

Modeling Search Response Time *

Dan Zhang

Department of Computer Science
Purdue University, West Lafayette, IN, US, 47906
zhang168@cs.purdue.edu

Luo Si

Department of Computer Science
Purdue University, West Lafayette, IN, US, 47906
lsi@cs.purdue.edu

ABSTRACT

Modeling the response time of search engines is an important task for many applications such as resource selection in federated text search. Limited research has been conducted to address this task. Prior research calculated the search response time of all queries in the same way either with the average response time of several sample queries or with a single probability distribution, which is irrelevant to the characteristics of queries. However, the search response time may vary a lot for different types of queries. This paper proposes a novel query-specific and source-specific approach to model search response time. Some training data is acquired by measuring the search response time of some sample queries from a search engine. Then, a query-specific model is estimated with the training data and their corresponding response times by utilizing Ridge Regression. The obtained model can be used to predict search response times for new queries. A set of empirical studies are conducted to show the effectiveness of the proposed method.

Categories and Subject Descriptors

H3.3 [Information Search and Retrieval]

General Terms

Performance, Experimentation, Theory.

Keywords

Source Selection, Response Time, Ridge Regression.

1. INTRODUCTION

The rapid growth of online searchable information sources on local area networks and the Internet creates a problem of finding the relevant information that may be distributed among many information sources [1][7].

Much literature has studied issues related to source selection and results merging. In source selection [2][3][5][6], we need to determine the search engines that should be queried, especially under the circumstance when it is time consuming or expensive to query every data source due to some resource constraints. Once the results are retrieved from

*This work is supported by NSF research grant IIS-0746830.

different search engines, a results merging method is often used to integrate the individual ranked lists into a single list [5].

Searching response time is a very important component in source selection [2][5] [6]. However, the related research is still limited. In [2], [5] and [6], the authors assume that the response times for the different queries should be the same or follow a specific distribution that is independent of the queries. In fact, the search response time may vary significantly for different kinds of queries with different query lengths and common or rare query terms in different databases. For example, if we search “cancer” in the NIH Award database¹, the response time would be much longer than that of “p53”, which is the name of a gene that causes cancer, because, in this database, “cancer” is much more common than “p53”.

In this paper, we propose a new method to model the relationship between the query and the corresponding response time. Our method predicts the time delay of different queries by: i) sending some training queries to a search engine and measuring the corresponding response times; and ii) modeling the response times given the training queries, and predicting response times for new queries sent by the users. This is accomplished by formalizing the response time predication procedure as an optimization problem and predict the response time of new queries by this optimization result.

2. THE PROPOSED METHOD

Our method is characterized by its ability to model the relationship between the response times and queries. To achieve this goal, we first need to extract features for queries.

Suppose we have a training query set $\{q_1, q_2, \dots, q_l\}$, where l is the total number of training queries. We map each training query q_i to a feature vector \mathbf{x}_i , which is calculated on a set of background databases with diverse topics. For the training query q_i , we use the sum of its corresponding idf features for its query terms to represent its map on the j -th background database. And we use this value as its j th feature in \mathbf{x}_i . More specifically, the j -th component in the feature vector \mathbf{x}_i is determined by:

$$\mathbf{x}_{i,j} = \sum_{q_{ik} \in q_i} \log \left(\frac{N_j}{df_j(q_{ik}) + 1} \right) \quad (1)$$

Here, N_j is the total number of documents in the j th background database, q_{ik} is the k -th query term in the query q_i , and $df_j(q_{ik})$ is the total number of documents, in the j th

¹http://crisp.cit.nih.gov/crisp/crisp_query.generate_screen

background database, that contains this particular term q_{ik} . We add this number by 1 to avoid the denominator being zero.

After the training feature vectors $\mathcal{X} = \{\mathbf{x}_i, i = 1, \dots, l\}$ are acquired, we build a linear model based on these vectors and their corresponding response times in a specific search engine, i.e., $\mathcal{T} = \{t_i, i = 1, \dots, l\}$. This model can be represented by a function f such that: $f : \mathcal{X} \Rightarrow \mathcal{T}$ and $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{b}$, where \mathbf{w} is the weight vector and b is the bias. To obtain the \mathbf{w} and b in this model, we use the Ridge Regression method [4]. The optimization formulation of Ridge Regression can be described as:

$$\min_w \sum_{i=1}^l (t_i - \mathbf{w}^T \mathbf{x}_i - b)^2 + \lambda \|\mathbf{w}\|^2 \quad (2)$$

where λ is a parameter that controls the trade-off between the complexity of the model and the square loss. After solving this optimization problem, we can predict the response time for any query sent to this search engine by: i) first mapping this query to a feature vector \mathbf{x} by Eq.(1). ii) estimating its response time by $\hat{t} = f(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + \mathbf{b}$.

3. EXPERIMENTS

Experiments are conducted on 8 different randomly selected academic search engines, including the IEEE explorer, the NIH AWARD Search Engine, the accessscience search engine, etc.

We have a set of 120 queries, whose topics mainly focus on computer science, biology, economics, etc. These queries are sent to the 8 academic search engines beforehand and we can acquire their corresponding response times in these search engines. The feature vectors of these 120 queries are calculated on 22 background databases. 20 of them are from the 20 newsgroup database and the remaining 2 are from the trec wt10g and the OHSUMED databases, respectively.

For the proposed method, the parameter λ in Eq.(2) is determined by 5-fold cross validation. We also compare the proposed method with some query independent methods, i.e., the ‘‘Average’’ method, which uses the mean of the response times on the training set as the predicted response time for the testing queries; the ‘‘Min’’ method, which uses the minimal response time of the training queries as the prediction for the testing queries, and the ‘‘Max’’ method, which uses the maximal response time of the training queries as the estimation for testing queries. For all of these methods, on each search engine, the final results are averaged over 30 independent trials. During each trial, 50 queries are randomly selected as training queries, while the others are left as the testing set \mathcal{E} . The following criterion is used to measure the loss on \mathcal{E} :

$$Loss = \frac{1}{|\mathcal{E}|} \sum_{i \in \mathcal{E}} \left(\frac{\hat{T}_i - T_i}{T_i} \right)^2 \quad (3)$$

where \hat{T}_i and T_i denote the estimated and actual response time of the i th query on the testing set \mathcal{E} , respectively. $|\mathcal{E}|$ represents the number of queries in \mathcal{E} . The loss comparison results are shown in Table 1. From these experimental results, we can see that the proposed method is superior.

	S1	S2	S3	S4
Proposed	0.783	0.085	0.138	0.041
Average	0.851	0.286	0.205	0.056
Min	0.809	0.139	0.199	0.082
Max	1.798	115.375	5.896	3.773
	S5	S6	S7	S8
Proposed	0.354	0.137	0.035	0.124
Average	0.658	0.260	0.043	0.161
Min	0.468	0.184	0.072	0.282
Max	13.608	22.700	2.699	2.168

Table 1: Response Time Comparison

4. CONCLUSIONS

Response time is a very important component in source selection. However, previous research only calculated the search response time of all queries in the same way either with the average response time of several sample queries or with a single probability distribution, which is query independent and can not reflect the characteristics of the queries. In this paper, we show, by modeling the relationship between queries and the corresponding response times, we can predict the response time for new queries more precisely. It is true that this model may be affected by some unknown factors, such as the network congestion and web caching. In the future, we plan to design new methods to make the prediction model more robust to these situations.

5. REFERENCES

- [1] J. Callan. Distributed information retrieval. *Advances in Information Retrieval*, pages 127–150, 2000.
- [2] D. Dreilinger and A. Howe. Experiences with selecting search engines using metasearch. *ACM Transactions on Information Systems (TOIS)*, 15(3):195–222, 1997.
- [3] J. French, A. Powell, and J. Callan. Effective and Efficient Automatic Database Selection. 1999.
- [4] A. Hoerl and R. Kennard. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *TECHNOMETRICS*, 42(1):80–86, 2000.
- [5] K. Hosanagar. A utility theoretic approach to determining optimal wait times in distributed information retrieval. In *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 91–97. ACM New York, NY, USA, 2005.
- [6] A. Montgomery, K. Hosanagar, R. Krishnan, and K. Clay. Designing a Better Shopbot. *Management Science*, 50(2):189–206, 2004.
- [7] C. Yu, K. Liu, W. Meng, Z. Wu, and N. Rishe. A Methodology to Retrieve Text Documents from Multiple Databases. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pages 1347–1361, 2002.