specific terms will often appear in more than 10000 documents. We implemented the proposed method to the highest precision degree possible, as the original publication did not disclose all details. For the newspaper corpus, the limit was set to 10000 documents, for the WT10g corpus the limit was increased to 20000 documents and for the GOV2 collection it was set to 50000 documents.

*AvQC* and *AvQCG* both require a great amount of computation; determining the document similarity between all document pairs of a collection for example is not feasible, samples have to be drawn instead.

## 2.8.2 Ambiguity as Covered by WordNet

WordNet [57] is an online lexical database developed at Princeton University, inspired by psycholinguistic theories. It is continuously enlarged and updated by human experts and can be viewed as a general domain knowledge base. WordNet's building blocks are sets of synonymous terms<sup>3</sup>, called *synsets*, each representing a lexical concept and each connected to others through a range of semantic relationships. Relations between terms instead of synsets exist as well but are not very frequent. WordNet also provides glosses, which are example sentences and definitions for the synsets. Relationships exist mainly between synsets of the same word type; there are separate structures for nouns, verbs, adjectives and adverbs. Notably, nouns make up by far the largest fraction of WordNet.

The number of WordNet senses of a term is an indicator of its ambiguity - the more senses a term has, the more ambiguous it is. For example, the term “go” has a total of thirty-five senses in WordNet<sup>4</sup>. These are, four noun senses, one adjective

<sup>3</sup>A term can be a single word, a compound or a phrase.

<sup>4</sup>All figures are based on WordNet version 3.0.

sense and thirty verb senses. On the other hand, “*Agoraphobia*” has a single noun sense and thus is considered to be unambiguous. A limiting factor of WordNet is the fact that it is a general knowledge semantic dictionary and therefore it is only useful for a general collection of documents. Additionally, WordNet also contains rare senses of terms, which may not appear at all in a corpus, while many proper nouns that do appear in a corpus may not be a part of WordNet.

For each synset, WordNet provides a gloss of varying length. Take for example the concepts “viral hepatitis” and “aspirin” from TREC description queries. The gloss of the former is: “hepatitis caused by a virus” whereas “aspirin” is described as follows: “the acetylated derivative of salicylic acid; used as an analgesic anti-inflammatory drug (trade names Bayer, Empirin, and St. Joseph) usually taken in tablet form; used as an antipyretic; slows clotting of the blood by poisoning platelets”.

The  $AvP$  [111] value is derived from WordNet in the following way. Initially, each query is tokenized and mapped to WordNet terms. Since WordNet contains phrases such as “organized crime”, the matching is first performed based on a window of five terms, then four terms and so on, with morphological variations also being tested. Then the number of senses of each phrase found is recorded - a term that is not found in WordNet and is not part of a WordNet phrase, is assigned a single sense. Finally, the average number of senses over all found phrases/terms is calculated. For example, TREC title topic “black bear attacks” is WordNet tokenized into  $\{black\ bear, attack\}$ . The phrase “black bear” has two senses, while “attack” has fifteen senses, and therefore  $AvP = 8.5$ . For comparison purposes, we also evaluate  $AvNP$ , which is similar to  $AvP$  but it only considers the noun senses instead of the senses over all word types.

### 2.8.3 Experimental Evaluation

This segment contains the results of the evaluation of the presented ambiguity based predictors. The presentation of numerical findings is accompanied by a discussion on the causes of discovered differences.

#### Predictor-Predictor Correlations

The correlation between the  $AvQC$  and  $AvQCG$  predictors is high across all three corpora. With increased collection size, the correlation approaches one, specifically  $\tau = 0.87$  for the queries of TREC Vol. 4+5 and  $\tau = 0.98$  for the queries of the GOV2 corpus. The two WordNet based prediction methods are less highly correlated, reaching  $\tau = 0.8$  at best. The correlation between the WordNet based and the collection based predictors is moderately negative for TREC Vol. 4+5 and approximately zero for the queries of WT10g and GOV2. The negative correlation can be attributed to the fact that the more WordNet senses the query terms have, the lower the quality of the query, whereas the collection based predictors predict a higher quality with increased score.

## Predictor Evaluation

The results in Table 2.8 show that the two WordNet based predictors (*AvP* and *AvNP*) generally perform very poorly; only for query sets 301-350 and 451-500 do they exhibit meaningful negative correlations across the range of retrieval methods. The reason for this failure can be attributed, in part, to the fact that the TREC title topics of the WT10g and the GOV2 corpus contain a significant number of proper nouns such as “Chevrolet”, “Skoda”, “Peer Gynt”, “Nirvana”, “John Edwards” and “TMJ” which are not part of WordNet. As these terms and phrases often make up the most important or even the sole part of a title topic, the results become unusable. A second reason for the discrepancy is rooted in the collection size and makeup. Arguably, the newswire corpus (TREC Vol. 4+5) employs a limited vocabulary and reasonably structured prose, while the newer Web and Terabyte corpora contain a more diverse vocabulary with more noise (frequent use of esoteric or non-sensical words and phrases). In such cases, WordNet does not provide an accurate sense count.

		<i>AvP</i>	<i>AvNP</i>	<i>AvQC</i>	<i>AvQCG</i>			<i>AvP</i>	<i>AvNP</i>	<i>AvQC</i>	<i>AvQCG</i>
301-350	TEIDF	-0.283	-0.354	0.545	<b>0.584</b>	301-305	TEIDF	-0.283	-0.329	0.483	<b>0.503</b>
	Okapi	-0.360	-0.467	<b>0.487</b>	0.436		Okapi	-0.314	<b>-0.375</b>	0.370	0.374
	$\mu = 500$	-0.347	-0.445	<b>0.449</b>	0.404		$\mu = 500$	-0.334	<b>-0.371</b>	0.350	0.347
351-400	TEIDF	-0.329	-0.290	0.525	<b>0.611</b>	351-400	TEIDF	-0.208	-0.186	0.297	<b>0.318</b>
	Okapi	-0.104	-0.160	0.192	<b>0.245</b>		Okapi	-0.039	-0.094	0.181	<b>0.209</b>
	$\mu = 2000$	-0.131	-0.153	0.238	<b>0.273</b>		$\mu = 2000$	-0.046	-0.065	0.213	<b>0.241</b>
401-450	TEIDF	0.009	-0.110	<b>0.732</b>	0.551	401-450	TEIDF	-0.129	-0.128	0.382	<b>0.412</b>
	Okapi	0.125	0.022	<b>0.611</b>	0.365		Okapi	0.029	0.032	<b>0.341</b>	0.311
	$\mu = 1000$	0.076	-0.069	<b>0.627</b>	0.390		$\mu = 1000$	0.007	-0.014	<b>0.385</b>	0.352
301-450	TEIDF	-0.187	-0.248	<b>0.591</b>	0.475	301-450	TEIDF	-0.210	-0.227	0.397	<b>0.421</b>
	Okapi	-0.115	-0.211	<b>0.456</b>	0.316		Okapi	-0.107	0.152	0.295	<b>0.298</b>
	$\mu = 1000$	-0.121	-0.220	<b>0.457</b>	0.330		$\mu = 1000$	-0.118	-0.157	0.296	<b>0.301</b>
451-500	TEIDF	-0.305	-0.253	<b>0.657</b>	0.544	451-500	TEIDF	-0.292	-0.185	<b>0.540</b>	0.529
	Okapi	<b>-0.303</b>	-0.262	0.208	0.091		Okapi	-0.271	-0.226	<b>0.293</b>	0.276
	$\mu = 1000$	<b>-0.305</b>	-0.246	0.138	-0.047		$\mu = 1000$	-0.195	-0.143	<b>0.260</b>	0.246
501-550	TEIDF	-0.282	-0.247	<b>0.512</b>	0.394	501-550	TEIDF	-0.247	-0.176	0.418	<b>0.423</b>
	Okapi	-0.064	0.063	<b>0.154</b>	0.056		Okapi	-0.053	0.053	<b>0.098</b>	0.093
	$\mu = 2000$	0.015	0.109	<b>0.210</b>	0.052		$\mu = 2000$	-0.044	0.084	<b>0.152</b>	0.147
451-550	TEIDF	-0.289	-0.247	<b>0.579</b>	0.460	451-550	TEIDF	-0.261	-0.194	<b>0.458</b>	0.454
	Okapi	<b>-0.210</b>	-0.141	0.195	0.088		Okapi	-0.153	0.088	<b>0.195</b>	0.178
	$\mu = 1000$	<b>-0.183</b>	-0.108	0.162	-0.021		$\mu = 1000$	-0.114	0.044	<b>0.210</b>	0.195
701-750	TEIDF	0.075	0.027	<b>0.221</b>	0.020	701-750	TEIDF	0.111	0.118	<b>0.264</b>	<b>0.264</b>
	Okapi	0.047	-0.068	<b>0.177</b>	0.131		Okapi	0.029	-0.027	<b>0.163</b>	<b>0.163</b>
	$\mu = 1000$	0.050	0.007	<b>0.253</b>	0.104		$\mu = 1000$	0.031	0.010	0.279	<b>0.287</b>
751-800	TEIDF	-0.184	-0.170	0.145	<b>0.206</b>	751-800	TEIDF	-0.110	-0.045	0.288	<b>0.298</b>
	Okapi	0.130	-0.042	<b>0.371</b>	0.182		Okapi	-0.031	-0.046	<b>0.268</b>	0.258
	$\mu = 1000$	0.014	-0.040	<b>0.410</b>	0.164		$\mu = 1000$	-0.015	0.007	<b>0.280</b>	0.267
801-850	TEIDF	-0.188	-0.224	0.384	<b>0.427</b>	801-850	TEIDF	-0.171	-0.208	0.272	<b>0.298</b>
	Okapi	-0.014	-0.119	<b>0.274</b>	0.010		Okapi	0.005	-0.093	0.247	<b>0.251</b>
	$\mu = 1000$	-0.038	-0.111	<b>0.265</b>	0.006		$\mu = 1000$	0.027	-0.038	<b>0.268</b>	0.249
701-850	TEIDF	-0.124	-0.124	<b>0.252</b>	0.214	701-850	TEIDF	-0.059	-0.029	0.269	<b>0.282</b>
	Okapi	0.075	-0.049	<b>0.263</b>	0.060		Okapi	0.017	-0.026	<b>0.232</b>	0.226
	$\mu = 1000$	0.017	-0.035	<b>0.298</b>	0.048		$\mu = 1000$	0.027	0.014	<b>0.273</b>	0.265

(a) Linear correlation coefficient  $r$ (b) Kendall's  $\tau$ 

Table 2.8: Correlation coefficients of ambiguity based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

The predictor performances of *AvQC* and *AvQCG* are mixed and greatly depend on the particular collection. The best performance is achieved for the query sets of TREC Vol. 4+5. For the query sets of WT10g and GOV2, in many instances only insignificant correlation coefficients are achieved. This might be due to the fact that *AvQC* is geared towards smaller collections or that our parameter settings were not optimal. Note that thorough calibration of parameters has not been conducted during this work, given time constraints and the great computational requirement of this method.

With respect to the level of smoothing in the Language Modeling approach, the two clustering based predictors *AvQC* and *AvQCG* show considerable performance increases over the three corpora when the amount of smoothing is increased. On query set 701-850 for instance, *AvQC* reaches  $\tau_{\mu=100} = 0.27$  for low amounts of smoothing; however, when  $\mu$  is raised to the maximum,  $\tau$  reaches 0.39. The two WordNet based predictors on the other hand show hardly any change in correlation with changing amounts of smoothing.

## 2.9 Term Relatedness

The previously introduced specificity and ambiguity based predictors ignore an important aspect of the query, namely the relationship between the query terms. Consider for example the two queries  $\mathbf{q}_1 = \{American, football\}$  and  $\mathbf{q}_2 = \{foot, porch\}$ . Specificity based predictors might predict  $\mathbf{q}_2$  to be an easier query because the terms *foot* and *porch* might occur less frequently than *American* and *football*. However, in a general corpus one would expect  $\mathbf{q}_1$  to be an easier query for a retrieval system than  $\mathbf{q}_2$  due to the strong relationship between the two query terms. Term relatedness measures predict a query to perform well, if there is a measurable relationship between query terms. The degree of relationship can either be derived from co-occurrence statistics of the collection or from WordNet based measures that determine the degree of *semantic* relatedness.

The predictors surveyed in this section are:

- Averaged Pointwise Mutual Information (*AvPMI*),
- Maximum Pointwise Mutual Information (*MaxPMI*),
- Averaged Path Length (*AvPath*) [124],
- Averaged Lesk Relatedness (*AvLesk*) [15], and,
- Averaged Vector Pair Relatedness (*AvVP*) [116].

A drawback of these predictors is, that queries consisting of a single term will be assigned a score of zero, as in such cases no relatedness value can be derived. If a significant number of queries in the query set used for evaluation are single term queries, the correlation will be lower than what the actual quality of the predictor implies.

### 2.9.1 Collection Based Relatedness

Predictors that exploit co-occurrence statistics of the collection are more precise than those based on standard deviations such as *DevIDF*. *AvPMI* and *MaxPMI* both rely on the concept of pointwise mutual information, which for two terms  $q_i$  and  $q_j$  is defined by:

$$PMI(q_i, q_j) = \log_2 \frac{P_{ml}(q_i, q_j)}{P_{ml}(q_i)P_{ml}(q_j)}.$$

The nominator is the probability that the two terms occur together in a document; the denominator is the probability of them occurring together by chance. If  $P_{ml}(q_i, q_j) \approx P_{ml}(q_i)P_{ml}(q_j)$ , the terms are independent and  $PMI \approx 0$ . Query terms that co-occur significantly more often than by chance lead to a high *PMI* value. *AvPMI* is the average over all *PMI* scores across all query term pairs, while *MaxPMI* is the maximum *PMI* score across all query term pairs.

### 2.9.2 WordNet Based Relatedness

As an alternative to the collection statistics based methods, the degree of relatedness of query terms can also be determined by exploiting the graph structure of WordNet. In general, the closer two terms are in the WordNet graph, the higher their semantic similarity. Diverse WordNet based measures exist; in this work, we evaluate three measures as pre-retrieval predictors.

*AvPath*, initially proposed by Rada et al. [124], determines the relatedness between two terms by the reciprocal of the number of nodes on the shortest path of the IS-A hierarchy between the two corresponding synset nodes. Since the IS-A relationship is defined on the noun graph, the measure ignores all non-noun query terms. The maximum relatedness score is one (two identical synsets) and the minimum is zero (no path between two synsets). The average over all query term pair scores is then utilized as *AvPath* score<sup>5</sup>.

*AvLesk* [14] is a relatedness measure that exploits the gloss overlap between two synsets, as well as the glosses of their related synsets. Generally, the more terms the glosses have in common, the more related the two synsets are.

Finally, *AvVP*, introduced by Patwardhan and Pedersen [116], is a measure where each synset is represented as a second-order co-occurrence vector of glosses, including the glosses of related synsets. Relatedness, in this case, is the cosine similarity between the gloss vectors.

The three measures just described rely on synsets instead of terms. In a practical applications, it would be necessary to first disambiguate the query terms and then to locate the correct synset in WordNet. Since in this experiment we are interested in the general feasibility of WordNet based relatedness measures, in a preprocessing step we manually disambiguated the query terms and identified the correct synset. A number of proper nouns in the queries could not be matched and had to be ignored in the relatedness calculations.

<sup>5</sup>All WordNet based predictors were calculated with the WordNet::Similarity package available at <http://wn-similarity.sourceforge.net/>.

### 2.9.3 Experimental Evaluation

#### Predictor-Predictor Correlations

The two collection based predictors are naturally highly correlated as one relies on the average and the other on the maximum of *PMI* scores. Moreover, in instances where a query consists of one or two terms only, both predictors produce exactly the same score. Intuitively, larger differences between the two predictors can be expected for queries derived from TREC description topics. With respect to the different corpora, a clear trend can be discerned: the larger the corpus, the more the correlation between *AvPMI* and *MaxPMI* degrades. While for the queries 301-450 of TREC Vol. 4+5 the correlation reaches  $\tau = 0.80$  ( $r = 0.91$ ), for the queries 701-850 of the GOV2 corpus, the correlation degrades to  $\tau = 0.63$  ( $r = 0.74$ ). The correlations between the WordNet and the corpus based measures are low, yet significant, for the queries of TREC Vol. 4+5 and WT10g; the correlation is close to zero for the queries of the GOV2 corpus. A comparison of the WordNet based measures with each other, yields erratic results, none of them are consistently highly correlated to each other.

#### Predictor Evaluation

Table 2.9 shows the quality of the five algorithms as query effectiveness predictors. *AvPMI* and *MaxPMI* exhibit significant correlations across all collections for the Okapi and Language Modeling approaches, although with respect to the best performing specificity and ambiguity based predictors the correlations are relatively low. The WordNet based predictors have a significant linear correlation for queries 301-350. However, for the same corpus the query set 401-450 leads to negative correlations, which, due to their unreliability, renders these WordNet based predictors unusable, even for the smallest of the evaluated corpora.

The influence of the smoothing parameter  $\mu$  is corpus dependent, but not particularly pronounced. For TREC Vol. 4+5, increasing the amount of smoothing also increases the correlation coefficients. In the case of queries 301-450, for instance, consider  $\tau_{\mu=100} = 0.22$  for *AvPMI*, which, when the level of smoothing is increased, becomes  $\tau_{\mu=2 \times 10^5} = 0.29$ . In contrast, for the WT10g corpus, both *AvPMI* and *MaxPMI* show consistent degradation in correlation with increased smoothing. Lastly, for the GOV2 corpus increasing  $\mu$  leads to slightly increased correlation coefficients. The development of the WordNet based measures is similarly mixed – slight improvements and degradations depending on the query set. However, apart from the query sets of TREC Vol. 4+5, the correlation coefficients are not significantly different from zero.

## 2.10 Significant Results

The previous sections have provided a comprehensive and detailed overview of a number of pre-retrieval predictors. The extensive evaluation that followed each

		AvPMI	MaxPMI	AvPath	AvLesk	AvVP			AvPMI	MaxPMI	AvPath	AvLesk	AvVP
301-350	TEIDF	0.297	0.295	0.253	<b>0.327</b>	0.318	301-350	TEIDF	0.288	0.295	0.022	0.169	0.024
	Okapi	0.314	0.315	0.312	<b>0.413</b>	0.411		Okapi	0.191	0.236	-0.029	0.185	0.059
	$\mu = 500$	0.316	0.298	0.294	0.374	<b>0.411</b>		$\mu = 500$	0.176	0.218	-0.037	0.191	0.039
351-400	TEIDF	<b>0.326</b>	0.196	0.120	0.142	-0.004	351-400	TEIDF	0.221	0.199	0.014	0.141	0.074
	Okapi	<b>0.331</b>	0.203	0.050	0.210	0.192		Okapi	0.247	0.252	0.070	0.202	0.089
	$\mu = 2000$	<b>0.376</b>	0.234	0.005	0.219	0.254		$\mu = 2000$	0.290	0.287	0.054	0.188	0.088
401-450	TEIDF	0.163	0.108	<b>-0.246</b>	-0.127	-0.165	401-450	TEIDF	0.250	0.164	-0.138	-0.144	-0.140
	Okapi	<b>0.401</b>	0.371	-0.247	-0.014	-0.238		Okapi	0.234	0.206	-0.097	-0.025	-0.210
	$\mu = 1000$	<b>0.438</b>	0.398	-0.240	-0.014	-0.214		$\mu = 1000$	0.232	0.195	-0.100	-0.046	-0.219
301-450	TEIDF	<b>0.275</b>	0.230	0.155	0.252	0.170	301-450	TEIDF	0.219	0.219	-0.033	0.062	-0.021
	Okapi	<b>0.336</b>	0.292	0.151	0.278	0.237		Okapi	0.228	0.229	-0.015	0.120	-0.025
	$\mu = 1000$	<b>0.353</b>	0.295	0.135	0.252	0.253		$\mu = 1000$	0.223	0.217	-0.027	0.114	-0.036
451-500	TEIDF	<b>-0.258</b>	-0.224	-0.084	0.076	-0.037	451-500	TEIDF	0.018	-0.037	-0.174	-0.144	-0.176
	Okapi	<b>0.199</b>	0.152	0.004	-0.034	-0.087		Okapi	0.140	0.163	-0.051	-0.062	-0.065
	$\mu = 1000$	<b>0.288</b>	0.199	-0.033	-0.022	-0.073		$\mu = 1000$	0.208	0.213	-0.030	-0.071	-0.027
501-550	TEIDF	-0.150	-0.178	<b>-0.242</b>	-0.133	-0.049	501-550	TEIDF	-0.074	-0.083	-0.230	-0.100	-0.139
	Okapi	0.176	<b>0.292</b>	0.124	0.243	0.005		Okapi	0.191	0.239	0.020	0.103	-0.049
	$\mu = 2000$	0.235	<b>0.403</b>	0.150	0.104	-0.057		$\mu = 2000$	0.212	0.263	0.072	0.045	-0.085
451-550	TEIDF	<b>-0.212</b>	-0.201	-0.151	0.032	-0.032	451-550	TEIDF	-0.039	-0.066	-0.192	-0.123	-0.134
	Okapi	<b>0.196</b>	0.195	0.041	0.001	0.068		Okapi	0.149	0.179	0.001	0.000	-0.045
	$\mu = 1000$	<b>0.285</b>	0.269	0.033	0.006	-0.058		$\mu = 1000$	0.204	0.225	0.029	-0.026	-0.040
701-750	TEIDF	0.250	<b>0.262</b>	0.010	-0.106	-0.059	701-750	TEIDF	0.204	0.205	0.055	0.002	-0.013
	Okapi	0.276	<b>0.333</b>	-0.101	0.066	0.064		Okapi	0.215	0.206	-0.081	0.112	0.114
	$\mu = 1000$	0.431	<b>0.436</b>	-0.118	0.035	0.057		$\mu = 1000$	0.301	0.339	-0.105	0.057	0.053
751-800	TEIDF	-0.044	-0.072	-0.020	-0.069	<b>-0.112</b>	751-800	TEIDF	0.034	0.076	-0.011	-0.036	-0.141
	Okapi	<b>0.425</b>	0.296	-0.116	-0.039	-0.089		Okapi	0.270	0.259	-0.121	-0.078	-0.044
	$\mu = 1000$	<b>0.456</b>	0.353	-0.089	0.019	0.016		$\mu = 1000$	0.314	0.302	-0.117	-0.027	-0.040
801-850	TEIDF	0.661	0.545	0.875	<b>0.916</b>	0.885	801-850	TEIDF	0.164	0.177	0.170	0.174	0.159
	Okapi	0.116	<b>0.188</b>	0.091	0.112	0.127		Okapi	0.078	0.158	0.008	0.010	0.098
	$\mu = 1000$	0.076	<b>0.203</b>	0.097	0.098	0.102		$\mu = 1000$	0.069	0.155	0.035	0.049	0.078
701-850	TEIDF	0.320	0.225	0.481	<b>0.509</b>	0.355	701-850	TEIDF	0.118	0.146	0.077	0.045	0.009
	Okapi	0.247	<b>0.257</b>	0.009	0.064	0.043		Okapi	0.189	0.186	-0.050	0.059	0.077
	$\mu = 1000$	0.277	<b>0.302</b>	0.025	0.069	0.071		$\mu = 1000$	0.215	0.227	-0.046	0.050	0.050

(a) Linear correlation coefficient  $r$ (b) Kendall's  $\tau$ 

Table 2.9: Correlation coefficients of term relatedness based pre-retrieval predictors. In bold, the highest correlation per query set and retrieval approach is shown.

category of predictors aimed to emphasize the strong dependency of the predictors on the retrieval approach, the collection and the particular query set.

What we have largely neglected so far, are a discussion of the significance of the correlations and the comparison of predictor performances across all categories. In this section, we address both issues. Testing the significance of a correlation coefficient can be considered from two angles. On the one hand, we need to test, whether a recorded correlation coefficient is significantly different from zero. While this test is performed and acknowledged in publications, it is usually neglected to test, whether a predictor is significantly different from the best performing predictor. As in retrieval experiments, where we routinely evaluate the significance of the difference between two retrieval approaches, we should do the same in the evaluation of query performance prediction.

In Table 2.10, we summarize the results of the significance tests. For each collection, all predictors, that result in a correlation significantly different from zero for both the linear correlation and Kendall's  $\tau$  coefficient, are listed. Additionally, we report the confidence intervals ( $\alpha = 0.95$ ) of the coefficients. As the retrieval

approach to predict for, Language Modeling with Dirichlet smoothing and the best setting of  $\mu$  was used (Table 2.2). Presented in bold, is the best predictor for each collection. Given the best performing predictor, all other predictors were tested for their statistical difference; underlined are those predictors for which no significant difference ( $\alpha = 0.95$ ) was found.

With respect to the linear correlation coefficient  $r$ , *MaxIDF* is the best predictor for the query sets of TREC Vol. 4+5. When determining the significance of the difference, six other prediction methods show no significant difference, including *AvICTF*, *AvQC* and *MaxVAR*. Thus, apart from the relatedness predictors, all categories provide useful predictors. When considering Kendall's  $\tau$ , *MaxVAR* is the best performing predictor. It is, however, not significantly different from *MaxIDF*. Note, that although *AvVAR* exhibits a higher Kendall's  $\tau$  than *MaxIDF*, its correlation is significantly worse than *MaxVAR*'s correlation. This is due to the fact, that the correlation between *AvVAR* and *MaxVAR* is larger ( $\tau = 0.72$ ) than between *MaxIDF* and *MaxVAR* ( $\tau = 0.53$ ).

For the query sets of WT10g, *MaxVAR* also reports the highest linear correlation with  $r = 0.41$ , but again the significance test shows, most predictors which achieve a significant correlation (different from zero) are not significantly different from the best performing predictor. It is notable that in this corpus, none of the ambiguity based predictors are significantly different from zero. The situation is similar for Kendall's  $\tau$ ; by absolute correlation scores *MaxSCQ* performs best, but apart from *DevIDF* all other predictors exhibit no significantly worse performance. When comparing the confidence intervals of TREC Vol. 4+5 and the WT10g corpus, it becomes apparent that the intervals are wider for WT10g. The reason is, that we only deal with a query set of size 100 in this corpus, whereas TREC Vol. 4+5 are evaluated for 150 queries. Hence, the more queries exist for evaluation purposes, the more reliable the correlation coefficient and thus the smaller the confidence interval.

Finally, Table 2.10 also shows that the GOV2 corpus is easier to predict for than the WT10g corpus; more predictors are significantly different from zero. The most accurate predictor is once more *MaxVAR*, although again, it can be shown that a variety of predictors are similarly useful.

## 2.11 Predictor Robustness

In the beginning of this chapter we stated that, ideally, since the pre-retrieval predictors are search independent, a robust predictor should be indifferent to the particular retrieval algorithm. Since the commonly relied upon retrieval models such as Okapi [125], Language Modeling [76, 97, 121, 170, 171], the Markov Random Field model [104] and the Divergence from Randomness model [5], are based exclusively on term and document frequencies, one might expect similar predictor performances across all of them. However, as seen in the previous sections, prediction methods are indeed sensitive to the retrieval approach as well as the specific parameter settings such as the level of smoothing  $\mu$ .

In the current section, we expand considerably on the variety of retrieval ap-

	<b>r</b>	<b>CI</b>	$\tau$	<b>CI</b>
<i>AvICTF</i>	<u>0.490</u>	[0.358,0.603]	0.266	[0.161,0.371]
<i>AvIDF</i>	<u>0.516</u>	[0.388,0.625]	0.290	[0.186,0.394]
<i>AvPMI</i>	0.352	[0.203,0.485]	0.222	[0.112,0.333]
<i>AvQC</i>	<u>0.457</u>	[0.320,0.575]	0.292	[0.193,0.391]
<i>AvQCG</i>	0.330	[0.179,0.465]	0.297	[0.199,0.395]
<i>AvSCQ</i>	0.256	[0.100,0.400]	0.204	[0.094,0.314]
<i>AvVAR</i>	<u>0.510</u>	[0.381,0.620]	0.356	[0.260,0.452]
<i>DevIDF</i>	0.239	[0.082,0.384]	0.180	[0.066,0.293]
<i>MaxIDF</i>	<b>0.532</b>	[0.407,0.638]	<u>0.339</u>	[0.237,0.442]
<i>MaxPMI</i>	0.295	[0.142,0.435]	0.216	[0.102,0.330]
<i>MaxSCQ</i>	0.341	[0.191,0.475]	0.332	[0.230,0.433]
<i>MaxVAR</i>	<u>0.513</u>	[0.384,0.622]	<b>0.411</b>	[0.316,0.505]
<i>QS</i>	0.407	[0.264,0.533]	0.190	[0.077,0.303]
<i>SCS</i>	<u>0.480</u>	[0.347,0.595]	0.251	[0.145,0.357]
<i>SumVAR</i>	0.297	[0.144,0.437]	0.283	[0.182,0.384]
<i>AvNP</i>	-0.220	[-0.367,-0.062]	-0.145	[-0.261,-0.029]

(a) Queries 301-450 (TREC Vol. 4+5)

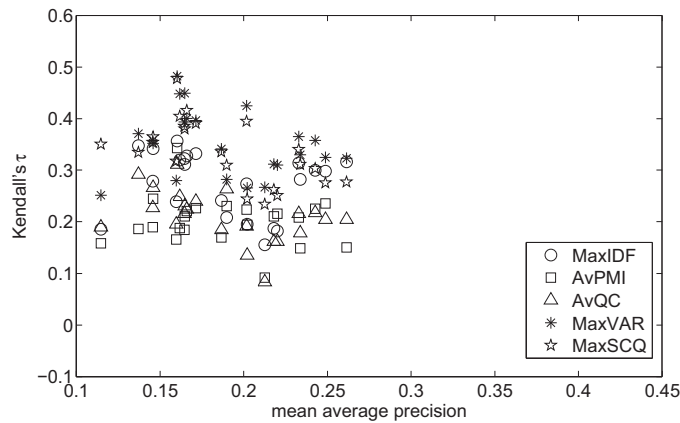
	<b>r</b>	<b>CI</b>	$\tau$	<b>CI</b>
<i>AvPMI</i>	<u>0.282</u>	[0.092,0.458]	<u>0.201</u>	[0.063,0.338]
<i>AvSCQ</i>	<u>0.307</u>	[0.116,0.477]	<u>0.227</u>	[0.098,0.356]
<i>AvVAR</i>	0.292	[0.099,0.463]	<u>0.249</u>	[0.123,0.375]
<i>DevIDF</i>	<u>0.233</u>	[0.037,0.413]	0.154	[0.016,0.292]
<i>MaxIDF</i>	<u>0.292</u>	[0.099,0.463]	<u>0.266</u>	[0.122,0.411]
<i>MaxPMI</i>	<u>0.269</u>	[0.075,0.444]	<u>0.221</u>	[0.096,0.393]
<i>MaxSCQ</i>	<u>0.400</u>	[0.218,0.554]	<b>0.322</b>	[0.195,0.448]
<i>MaxVAR</i>	<b>0.411</b>	[0.231,0.563]	<u>0.321</u>	[0.197,0.445]
<i>SumVAR</i>	<u>0.300</u>	[0.108,0.470]	<u>0.213</u>	[0.084,0.341]

(b) Queries 451-550 (WT10g)

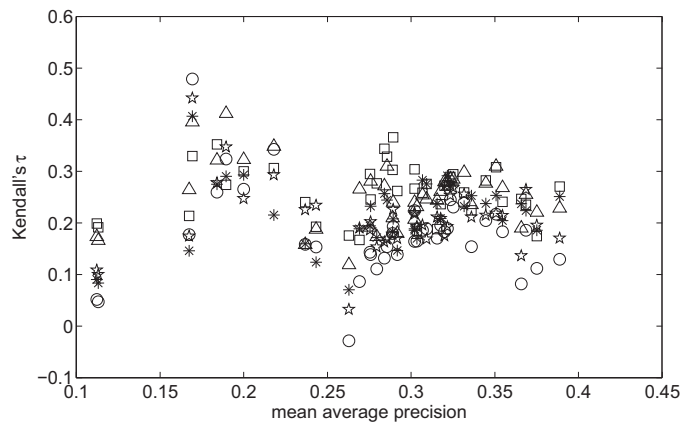
	<b>r</b>	<b>CI</b>	$\tau$	<b>CI</b>
<i>AvICTF</i>	<u>0.281</u>	[0.125,0.423]	<u>0.209</u>	[0.118,0.299]
<i>AvIDF</i>	<u>0.309</u>	[0.155,0.448]	<u>0.229</u>	[0.139,0.320]
<i>AvPMI</i>	<u>0.277</u>	[0.121,0.419]	<u>0.215</u>	[0.113,0.317]
<i>AvQC</i>	<u>0.298</u>	[0.144,0.439]	<u>0.240</u>	[0.154,0.325]
<i>AvSCQ</i>	<u>0.352</u>	[0.202,0.520]	<u>0.234</u>	[0.137,0.331]
<i>AvVAR</i>	<u>0.392</u>	[0.246,0.520]	<u>0.269</u>	[0.179,0.359]
<i>DevIDF</i>	0.185	[0.024,0.336]	0.143	[0.038,0.249]
<i>MaxIDF</i>	<u>0.331</u>	[0.179,0.468]	<u>0.243</u>	[0.147,0.340]
<i>MaxPMI</i>	<u>0.302</u>	[0.148,0.442]	<u>0.227</u>	[0.114,0.341]
<i>MaxSCQ</i>	<u>0.403</u>	[0.259,0.530]	<u>0.274</u>	[0.175,0.374]
<i>MaxVAR</i>	<b>0.412</b>	[0.269,0.538]	<b>0.280</b>	[0.183,0.376]
<i>QS</i>	0.179	[0.019,0.331]	0.137	[0.030,0.245]
<i>SCS</i>	0.248	[0.090,0.394]	<u>0.186</u>	[0.094,0.278]
<i>SumSCQ</i>	0.235	[0.076,0.382]	<u>0.179</u>	[0.061,0.297]
<i>SumVAR</i>	<u>0.337</u>	[0.186,0.472]	<u>0.241</u>	[0.127,0.355]

(c) Queries 701-850 (GOV2)

Table 2.10: Prediction quality of all predictors significantly different from 0 for both  $r$  and  $\tau$ . The best performing predictor for each corpus with respect to  $r$  or  $\tau$  are in bold. Underlined are all predictors that are not significantly different from the best performing predictor.



(a) Title topics 351-400



(b) Title topics 751-800

Figure 2.6: Robustness behavior of pre-retrieval predictors on the automatic TREC title runs.

proaches under investigation. Specifically, we evaluate the predictor performances against retrieval runs submitted to TREC. The runs are restricted to those with a MAP above 0.1 and they are required to be automatic title runs (Appendix B.3.2). Since pre-retrieval predictors rely on the query terms to predict a query's quality, it would be an unfair comparison to include runs based on TREC topic description or narratives. For the same reason, we also exclude manual runs, as they have not necessarily a strong overlap with the query terms. Based on our experimental results in the earlier sections, we selected the five best performing predictors: *MaxIDF*, *AvPMI*, *AvQC*, *MaxVAR* and *MaxSCQ*. Incidentally, this means that a predictor of each category is evaluated. For these predictors, their correlations with all selected TREC runs are determined.

To aid understanding, consider Figure 2.6, where exemplary the results of title topics 351-400 and 751-800 are shown in the form of scatter plots. Each point in a plot indicates a particular correlation coefficient between a TREC run and a pre-retrieval predictor. Therein, the wide spread in predictor performance is clearly visible. For title topics 751-800 (Figure 2.6a) for instance, *MaxIDF* exhibits both the

highest ( $\tau = 0.48$ ) and the lowest ( $\tau = -0.03$ ) correlation, depending on the TREC run. In general, the highest correlations are achieved for rather poorly performing TREC runs, while the predictors' capabilities continuously degrade with increasing retrieval effectiveness of the runs. We speculate that this development is due to the more advanced retrieval approaches of the well performing runs, which do not only rely on term and document frequencies but possibly among others take into account n-grams and the hyperlink structure (for WT10g and GOV2). There are differences between the predictors though. *AvPMI*, for instance, is somewhat less susceptible to the above effects as its performance does not change considerably over the different TREC runs; however compared to other predictors the achieved correlations are lower.

Although not shown, we note that when considering the results over all title topic sets *MaxVAR* and *MaxSCQ* can be considered as the most stable; over most topic sets they exhibit the highest *minimum* correlation with all TREC runs. Note, though, that this stability is relative and the correlation range is still considerable, for example between  $\tau = 0.27$  and  $\tau = 0.49$  for *MaxSCQ* and title topics 401-450. Across all topic sets, the maximum correlation achieved by a predictor and TREC run is  $\tau = 0.50$  (*MaxVAR*, title topics 401-450) and  $r = 0.74$  (*MaxSCQ*, title topics 401-450) respectively. This implies that high correlations can be achieved, if pre-retrieval predictors are used with the “right” retrieval approach.

In our experiments, we determined the predicted scores from stemmed and stopworded indices. As such, if stemming and/or stopword removal did not occur in the submitted TREC runs, the results will not reflect the quality of the predictors accurately. In order to determine the influence of those two preprocessing steps on the reported predictor performances, six indices of TREC Vol. 4+5 were created, each with a different combination of stemming (Krovetz stemmer, Porter stemmer [122] and no stemmer) and stopwording (removal, no removal). Three observations could be made. First of all, the type of stemmer employed, that is Krovetz or Porter, is not of importance, the reported correlations change only slightly in both directions, depending on the prediction method. Secondly, stopword removal across all three stemming options leads to somewhat higher correlations than no stopword removal, though again the changes are minor. Finally, switching off stemming has the largest effect, the correlations degrade across all pre-retrieval predictors to a considerable degree, for instance *MaxVAR* degrades from  $\tau = 0.41$  with Krovetz stemming and stopwording to  $\tau = 0.34$  without stemming and no stopword removal. We conclude that our indices and the pre-retrieval scores derived from them are sufficiently good to draw conclusions about the methods' robustness. One influence that we have not tested though is influence of tokenization. There might still be some differences in performance.

## 2.12 Combining Pre-Retrieval Predictors

Despite the numerous methods proposed, little research has been performed on combining predictors in a principled way. In [174] the proposed predictors are lin-

early combined, and the best performing combination is reported. In this section, we explore if predictor combination with penalized regression methods, which have shown to perform well in analogous prediction scenarios in microarray data analysis [49, 129, 178], lead to better performing prediction methods.

To measure the algorithms' performance we make use of the  $f_{norm}$  setup, which is commonly evaluated by reporting  $r$ . While this approach is reasonable for parameter-free methods, such as the predictors introduced earlier, it is problematic when combining different predictors. Combination methods have a higher degree of freedom and can thus fit the set of predictor/retrieval effectiveness values very well (see Section 2.3.2). Overfitting leads to a high value of  $r$ , while at the same time lowering the prediction accuracy, which is the accuracy of the predictor when predicting values of unseen queries. To avoid this issue, we adopt the methodology applied in machine learning [20] 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.

### 2.12.1 Evaluating Predictor Combinations

Let  $\hat{y}$  be the predictions of  $m$  queries and let  $y$  be the true effectiveness values, then the *RMSE* is given by:

$$RMSE = \sqrt{\frac{1}{m} \sum_i (y_i - \hat{y}_i)^2}. \quad (2.14)$$

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 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 would 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. However, 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, where the model is tuned on  $k - 1$  partitions and the  $k^{th}$  partition functions as test set. This process is repeated for all  $k$  partitions and the overall *RMSE* is reported.

### 2.12.2 Penalized Regression Approaches

Modeling a continuous dependent variable  $y$ , which in our case is a vector of average precision values, as a function of  $p$  independent predictor variables  $x_i$  is referred to as multiple regression. If we also assume a linear relationship between the variables, we refer to it as multiple linear regression. Given the data  $(x^i, y_i)$ ,  $i = 1, 2, \dots, m$  and  $x^i = (x_{i1}, \dots, x_{ip})^T$ , the parameters  $\beta = (\beta_1, \dots, \beta_p)$  of the model  $y = \mathbf{X}\beta + \epsilon$  are to be estimated.  $\mathbf{X}$  is the  $m \times p$  matrix of predictors and  $\epsilon$  is the vector of errors, which are assumed to be normally distributed.

The ordinary least squares (OLS) estimates of  $\beta$  are derived by minimizing the squared error of the residuals:  $\sum_i (y_i - \hat{y}_i)^2$ , where  $\hat{y}_i = \sum_j \beta_j x_{ij}$ . The two drawbacks

of OLS are the low prediction accuracy due to overfitting and the difficulty of model interpretation. 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, thereby only introducing the most important subset of predictors into the model. While this has not yet been explored in the context of query effectiveness prediction, it has received considerable attention among others in microarray data analysis [49, 129, 178] where good results have been 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.

Penalized regression approaches place penalties on the regression coefficients  $\beta$  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) [138] is such a method:

$$LASSO(\hat{\beta}) = \arg \min \left\{ \sum_{i=1}^m (y_i - \sum_{j=1}^p \beta_j x_{ij})^2 \right\} \text{ subject to } \sum_{j=1}^p |\beta_j| \leq t. \quad (2.15)$$

The total weight of the coefficients is restricted by the 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 [179] 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) [54]. LARS determines the full regularization path: in each step, LARS selects the predictor that is most highly correlated with the residuals  $y - \hat{y}$  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 one of the random predictors is picked to enter the model. An alternative is cross-validation: the beta coefficients are learned from  $k - 1$  partitions of the training data and the  $k^{th}$  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 (BOLASSO) [11], 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.

We investigate four variations of these approaches: LARS with traps as stopping criterion (LARS-Traps), LARS with cross-validation to determine the beta coefficients (LARS-CV), BOLASSO and the Elastic Net.

### 2.12.3 Experiments and Results

All predictors described in the previous sections were utilized, with the exception of the WordNet based term relatedness predictors, as they exhibited no predictive power over any of the corpora. As retrieval approach to predict for, we relied on Language Modeling with Dirichlet smoothing and the best performing  $\mu$  (Table 2.2). The parameter settings of the Elastic Net were taken from [179]. LARS-Traps was tested with 6 randomly generated traps while LARS-CV was set up with 10-fold cross validation. BOLASSO was used with 10 bootstrapped samples and each sample was cross-validated to retrieve the best beta coefficient vector.

	TREC Vol. 4+5	WT10g	GOV2
<i>AvPMI</i>	0.207	0.191	0.187
<i>AvQC</i>	0.191	0.196	0.184
<i>AvQL</i>	0.215	0.197	0.194
<i>MaxIDF</i>	0.181	0.187	0.181
<i>MaxSCQ</i>	0.205	0.184	0.178
<i>MaxVAR</i>	0.182	0.184	0.176
<i>AvP</i>	0.214	0.195	0.194

Table 2.11: Performance of selected pre-retrieval predictors given in *RMSE*.

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 with a maximum of 150. To emphasize the cross validation *RMSE* approach being different from  $r/CI$  established on the training set only, we write  $r_{train}$  and  $CI_{train}$ .

In order to provide a comparison between the combination methods and the constituent predictors, for a number of best and worst (*AvQL*, *AvP*) performing predictors, we list their *RMSE* in Table 2.11. In this set of experiments, we summarized all queries of each corpus.

The penalized regression results are reported in Table 2.12 along with  $r_{train}$  and  $CI_{train}$ . For illustration, the predictors selected for LARS-Traps and LARS-CV are shown in the form of histograms in Figure 2.7. The bars indicate in how many of the  $m$  times the algorithm run each predictor was selected to be in the model.

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

Table 2.12: Results of the penalized regression approaches. In bold the improvements over the best single predictor per collection are shown.

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, different conclusions can be drawn. There is a relatively small difference in error of predicted average precision in cases where the correlation coefficients appear quite distinct, e.g.  $r_1 = 0.18$  and  $r_2 = 0.41$  lead to  $RMSE_1 = 0.184$  and  $RMSE_2 = 0.194$  respectively. 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 corpus the error increased.

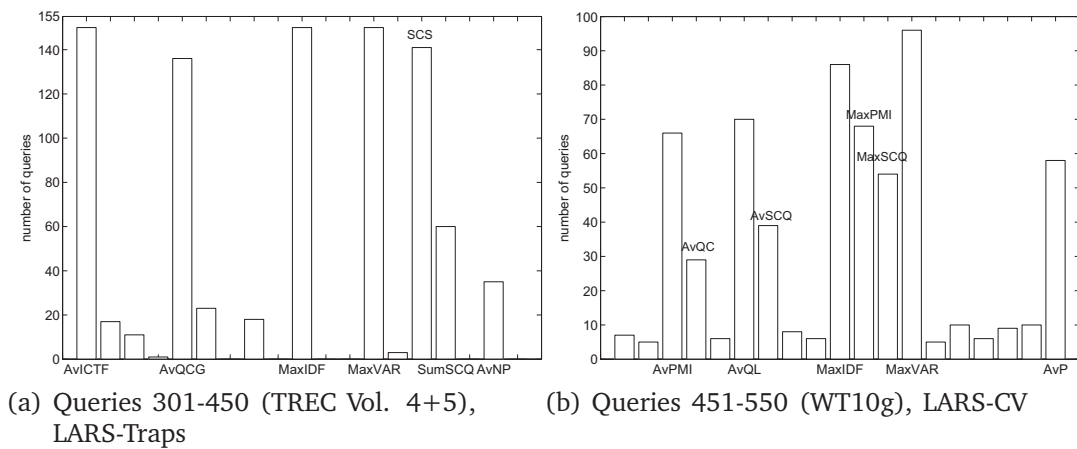


Figure 2.7: Penalized regression: predictor selection.

The histograms show that as desired, not all predictors are selected. For TREC Vol. 4+5 (Figure 2.7a) in most instances the same five predictors appear in the models. Notably is the absence of any term relatedness based predictor. The results are less clear for WT10g (Figure 2.7b): *MaxVAR* appears in most instances, the remaining included predictors fluctuate to a greater degree. The majority of single predictors fail to capture any more variance within the data and so are not used. This is due to two reasons: the poor accuracy of a number predictors which might not be better than random, and, as evident from the scatter plots in Section 2.3.2 (Figure 2.2), the training data is over-represented at the lower end of the average precision values (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 2.7b where a large variety of predictors are added to the model.

## 2.13 Conclusions

This chapter has provided a detailed overview of a large number of pre-retrieval prediction methods. A taxonomy of predictors was introduced and in turn the predictors in each class were analyzed and empirically evaluated. The evaluation was performed on three diverse test collections: a corpus of newswire documents, a

Web corpus and an Intranet-like corpus. We were able to show that the prediction method accuracy – independent of the particular evaluation goal ( $f_{diff}$ ,  $f_{pred}$ ,  $f_{norm}$ ) and thus independent of the particular correlation type used for evaluation – is dependent on the retrieval approach, the collection and the query set under investigation. First, this was shown on three standard retrieval approaches, and later confirmed when we investigated the predictor performances on widely differing TREC runs. For most prediction methods, the Web corpus proved to be the most difficult corpus to predict the effectiveness for.

Moreover, we showed that when using significance tests, significant differences in performance between the prediction methods are difficult to obtain. This is mainly because the query set sizes available to us are so small and thus the correlation coefficients need to have a large difference to point to potential significant improvements. Despite this lack of significant differences, we can use the correlation coefficients and the observations of the predictors performances on diverse TREC runs together to conclude that the ranking sensitivity based *MaxVAR* and the specificity based *MaxSCQ* predictors are overall the best performing predictors: they are among the top performing and they are the most stable across all TREC runs. Term relatedness based predictors on the other hand only perform well on specific query sets and can be considered unreliable. WordNet based measures failed to achieve meaningful predictor accuracies for most query sets.

Experimenting with combining predictors in a principled way through penalized regression which has the advantage of predictor sparseness, lead to reporting the cross-validated *RMSE* as a measure of the prediction accuracy to achieve a more reliable indicator of a method's quality. We showed that under the previous evaluation methodology the combination methods would be considered better in terms of  $r$ , though they are in fact not considerably better than single predictors.