Lang H, Wang B, Jones G *et al.* Query performance prediction for information retrieval based on covering topic score. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 23(4): 590–601 July 2008

Query Performance Prediction for Information Retrieval Based on Covering Topic Score

Hao Lang[1] (郎　皓), Bin Wang[1] (王　斌), Gareth Jones[2], Jin-Tao Li[1] (李锦涛), Fan Ding[1] (丁　凡)
and Yi-Xuan Liu[1] (刘宜轩)

[1]*Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China*

[2]*School of Computing, Dublin City University, Ireland*

E-mail: {langhao,wangbin, jtli, dingfan, liuyixuan}@ict.ac.cn; Gareth.Jones@computing.dcu.ie

Received June 18, 2007; revised March 20, 2008.

Abstract We present a statistical method called Covering Topic Score (CTS) to predict query performance for information retrieval. Estimation is based on how well the topic of a user's query is covered by documents retrieved from a certain retrieval system. Our approach is conceptually simple and intuitive, and can be easily extended to incorporate features beyond bag-of-words such as phrases and proximity of terms. Experiments demonstrate that CTS significantly correlates with query performance in a variety of TREC test collections, and in particular CTS gains more prediction power benefiting from features of phrases and proximity of terms. We compare CTS with previous state-of-the-art methods for query performance prediction including clarity score and robustness score. Our experimental results show that CTS consistently performs better than, or at least as well as, these other methods. In addition to its high effectiveness, CTS is also shown to have very low computational complexity, meaning that it can be practical for real applications.

Keywords information storage and retrieval, information search and retrieval, query performance prediction, covering topic score

1 Introduction

A typical information retrieval (IR) scenario comprises a collection of documents and an IR system which retrieves documents in response to a user's query. A user submitting a query to the IR system has his/her information need in mind. He/she can judge the returned documents according to their relevance to this information need. Query performance prediction is an attempt to quantify the quality of results returned by an IR system for a specific query *without* any relevance information from the user. Recently predicting query performance has been recognized as an important capability for IR systems and is attracting growing interest within the IR research community[1,2].

Ideally, a system which is able to predict query performance can adapt parameters or change algorithms to suit the query. For example, some researchers[3,4] utilized prediction of query performance to improve retrieval through deciding automatically whether to use the phase of query expansion. This first step illustrates how predicting query performance can improve an IR system's performance by performing query-specific processing.

Besides its impact on IR systems, predicting query performance can provide valuable feedback for users, e.g., reporting confidence scores for results and possibly asking the user to revise the query. It can also inform the system administrator regarding topics that are of increasing interest to users, but not answered well by the system[1].

In this paper, we propose a statistical method called Covering Topic Score (CTS) to predict query performance. CTS measures how well the topic of the user's query is covered by retrieved documents to do the estimation. The idea of predicting query performance by studying to what extent retrieved documents cover the topic of the query is inspired by our observations, which we present in detail below. If a retrieved document set has high occurrences of all the query terms, it is quite likely to be a successful search. In that case, all the topic concepts are covered well by the retrieved documents, and each query term is deemed to indicate one topic concept of the query respectively. Queries of low

Regular Paper
This work is supported by the National Natural Science Foundation of China under Grant No. 60603094 and the National Grand Fundamental Research 973 Program of China under Grant No. 2004CB318109.

performance are those with most query terms seldom appearing in returned documents or most query terms remarkably appearing except some primary terms of great importance. In this type of situation, most or at least some of the topic concepts of the query are missed, resulting in an unsuccessful search.

For example, consider a query "lyme disease arthritis", which is the Title field of TREC topic 604. In this query, the three query terms are supposed to represent three independent topic concepts and all the three comprise the information request of the user. All the three query terms appear observably together in top returned documents. Hence all the topic concepts of the query are well covered, promising a high average precision of that query. Consider another query "profiling motorists police", the Title field of TREC topic 432. In this query, many irrelevant documents retrieved by the IR system contain words "motorist" and "police", but they all miss the very important word "profile" and thus one main concept of the query is missed by the result.

CTS tries to predict query performance based on how well the topic of the user's query is covered by retrieved documents. For a submitted query, CTS first analyzes the query and decomposes it into separate topic concepts with different weights indicating different importance of concepts. CTS finds out whether the returned documents cover all these topic concepts well. In the following section, we will describe CTS in full detail.

The main advantage of CTS is its high speed and effectiveness. Since CTS only needs to calculate the relative frequencies of query terms in the returned data. set, it is extremely easy and fast to compute, which means that it is practical for utilization in real applications. More importantly, experiments demonstrate CTS significantly correlates with query performance in a variety of TREC test collections, and consistently performs better than, or at least as well as, existing state-of-the-art methods for query performance prediction, including clarity score and robustness score. In particular, CTS is an intuitive and general model that can be easily extended to incorporate features beyond bag-of-words, such as phrases and proximity of terms, which results in more predictive power for the model.

The rest of the paper is organized as follows. Section 2 describes some related work. Section 3 describes our approach for predicting query performance in full detail. Section 4 describes the experiments we conducted on TREC data. Section 5 analyzes the efficiency of our proposed predictors. Finally Section 6 concludes the paper and presents some directions for future work.

2 Related Work

Prediction of query performance has been attracting growing interest within the IR research community, and it has been carried out under different names such as query difficulty or query ambiguity. The problem of predicting query performance was proposed because many IR systems suffer a significant variation in performance across queries. Given this observation, it is desirable that IR systems can identify those queries of low performance in order to handle them properly[1,2].

The Reliable Information Access (RIA) workshop[5] investigated the reasons for system variance in performance. By performing failure analysis on TREC topics, 10 failure categories were identified. Five categories of the ten relate to the systems' failure to identify all aspects of the topic. Our work follows this direction by predicting query performance on the basis of how well the topic of a user's query is covered by retrieved documents.

Query performance prediction is a challenging task, as shown in [1, 2]. In the Robust Track[2], systems were asked to rank the topics by predicted performance in order to do topic-specific processing. Performance of prediction was measured by comparing the ranking of topics based on their actual precision to the ranking based on their predicted precision. Many methods were tried in this track and as the track report claimed[2] predicting query performance is intrinsically difficult.

Existing methods of predicting query performance can be divided into two categories: pre-retrieval predictors and post-retrieval predictors. Pre-retrieval predictors utilize the static information of a query, which can be computed before retrieval; besides static information, post-retrieval predictors also utilize the dynamic information, which can only be computed after retrieval.

As shown in Fig.1, static information and dynamic information are the two main kinds of features used for performing the prediction. Static information is the natural property of the query and can be computed before retrieval, e.g., statistic characteristics of query terms (IDF etc.) in the collection and linguistic characteristics of a query; dynamic information captures the main characteristics of the search result and can only be computed after retrieval, e.g., a query term's distribution in the returned documents and interrelationships within them etc. Static information is usually easy and fast to compute by using term statistics of index; dynamic information is more reliable to be used in query performance estimation due to making good use of search results, but it often involves complex computations to obtain the features.

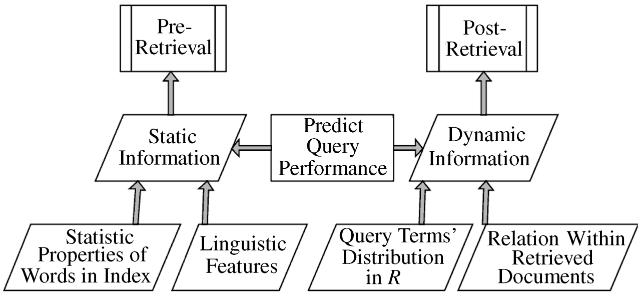
Pre-Retrieval
Post-Retrieval
Static Information
Predict Query Performance
Dynamic Information
Statistic Properties of Words in Index
Linguistic Features
Query Terms' Distribution in R
Relation Within Retrieved Documents

Fig.1. Static and dynamic information of a query to predict query performance.

Category I: *Pre-Retrieval Predictors*

Pre-retrieval predictors attempt to utilize query borne properties to predict query performance. These query properties typically include specificity and generality. The resolution power of the query to discriminate the relevant documents from other documents for the request is the most significant query property, and has been explored extensively in query performance estimation. It has been observed that this seems to be associated with statistical characteristics of query terms within the document index file and linguistic characteristics of a query.

He and Ounis[6] suggest that the standard deviation of the inverse document frequency (idf) of the query terms correlates positively with average precision. They observe that the distribution of the informative amount in query terms can affect the retrieval performance. They suggest good retrieval performance be correlated with high variation of query term idf. Plachouras *et al.*[7] studied a predictor based on the average inverse collection term frequency (avICTF) of the query terms. They hypothesized that the performance of a query can be reflected by the average qualities of its composing terms. The above two predictors are easy and fast to compute by utilizing query term statistics of index. Hence they are useful for practical applications.

Considering linguistic characteristics of a query, Mothe and Tanguy[8] analyzed the correlation between 16 different linguistic features of TREC queries and the average precision scores. Each of these features can be viewed as a clue to a linguistically specific characteristic, either morphological, syntactical or semantic. Two of these features (syntactic links span and polysemy value) show a significant impact on effectiveness of searching. Swen *et al.*[9] also tried to use the total ambiguity of query terms to estimate query performance. They used a sense distribution dictionary constructed from WordNet and a sense-tagged Brown corpus to estimate the ambiguity of a single query word. Although the correlation values are not very high, these predictors indicate a promising link between some linguistic characteristics and query performance.

Category II: *Post-Retrieval Predictors*

Post-retrieval predictors try to analyze characteristics of top-retrieved documents which are supposed to be relevant to the query by the IR system. The clarity score method[10] belongs to this category, and was one of first successful methods used for quantifying query performance. The clarity score measures the degree of dissimilarity between the language usage associated with the query and the generic language of the collection. Experiments in [10] report that clarity score significantly correlates with query performance in a variety of TREC test collections. Thus the clarity score is considered one of the state-of-the-art methods. Amati *et al.*[11] propose the DFR scoring model by using the KL-divergence between a query term's frequency in the top retrieved documents and the frequency in the whole collection, which is very similar to the definition of the clarity score.

Zhou *et al.*[12] introduced the notion of ranking robustness, which refers to a property of a ranked list of documents that indicates how stable the ranking is in the presence of uncertainty in the ranked documents. A statistical measure called the robustness score is proposed to quantify that notion and is used for the estimation. Ranking robustness provides us a novel framework to predict query performance. Robustness score shows the highest correlation with the average precision in a variety of TREC test collections, to our best knowledge.

Vinay *et al.*[13] studied the clustering tendency of retrieved documents based on the "cluster hypothesis"[14] and proposed four methods of predicting query performance. These methods primarily attempt to study whether the retrieved documents are a random set of points or a clustered set of points. The most effective measure is that of the sensitivity to document perturbation, which is somewhat similar to the robustness score.

Carmel *et al.*[15] attempted to answer the question of what makes a query difficult and discussed the problem thoroughly. They found that the distance measured by the Jensen-Shannon divergence between the retrieved document set and the collection significantly correlated with average precision.

Yom-Tov *et al.*[3] proposed a learning estimator based on the agreement between results of the full query and results of its sub-queries. If the query terms agree on most of the returned documents, high performance of the query can be expected. The main advantage of the estimator is its simplicity and that it can be computed efficiently during query execution. Our proposed method CTS shares closely related insights, which perform the estimation based on the contribution of each query term to the final retrieved documents. Despite the high level similarity, the details of the methods differ greatly, with our method allowing more general features such as phrases and proximity of terms to measure the topic concept in a conceptually simple and intuitive statistical model.

Pre-retrieval predictors are the most desirable since they tend to easily and efficiently compute, just using query term statistics of the document index to do simple computation. Hence, they are very suitable for real IR applications. However, they do not make use of any information taken from the search results, so they cannot predict the quality of results perfectly. All existing pre-retrieval predictors only achieve limited success. Post-retrieval predictors examine the search results in detail to capture their important characteristics, which brings in more prediction power. Our goal is to propose a high speed query performance predictor which is more effective than, or at least as well as, these state-of-the-art predictors. By stressing both static and dynamic information simultaneously and integrating the right features in a simple manner, CTS is able to increase effectiveness and efficiency at the same time.

3 Measure Topic Coverage Density

In this work, the Covering Topic Score (CTS) method is proposed to measure how well retrieved documents cover the topic of a user's query. In a classical IR scenario, a user submitting a query to the IR system has her information need in her mind, and she can then judge the retrieved documents according to their relevance to this information need. Information need can be comprised of several separate topic concepts. To what extent retrieved documents cover those topic concepts can reflect the user satisfaction with the search result. We can hypothesize that if the search result covers all the topic concepts well, it is quite likely to be a successful search, and satisfy the user's information need.

The idea of predicting query performance by measuring CTS is inspired by two considerations. Firstly, CTS concerns the basic processing of IR, e.g., trying to understand the user's query and decomposing it into separate topic concepts, as well as judging how well the search result covers these topic concepts with the goal of eventually being able to predict query performance. Secondly, we observe that often for a successful search, the returned document set has high occurrences of all the query terms; for an unsuccessful search, most or some query terms seldom appear in the top result. We hypothesize that high occurrence of one query term indicates one topic concept is well covered; the distribution of query terms in the returned documents can be used to perform the estimation of query effectiveness. We examine this hypothesis thoroughly in the next section.

Now we describe our approach to computing the CTS in detail. For a newly submitted query Q, we decompose it into a set of topic concepts $TC = \{tc_1, tc_2, tc_3, \ldots, tc_n\}$. For each topic concept tc_i in TC, we set a weight $IM(tc_i)$ indicating variance of importance across topic concepts. Next we compute the covering degree CD of the retrieved documents for all these topic concepts. We define $CD(tc_i)$ as the degree of the search result covering the topic concept of tc_i. Finally, we compute the CTS by adding covering degree $CD(tc_i)$ multiplied by the weight $IM(tc_i)$ for each topic concept tc_i in TC. The whole process is illustrated in Fig.2.

$$CTS = \sum_{tc_i \in TC} IM(tc_i) \times CD(tc_i). \tag{1}$$

As shown in (1), after query Q is decomposed into a set of topic concept tc_i in TC, we need to compute the importance $IM(tc_i)$ and the covering degree $CD(tc_i)$ respectively. The first part relates to the property of the query itself. The latter one correlates with the statistical characteristics of retrieved documents. We describe ways to measure those two parts in the next subsection.

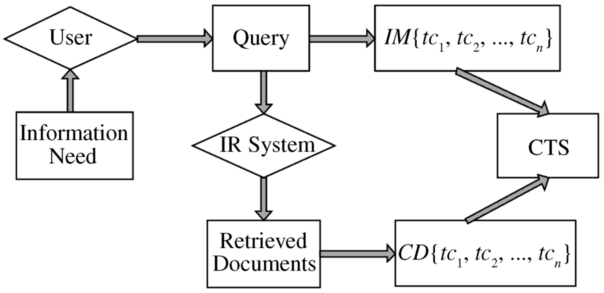

Fig.2. Covering topic score calculation.

3.1 Measure Importance of Topic Concept

In the long history of developing sophisticated retrieval models for improving performance in IR, words in the documents and queries are considered to be independent with each other. In this subsection, we follow this standard hypothesis. Each term of the query is supposed to be independent from other terms and represents one topic concept. Let us return to the

TREC topic 604 "lyme disease arthritis" examined earlier. The three query terms indicate three separate topic concepts, all of which compose the information request of the user. Up to now, every query term is thought to be one topic concept, and (1) can be transferred to (2) as follows:

$$CTS = \sum_{q \in Q} IM(q) \times CD(q) \tag{2}$$

where q is a query term of query Q.

In this work, we propose two statistical methods to quantify the importance of a query term, principally utilizing term statistics of the document index. One method named $IM1$ uses the inverse document frequency (IDF) to measure the importance. We suppose that terms that are of great importance and semantic significance tend to occur in a smaller fraction of a collection than other terms. $IM1$ is defined as

$$IM1(q) = \frac{1}{DF(q)} \tag{3}$$

where $DF(q)$ is document frequency of term q in the collection.

The other statistical method named $IM2$ follows two criteria. First, important terms have lower frequency than other terms in the collection. Second, important terms tend to have variant frequency in different document of the collection. In other words, words of no importance often have stable distributions in the collection. Since mean and variance are two important measures of a variable, we use the combination of the two to compute $IM2$.

$$IM2(q) = \frac{VP(q)}{MP(q)} \tag{4}$$

where $MP(q)$ is the mean of the probability of term q in an individual document and $VP(q)$ is the variance of the probability of term q. These are defined as follows:

$$MP(q) = \frac{\sum_{1 \leqslant i \leqslant C} P_i(q)}{C} \tag{5}$$

where C is the number of documents in the collection and each document is denoted as D_i. $P_i(q)$ is probability of term q in D_i which is its frequency in document D_i divided by the total number of terms in D_i.

$$VP(q) = \sqrt{\frac{\sum_{1 \leqslant i \leqslant C}(P_i(q) - MP(q))^2}{C-1}}. \tag{6}$$

Both $IM1$ and $IM2$ use the *static* information attempting to grasp the property of query terms, and hence they can be computed before retrieval.

3.2 Measure Covering Degree of Topic Concept

In this subsection, we analyze the retrieved documents for a given query to measure the covering degree CD for each topic concept. Under the assumption of independence of query terms, we prefer to calculate query terms' frequencies in the top N returned documents to quantify the covering degree. If a query term has high occurrences in the search result, the CD of that term is considered to be high; by contrast, if a query term seldom appears in the top returned documents, the CD tends to be low. Formally, we define covering degree as follows:

$$CD(q) = \frac{\sum_{D \in R} P_D(q) \times W(D)}{N} \tag{7}$$

where D is a document, R is the returned dataset, N is the size of R, $W(D)$ is the weight of document D and $P_D(q)$ is probability of term q in D which is its frequency in document D divided by the total number of terms in D.

Documents in the returned dataset are not totally independent of each other. They are returned by the IR system in descending order of relevance to a given query. Hence, top ranked documents tend to be the right answers and show more statistical significance to calculate the covering degree values. And other documents in the returned dataset tend not to be relevant to the query and bring in more noise. Hence, we should attach different weights to the returned documents. Top ranked documents should weigh more than low ranked documents. In this paper, we explore two different weighting schemes as follows for this purpose. Experimental results in Section 4 show the effectiveness of this weighting strategy.

One weighting strategy named $W1$ weights top N returned documents the same as a baseline approach, which is defined as:

$$W1(D) = 1. \tag{8}$$

The other strategy named $W2$ weights a document D by the probability of D given query Q and is defined as:

$$W2(D) = P(D|Q). \tag{9}$$

(9) can be transformed to (10) by Bayesian inversion:

$$W2(D) = P(D|Q) \propto P(Q|D)P(D). \tag{10}$$

We compute $P(Q|D)$ by estimating the likelihood of an individual document model generating the query[16] as:

$$P(Q|D) = \prod_{q \in Q} P(q|D). \tag{11}$$

We suppose longer documents tend to offer more statistic information. Hence, here longer document is set higher prior probability and defined as:

$$P(D) = \frac{|D|}{|D| + \delta} \tag{12}$$

where $|D|$ is the length of the document D, δ is set to be 10 000.

We estimate $P(q|D)$ by relative frequencies of terms linearly smoothed with collection frequencies as:

$$P(q|D) = \lambda P_{\rm ml}(q|D) + (1 - \lambda) P_{\rm coll}(q) \tag{13}$$

where $P_{\rm ml}(q|D)$ is the relative frequency of term q in document D, $P_{\rm coll}(q)$ is the relative frequency of term q in the collection as a whole, and λ is set to be 0.9 in this work.

Covering degree using $W1$ is named $CD1$ while covering degree using $W2$ is named $CD2$. CD captures the main characteristics of returned document set. They belong to the dynamic information. In this way, our method integrates both static and dynamic information as main components in a simple manner.

3.3 Combing Features Beyond Bag-of-Words

Recently features beyond bag-of-words, such as phrases and proximity of terms, have been shown to be able to improve the performance of IR systems[17,18]. However, existing research on prediction of query performance has not examined the usefulness of features beyond the bag-of-words, as far as we know. We would like to see whether these features can lead to improved query performance prediction results.

In the Web context, users tend to supply short queries, thus it is obviously important to make the most out of what little information these short queries give us. A phrase match between a document and a query is usually an indication that the document deals with topic concept of the query described by the phrase.

In this work, we treat each term-pair with the same order from the query as a phrase. Let us return to the query "lyme disease arthritis", we suppose "lyme disease" and "disease arthritis" as well as "lyme arthritis" are all phrases contained in the query. When we count the frequency of a phrase in an individual document, the two terms of the term-pair need to co-occur contiguously in the document, where the two terms appear ordered within a window of 1 term.

For us, proximity of terms is a natural extension of phrases where the restriction on the nearness of the terms is somewhat more relaxed[17,18]. The only difference concerning phrases and proximity of terms in this paper is in the process of counting frequencies in an individual document. For proximity of terms, terms of term-pair could appear ordered or unordered within a window of terms. In our experiment, we set N to 8.

Since *CTS* is an intuitive and general model, features beyond the bag-of-words can be easily incorporated. Just like a query term, each phrase or proximity of terms is considered to indicate one topic concept, and their importance and covering degree can be computed in the same way described in the above subsections. Thus we can compute the *CTS* for phrases and proximity of terms respectively, defined as follows:

$$CTS_{\rm phrase} = \sum_{tc_i \in TC_{\rm phrase}} IM(tc_i) \times CD(tc_i), \tag{14}$$

$$CTS_{\rm proximity} = \sum_{tc_i \in TC_{\rm proximity}} IM(tc_i) \times CD(tc_i). \tag{15}$$

In general, we denominate *CTS* based on query terms and phrases as well as proximity of terms as $CTS_{\rm term}$, $CTS_{\rm phrase}$, $CTS_{\rm proximity}$ respectively. As we have the models for each kind of feature separately, we could consider general ways of combining them. In this work, the combination of each two models is commonly handed through interpolation.

$$CTS1 = \lambda CTS_{\rm term} + (1 - \lambda) CTS_{\rm phrase}, \tag{16}$$

$$CTS2 = \lambda CTS_{\rm term} + (1 - \lambda) CTS_{\rm proximity}, \tag{17}$$

where λ is a weighting parameter between 0 and 1. $CTS1$ is the covering topic score based on features of terms and phrases of the query while *CTS2* is based on features of terms and proximity of terms.

4 Evaluation

In this section, we describe the experiments that were performed to evaluate the effectiveness of *CTS* to predict query performance. The result of prediction was compared to previous state-of-the-art methods including clarity score and robustness score. Query performance is measured by average precision. In the main, we seek to answer the following four questions.

1) Is *CTS* effective enough to predict query performance? What is the contribution to the performance of *CTS* for different factors such as ways of measuring importance and covering degree? We will measure the correlation between results of *CTS* and average precision scores in a variety of TREC test collections.

2) Is *CTS* robust enough to perform this prediction? The size of result set N is the main parameter of the model *CTS*. We examine the influence of N in the performance of *CTS*.

3) Does features beyond bag-of-words result in more prediction power? Is *CTS* able to gain better results when extended to incorporate features like phrases and proximity of terms?

4) Does *CTS* have high significance compared to other state-of-the-art methods regarding effectiveness and efficiency?

4.1 Experimental Setup

Our experiments were performed using a variety of TREC test collections. All queries used in our experiments are the Title field of TREC topics (usually a few words), except for TREC-4 where the description field is used due to a lack of Title field. Table 1 gives the summary of these collections. The retrieval was done with the query likelihood model[16] using Dirichlet smoothing (Dirichlet prior μ is set to 1000).

Table 1. Summary of Test Collections

TREC	Collection	Topic Number	Number of Document
1+2+3	Disk 1+2	51∼150	741 856
4	Disk 2+3	201∼250	567 529
5	Disk 2+4	251∼300	524 939
Robust2004	Disk 4+5 minus CR	301∼450; 601∼700[1]	528 155
Terabyte2004	GOV2	701∼750	25 205 197
Terabyte2005	GOV2	751∼800	25 205 197

We used the top 10 retrieved documents to calculate covering degree defined in (7), unless otherwise stated. Figs.3 and 4 show that the performance of *CTS* is insensitive to the size of result set and it is less time-consuming to use only 10 documents. We computed variations of *CTS* only based on the features of bag-of-words, except for in Subsection 4.4 where we examined the prediction power of *CTS* combining features of phrases and proximity of terms.

4.2 Correlation with Average Precision

We measure the correlation between *CTS* and average precision by Kendall's rank correlation test. Kendall's rank correlation is used for measuring the degree of correspondence between two rankings. The values of it range between −1.0 and 1.0 where −1.0 means perfect negative correlation and 1.0 means perfect positive correlation.

In this paper, we prefer to use Kendall's rank correlation to compare the ranking of queries by average precision to the ranking by *CTS* of these queries, because lots of previous research of prediction[3,12,13] utilize this non-parametric test, which makes our results of prediction comparable to those results of previous research.

The results for correlation with average precision are shown in Table 2, where bold cases mean the results are statistically significant at the 0.05 level. As described in the previous section, *CTS* is mainly comprised of two parts, namely importance of topic concepts (*IM*) and covering degree of topic concepts (*CD*). Here we try several variations of *CTS* using different measures of *IM* and *CD*. The first column of Table 2 refers to the way to measure importance of a topic concept, while the second column refers to the way to measure covering degree. When *CD* is set to const, it means that the factor of covering degree is ignored, and we set the covering degree of all the topic concepts for each query to be 1. In this case, *CTS* only uses the static information

Table 2. Kendall's Rank Correlation Coefficient for Correlation Between Average Precision and Variations of CTS with Different Ways of Measuring Importance and Covering Degree

IM	CD	TREC123	TREC4	TREC5	Robust2004	Terabyte2004	Terabyte2005
*IM*1	const	**0.2885**	**0.2180**	0.1037	**0.3327**	**0.2636**	**0.2506**
*IM*1	*CD*1	**0.4048**	**0.3796**	**0.3322**	**0.4001**	**0.3027**	**0.2882**
*IM*1	*CD*2	**0.4036**	**0.5004**	**0.3306**	**0.4260**	**0.2908**	**0.3012**
*IM*2	const	**0.1782**	**0.2473**	0.0988	**0.3331**	**0.2704**	**0.2833**
*IM*2	*CD*1	**0.3980**	**0.5624**	**0.4776**	**0.4355**	**0.3112**	**0.3371**
*IM*2	*CD*2	**0.3814**	**0.6147**	**0.4220**	**0.4543**	**0.3129**	**0.3551**
const	*CD*1	0.1014	**0.3029**	0.1102	0.0778	0.0374	0.0971
const	*CD*2	0.0776	**0.3029**	**0.1951**	**0.1281**	0.1003	0.0955

[1] Topic 672 is removed because it has no relevant documents.

and becomes a pre-retrieval predictor. When *IM* is set to const, it means that we weigh all the topic concepts the same.

From these results, we firstly observe that *CTS* significantly correlates with average precision in a variety of TREC test collections.

Secondly, variations of *CTS* using *IM*2 have better results than using *IM*1 most of the time. This indicates *IM*2 is a more sophisticated method of quantifying the importance of topic concept compared to *IM*1.

Thirdly, predictors only utilizing the static information ($CD=$ const) cannot predict query performance well, and powerful predictors of query performance have to integrate both static and dynamic information simultaneously.

Fourthly, variations of *CTS* weighing all the topic concepts the same ($IM=$ const) lose their prediction power significantly. It demonstrates that it is necessary to weigh the topic concepts based on their importance.

Finally, it is observed that variations of *CTS* using *CT*2 have better results than using *CT*1 in 4 of the 6 datasets (datasets of TREC4, Robust2004, Terabyte2004 and Terabyte2005). It indicates that setting the returned documents to different weights can increase the prediction power in some times but not in all the times. We attribute the exception to the following cause: *CT*2 cannot weigh the documents in those collections properly. Hence a more sophisticated method of weighting documents has to be developed. Concerning that *CTS* using *CD*1 has consistent and robust performance in different collections, while *CD*1 is simple and easy to compute, *CD*1 could be an alternative to the more sophisticated ones.

4.3 Influence of Parameters on Predicting

CTS has one main parameter, the size of result set N. Obviously when more documents are used, more statistical information can be utilized to do the estimation. However, as N increases, the newly joined documents tend to be irrelevant for queries in both high and low performances and they have little contribution to estimation. Hence, whether the performance relies on N is an important issue.

Figs.3 and 4 show the influence of N on *CTS* using *IM*1 and *IM*2 respectively (both using *CD*1). The curves shown in these figures have their highest values between 10 and 30. However, the curves are all quite flat. This shows that the performance of *CTS* is insensitive to N, and is robust and easy to implement in real applications, and does not require parameter tuning.

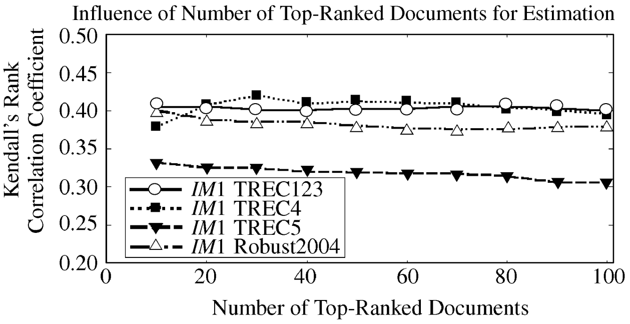

Fig.3. Influence of result size on estimation, *IM*1.

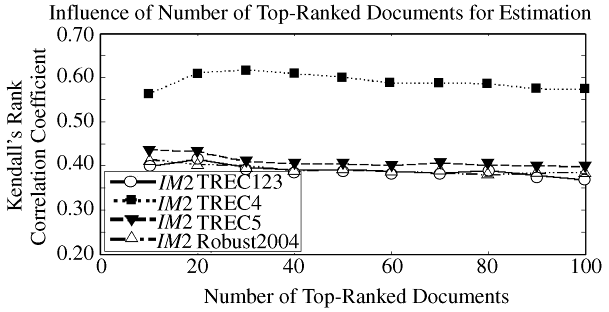

Fig.4. Influence of result size on estimation, *IM*2.

4.4 Power of Features Beyond Bag-of-Words

In this subsection, we examine the prediction power of *CTS* combining features beyond the bag-of-words. We compare the performance of *CTS*1 and *CTS*2 defined in (16) and (17), respectively, to that of $CTS_{\rm term}$ only using topic concepts of query terms in the data collection of Robust2004. For the case (*IM*1, *CD*1) in Table 3, we adopt the parameter λ that yields the highest value of Kendall's rank correlation coefficient by linear search. For all the other cases in Table 3, we use the same parameter trained in the case (*IM*1, *CD*1). In fact, experimental results demonstrate that the optimal combination weight λ changes little across our tested cases.

As shown in Table 3, *CTS*1 and *CTS*2 have better results than $CTS_{\rm term}$ for variances of *CTS*, especially for models using *IM*1. This is a clear indication by which phrases and proximity of terms can be useful in improving the performance of query prediction. It also demonstrates that *CTS* can benefit from those features easily as *CTS* is quite a general model.

4.5 Significance of *CTS*

In order to better assess the significance of *CTS*, we compare *CTS* with previous state-of-the-art methods including clarity score and robustness score. Robustness score is reported to have the highest correlation with the average precision in a variety of TREC test collections, to our best knowledge[12]. In order to ensure the comparability, we use the same retrieval model and the same data collection when computing predicting scores.

Table 3. Kendall's Rank Correlation Coefficient for Correlation Between Average Precision and $CTS_{\rm term}$, $CTS1$ as Well as $CTS2$ with Different Ways of Measuring Importance and Covering Degree in the Data Collection of Robust2004

IM	*CD*	$CTS_{\rm term}$	$CTS1$	$CTS2$
*IM*1	*CD*1	**0.4001**	**0.4304 (+7.58%)**	**0.4444 (+11.07%)**
*IM*1	*CD*2	**0.4260**	**0.4415 (+3.64%)**	**0.4468 (+4.88%)**
*IM*2	*CD*1	**0.4355**	**0.4407 (+1.19%)**	**0.4375 (+0.46%)**
*IM*2	*CD*2	**0.4543**	**0.4567 (+0.53%)**	**0.4583 (+0.88%)**

We compare *CTS*(*IM*2, *CD*2) with clarity score and robustness score using the data collection described in Table 1. As shown in Table 4, *CTS* has a stronger correlation with average precision compared to clarity score and robustness score. We point out that we only used the reported results[12] here because we cannot get the same results as reported when we redid the experiments.

Table 4. Comparison of Kendall's Rank Correlation Coefficient for Several Estimation Methods

TREC	Clarity Score	Robust Score	*CTS*(*IM*2, *CD*2)
123	**0.331**	**0.329**	**0.381(+15.1% +15.8%)**
4	**0.353**	**0.548**	**0.615(+74.2% +12.2%)**
5	**0.311**	**0.328**	**0.422(+35.7% +28.7%)**
Robust2004	**0.412**	**0.392**	**0.454(+10.2% +15.8%)**
Terabyte2004	0.134	**0.213**	**0.313(+136% +47.0%)**
Terabyte2005	0.171	**0.208**	**0.355(+107% +70.7%)**

We use the Title field of TREC topics (usually a few words) as queries in our experiments in order to model what real users do in the Web context (e.g., users submit short queries to search engines). The prediction power shown in Table 4 gives *CTS* significant potential for use in real applications.

To validate whether *CTS* outperforms other statistical methods significantly[2], we use a significance test borrowed from [19]. Here, query difficulty prediction problem can be regarded as a binary decision problem on every two queries Q_i and Q_j. For one algorithm, if the order of two prediction scores is the same as the order of the AP (average precision) values, a successful decision is made, otherwise a wrong decision is made.

Let $X_i \in \{0, 1\}$ be the measure of success for algorithm X on the i-th decision, where 1 means correct and 0 means incorrect. $Y_i \in \{0, 1\}$ is the measure of success for algorithm Y on the i-th decision. It is obvious that if the number of total queries is M, the number of total decisions should be $M(M-1)/2$. Let n be the number of times where X_i differs from Y_i, k be the number of times where Y_i is bigger than X_i. The null hypothesis is that k has a binomial distribution[3] of $Bin(n, p)$ where $p = 0.5$, while the alternative hypothesis is that k has a binomial distribution of $Bin(n, p)$ where $p > 0.5$, meaning algorithm Y is better than X. If $n > 12$, the P-value can be computed using the standard normal distribution for

$$Z = \frac{k - 0.5n}{0.5\sqrt{n}}. \tag{18}$$

If the P-value is low enough, we can reject the null hypothesis, i.e., algorithm Y is better than X.

We used the above hypothesis testing to compare *CTS* with clarity score and robustness score algorithms. The P-level results are listed in Table 5.

Table 5. Significance Test Results

TREC	CTS vs. Clarity Score	CTS vs. Robust Score
123	$< \mathbf{10^{-20}}$	$< \mathbf{10^{-20}}$
4	$< \mathbf{10^{-20}}$	$\mathbf{4.65 \times 10^{-14}}$
5	$\mathbf{9.07 \times 10^{-12}}$	$< \mathbf{10^{-20}}$
Robust2004	$< \mathbf{10^{-20}}$	**0.006**
Terabyte2004	$< \mathbf{10^{-20}}$	$< \mathbf{10^{-20}}$
Terabyte2005	$< \mathbf{10^{-20}}$	$\mathbf{2.89 \times 10^{-10}}$

From Table 5, we can see clearly that *CTS* outperforms clarity score and robust score in statistics significantly.

In addition, we did another significance test from [20] (see p.154). Surprisingly, the results are almost the same as Table 5. We also did random tests to enhance our conclusion. From the prediction sequence by algorithm X, we randomly exchange two elements and compute the Kendall's tau of the new sequence. The

[2] Because we cannot get the same results as previous work, we only use the results in our experiments to do significance test. In fact, though most of our repeated results are not so high as reported, their differences are not so great. So we believe the significance test results should be reliable.

[3] The events that X_i differs Y_i are not strictly independent with each other. But when the number n is big enough, the events can be regarded as approximately independent. In our experiments, the smallest n is bigger than 300.

process repeats 10 000 times. Then we plot the distribution figure and find the point of the Kendall's tau of algorithm Y. One of the figures is as follows[4].

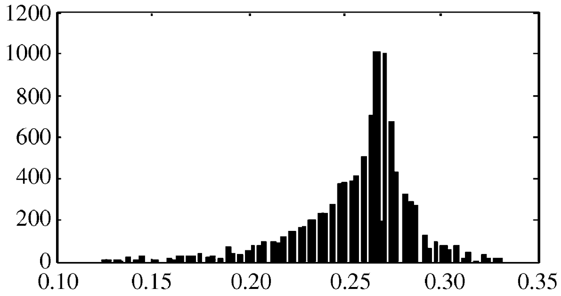

Fig.5. Simulated distribution of Kendall's tau for robust score algorithm using Terabyte2004.

The point corresponding to 95% percentile is 0.291 which is lower than 0.313 (see Table 4). Thus we can say *CTS* is statistically better than Robust Score algorithm from Fig.5. If we randomly exchange more elements, the mean value will be closer to zero. We can get the above more confidently. Using other test collections, we also got similar results and the same conclusion.

5 Efficiency

Besides the effectiveness comparison shown in Subsection 4.5, we try to analyze the computational complexity of the predictors, which is one of the main factors used for evaluating the performance of an algorithm. In this work, we use the time complexity to measure the efficiency for a given algorithm, which represents the scale of floating point operations for the same scale of inputs.

The efficiency of *CTS* computation with (1) is dominated by the estimation of the covering degree of topic concepts by (7), since importance of query terms can be precomputed at index-time as static information. As a result, *CTS* only needs to calculate the relative frequencies of query terms in the returned dataset in practice, which makes it extremely easy and fast to compute.

Table 6 shows the time complexity of clarity score and robustness score, as well as *CTS*. In Table 6, m is the length of the query, N is size of returned dataset, V is the size of vocabulary in the whole collection, and K is the number of circulation in robustness score. The efficiency of clarity score computation is dominated by the estimation of the query language model[10]. In fact, it needs to compute probabilities of all the unique terms in the whole vocabulary among the returned dataset. Thus, the time complexity of clarity score is $O(VN)$. The efficiency of robustness score computation is dominated by the estimation of document models in V dimension for the top N documents retrieved and times K of sampling documents from the document models, as well as ranking the sampled documents given the retrieval model. The time complexity of robustness score is $O(VNK)$. For *CTS*, we only need to calculate the relative frequencies of query terms in the returned data set. The time complexity of *CTS* is $O(mN)$.

Table 6. Comparison of Time Complexity

Estimation Method	Time Complexity
Clarity Score	$O(VN)$
Robustness Score	$O(VNK)$
CTS	$O(mN)$

We point out that both clarity score and robustness score are calculated in high V dimension while V is the number of unique terms in the whole collection. V tends to be an extremely large number (e.g., V is 630 035 in dataset of Robust2004). For *CTS*, the predictor performs the prediction in low m dimension, calculating the relative frequencies of query terms. m is the length of a query, which is often comprised of a few terms.

At the same time, *CTS* needs a small proportion of returned documents to perform the prediction (N is set to 10), while N of clarity score is 500 and N of robustness score is 50. Hence, for *CTS*, it takes less time to access documents and less storage space when processing the documents. As a result, *CTS* has a low level of computational complexity, compared to clarity score and robustness score.

6 Conclusion and Future Research

We have proposed a conceptually simple and intuitive statistical method named Covering Topic Score (CTS) to predict query performance. It is based on how well the topic of the user's query is covered by retrieved documents from a certain retrieval system. Models obtained from different sources, e.g., query terms, phrases, proximity of terms, are kept separate, and as a result, our model is easy to understand and expand. Experiments demonstrate that *CTS* significantly correlates with query performance in a variety of TREC test collections, and in particular *CTS* gains more prediction power benefiting from features of phrases and proximity

[4] Here we use the results of Terybyte2004, our robust score algorithm got 0.269 of Kendall's tau, which is higher than reported results in Table 4.

of terms. We compare *CTS* with previous state-of-the-art methods and our experimental results show that *CTS* consistently performs better than, or at least as well as, other methods. Furthermore, *CTS* is demonstrated to have very low computational complexity, given significant potential for use in real applications.

There are still open issues related to this research. First of all, we can individualize and optimize the combination of features beyond the bag-of-words through a learning phase like the hidden Markov training algorithm. Secondly, in this work, we use words and richer proximity features to represent topics. We plan to develop a more sophisticated concept-based model to define the topic by using linguistic and data mining techniques. Thirdly, our general model can be used to measure the quality of feedback documents and allow some flexibility in combining the query model prior to feedback information, which is discussed in [21]. Finally, we plan to extend our model to make it adaptive in real Web context. In the Web environment, there are more features which can be used to perform the prediction, such as document structure information (containing many different fields like title, head, body etc.) and link information, as well as URL information. Due to the simplicity of our model, we believe these new features can be easily incorporated.

Acknowledgements We thank Jian Zhang from Purdue University for his help about statistics. We also thank all anonymous reviewers for their kind help to improve the paper.

References

[1] Carmel D, Yom-Tov E, Soboroff I. Predicting query difficulty. In *Proc. SIGIR Workshop*, Salvador, Brazil, 2005, http://www.haifa.ibm.com/sigir05-qp/index.html.

[2] Voorhees E M. Overview of the TREC 2004 robust track. In *the Online Proceeding of 2004 Text Retrieval Conference* (*TREC 2004*).

[3] Yom-Tov E, Fine S, Carmel D, Darlow A. Learning to estimate query difficulty: Including applications to missing content detection and distributed information retrieval. In *Proc. the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Salvador, Brazil, 2005, pp.512–519.

[4] Cronen-Townsend S, Zhou Y, Croft B. Precision prediction based on ranked list coherence. *Information Retrieval,* 2006, 9(6): 723–755.

[5] Harman D, Buckley C. The NRRC reliable information access (RIA) workshop. In *Proc. the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Sheffield, United Kingdom, 2004, pp.528–529.

[6] He B, Ounis I. Inferring query performance using pre-retrieval predictors. In *Proc. the SPIRE 2004,* Padova, Italy, 2004, pp.43–54.

[7] Plachouras V, He B, Ounis I. University of Glasgow at TREC2004: Experiments in web, robust, and terabyte tracks with terrier. In *the Online Proc. 2004 Text Retrieval Conference* (*TREC 2004*).

[8] Mothe J, Tanguy L. Linguistic features to predict query difficulty. In *Proc. ACM SIGIR 2005 Workshop on Predicting Query Difficulty-Methods and Applications,* 2005.

[9] Swen B, Lu X-Q, Zan H-Y, Su Q, Lai Z-G, Xiang K, Hu J-H. Part-of-speech sense matrix model experiments in the TREC 2004 robust track at ICL, PKU. In *the Online Proceeding of 2004 Text Retrieval Conference* (*TREC 2004*).

[10] Cronen-Townsend S, Zhou Y, Croft W B. Predicting query performance. In *Proc. the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Tampere, Finland, 2002, pp.299–306.

[11] Amati G, Carpineto C, Romano G. Query difficulty, robustness and selective application of query expansion. In *Proc. the 25th European Conference on Information Retrieval,* Sunderland, Great Britain, 2004, pp.127–137.

[12] Zhou Y, Croft W B. Ranking robustness: A novel framework to predict query performance. In *Proc. the 15th ACM International Conference on Information and Knowledge Management.* Virginia, USA, 2006, pp.567–574.

[13] Vinay V, Cox I J, Milic-Frayling N, Wood K. On ranking the effectiveness of searches. In *Proc. the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Seattle, USA, 2006, pp.398–404.

[14] C J van Rijsbergen. Information Retrieval. Second Edition, London: Butterworths, 1979.

[15] Carmel D, Yom-Tov E, Darlow A, Pelleg D. What makes a query difficult? In *Proc. the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Seattle, USA, 2006, pp.390–397.

[16] Song F, Croft W B. A general language model for information retrieval. In *Proc. the 18th ACM International Conference on Information and Knowledge Management,* Kansas City, USA, 1999, pp.316–321.

[17] D Metzler, W Bruce Croft. A Markov random field model for term dependencies. In *Proc. the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Salvador, Brazil, 2005, pp.472–479.

[18] G Mishne, M de Rijke. Boosting web retrieval through query operations. In *Proc. the 27th European Conference on Information Retrieval,* pp.502–516.

[19] Yang Y, Liu X. A re-examination of text categorization methods. In *Proc. the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Berkeley, California, USA, 1999, pp.42–49.

[20] Wasserman L. All of Statistics: A Concise Course in Statistical Inference. Springer Press, 2004.

[21] Tao T, Zhai C. Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proc. the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval,* Seattle, USA, 2006, pp.162–169.

Hao Lang received the B.A. degree in computer science and engineering from Jilin University in 2004. Since 2004, he has been a post graduate student in Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include algorithm design and query performance prediction in information retrieval.

Bin Wang is currently an associate professor in Institute of Computing Technology, Chinese Academy of Sciences, China. He received a computer science degree in 1993 and then accomplished his master's degree in 1996, Wuhan University. He obtained his Ph.D. degree in 1999 from Institute of Computing Technology, Chinese Academy of Sciences. His research interests include information retrieval and natural language processing. He is a senior member of China Computer Federation, a member of IEEE and Chinese Information Processing Society, an editorial member of Journal of Chinese Information Processing.

Gareth Jones is a senior lecturer in School of Computing, Dublin City University, Ireland. He received the Ph.D. degree in electrical and electronic engineering from the University of Bristol in 1994. His current research interests are in the areas of Information Access and Affective Computing. He is on the program committee of the following events in 2007: SAC2007, ECIR2007, WAC2007, SIGIR2007, CIKM2007, ICTIR2007, CIICT2007 and IDEAL2007 etc.

Jin-Tao Li is a professor in Institute of Computing Technology, Chinese Academy of Sciences, China, where he was the director of the Digital Laboratory, and is currently the deputy director of the institute and the director of the Center for Advanced Computing Research. He received the Ph.D. degree in computer science from the institute in 1989. His current research interests are in the areas of digital image processing, video coding, multimedia content analysis and retrieval, digital watermarking, network computing technology. Dr. Li is a member of IEEE and a senior member of China Computer Federation.

Fan Ding received the B.A. degree in computer science and engineering from Beijing Information Technology Institute in 2002. Since 2002, he has been a post graduate student in Institute of Computing Technology, Chinese Academy of Sciences. His current research interests include algorithm design and query relationship in information retrieval.

Yi-Xuan Liu received the B.A. degree in computer science and engineering from Nankai University in 2006. Since 2006, she has been a post graduate student in Institute of Computing Technology, Chinese Academy of Sciences. Her current research interests include query performance prediction in information retrieval and new word recognition.