# Long-Term Learning for Web Search Engines

Charles Kemp and Kotagiri Ramamohanarao

Department of Computer Science & Software Engineering
The University of Melbourne, Australia
{cskemp,rao}@cs.mu.oz.au

**Abstract.** This paper considers how web search engines can learn from the successful searches recorded in their user logs. Document Transformation is a feasible approach that uses these logs to improve document representations. Existing test collections do not allow an adequate investigation of Document Transformation, but we show how a rigorous evaluation of this method can be carried out using the referer logs kept by web servers. We also describe a new strategy for Document Transformation that is suitable for long-term incremental learning. Our experiments show that Document Transformation improves retrieval performance over a medium sized collection of webpages. Commercial search engines may be able to achieve similar improvements by incorporating this approach.

## 1 Introduction

Internet search engines are collecting thousands of user histories each day that could be exploited by machine learning techniques. To give a simple example, suppose that 70% of users who type the query 'cola' into Excite choose the third link presented on the results page. This statistic provides strong evidence that the third link should be promoted to the top of the list next time someone enters the same query.

Embedded within each user history is a set of semantic judgements. Long term research in Information Retrieval is directed towards the goal of giving machines the ability to make these judgements on their own. In the short term, however, progress is more easily achieved by taking advantage of the judgements users are already making as they interact with a collection.

Direct Hit (`www.directhit.com`) is the only current search engine that claims to adapt to these judgements. The owners of this system claims that it learns by monitoring which sites searchers select from the results page, and how much time the searchers spend at these sites [9]. As far as we are aware, however, the algorithms used by Direct Hit have never been published. We believe that algorithms for exploiting user histories are too valuable to be left to a single company, and that the best strategies will only be found if the research community develops an interest in this area.

A small-scale IR system might use many strategies for learning from its user histories. Over the past few decades, most of the popular approaches to machine

T. Elomaa et al. (Eds.): PKDD, LNAI 2431, pp. 263–274, 2002.

learning have been applied to Information Retrieval. [15] and [8] both provide surveys of this work, but to give just a few examples, neural networks [3,14], genetic algorithms [12], probabilistic models [11,17] and decision trees [8] have all been tried. [4] used clickthrough hierarchies to rank the documents.

None of these approaches, however, is an option for a large-scale system. Web search engines are already straining against the limitations of speed imposed by current technology. The additional burden imposed by any viable learning strategy must be small indeed.

Document Transformation is one strategy that is simple enough to become part of a large-scale search engine. It involves a modification of the space of document representations so that documents are brought closer to the queries to which they are relevant. Although proposed as early as 1971, Document Transformation has received little attention, and has never been adequately tested. [19, p 326] identifies the main reason for this neglect:

> Document-space modification methods are difficult to evaluate in the laboratory, where no users are available to dynamically control the space modifications by submitting queries.

The development of the Internet has solved Salton's problem. Bringing users into the laboratory is still a problem, but all of a sudden it has become possible to bring the laboratory to the users. Millions of users are submitting millions of queries to web search engines every day. The user log of any of these search engines would provide ample data for a large-scale investigation of Document Transformation.

This paper, then, has two main goals. First, we hope to show that Document Transformation can improve web search engines. Second, Document Transformation is a research area in its own right, and we hope to use data from web search engines to evaluate it more rigorously than ever before.

## 2 The Vector Space Model

The algorithms for Document Transformation are built on top of the vector space model of Information Retrieval. An overview of this model will be given before Document Transformation is described in more detail.

The vector space model represents each document and query as a vector of concept weights, and computes the similarity between a document and a query by measuring the closeness of their corresponding vectors [19]. In the simplest case, every term is a separate concept. The similarity $S_{d,q}$ between document $d$ and query $q$ can therefore be calculated as $S_{d,q} = \sum_{t \in d \cap q} w_{d,t} \cdot w_{q,t}$ where $w_{d,t}$ and $w_{q,t}$ are the weights of term $t$ in document $d$ and query $q$. Given a query $q$, a ranked keyword search involves computing $S_{d,q}$ for all documents $d$ in the collection, and returning the top-ranked documents to the user.

Many schemes for calculating $w_{d,t}$ and $w_{q,t}$ have been tried [24]. Most of these express the weight of a term in a document as a product of three factors:

**Table 1.** The BD-ACI-BCA similarity measure. $W_d$ is the Euclidean length of the normalised document vector; $W^a$ is the average value of $W_d$ across the collection; $f^m$ is the greatest value of $f_t$ over the collection, and $s$ is a constant with a typical value of 0.7 [22]

| FACTOR | $w_{d,t}$ | $w_{q,t}$ |
|---|---|---|
| TF | $1 + \log_e f_{d,t}$ | $1 + \log_e f_{q,t}$ |
| IDF | 1 | $\log_e(1 + f^m/f_t)$ |
| IDL | $1/((1-s) + s \cdot W_d/W^a)$ | 1 |

$w_{d,t} = TF \times IDF \times IDL$. The TF or 'Term Frequency' component is a function monotonic in $f_{d,t}$, the frequency of $t$ in $d$. The IDF or 'Inverse Document Frequency' is monotonic in $1/f_t$, where $f_t$ is the frequency of $t$ across the entire collection. The IDL or 'Inverse Document Length' is monotonic in $1/W_d$, the reciprocal of the length of document $d$.

[24] found that none of the versions of $S_{d,q}$ that they tested consistently outperformed the others. The formulation they label BD-ACI-BCA, however, performed well across a range of measures, and gave the best overall performance for the precision at 20 metric. This formulation is described in Table 1, and will be used in the experiments described later.

## 3 Document Transformation

The hope underlying the vector space model is that a document vector will end up close to the vectors representing queries relevant to that document. Document Transformation aims to fix up cases where this goal is not achieved. Given a query and a document thought to be relevant to that query, Document Transformation is the process of moving the document vector towards the query vector.

Document Transformation was described in the late 1960s by the SMART team [5,10]. It is a close relative of Relevance Feedback, the process of refining a query vector by moving it towards documents identified as relevant in the hope that the move will also bring it closer to relevant documents that have not yet been identified [16,18]. One important difference between the two strategies is that Document Transformation alone leaves permanent effects on a system.

Several formulae for Document Transformation have been tested. One version (labelled DT1 for later reference) is:

$$D_{i+1} = (1-\alpha)D_i + \alpha\frac{|D_i|}{|Q|}Q$$

where
$D_i$ = the vector for the ith iteration of the document

$Q$ = a query known to be relevant to the document
$|Q|$ = the 1-norm of $Q$, that is, the sum of all weights in $Q$, and
$\alpha$ is an experimental parameter

Document Transformation has been shown to improve retrieval performance over small and medium-sized collections [5,20]. There is no clear victor among the strategies that have been tried: different strategies perform best on different test collections [20].

## 4 The Test Collection

The TREC collections have become the clear first choice for researchers considering an experiment in information retrieval. Every TREC collection comes with a set of test queries, and a set of relevance judgements identifying the documents in the collection that are relevant to the test queries. Our experiments, however, required a set of user histories recording interactions with a collection. There are no standard test collections for which user histories are freely available, and we therefore decided to create our own.

The best source of user histories would be a log kept by a major search engine. Although we did not have access to one of these logs, we realised that user histories involving pages on servers at the University of Melbourne could be reconstructed from the logs kept locally. The collection chosen was therefore a set of 6644 web pages spidered from the Faculty of Engineering website at the University of Melbourne (`www.ecr.mu.oz.au`). User histories describing interactions with this collection were taken from the 'referer log' (sic)[1] kept by the Engineering web server.

A referer log contains information about transitions users have made between pages. Suppose that a user clicks on a link which takes her from one page to another. The HTTP standard allows the user's browser to tell the second server the URL of the first page, and this information can be stored in the referer log kept by the second server.

Transitions between the results page of a search engine and a page at the University of Melbourne are the only transitions relevant to this study. These transitions will be called 'clickthroughs,' and one example is:

```
http://www.google.com/search?hl=en&safe=off&q=
history+of+baritone+saxophone →
/~samuelrh/mushist.html
```

This clickthrough indicates that somebody found the page `www.ecr.mu.oz.au/~samuelrh/mushist.html` by searching for 'history of baritone saxophone' on Google. A simple script was written to extract queries from clickthroughs.

[1] The spelling error has become part of the HTTP standard.

### 4.1 Generating the Collection

The 6644 pages in the collection were spidered on September 20, 2001. All of the pages were passed through a HTML to ASCII converter. As far as we are aware, this is the largest collection that has ever been used for an investigation of Document Transformation.

### 4.2 The Test Set

Our test set was created by extracting queries from the 100 most recent clickthroughs as of September 28. Two queries were later removed from the test set, since none of the systems tested returned any relevant documents in response to these queries. Duplicate queries were not removed from the test set: there are eight queries among the 98, for example, that refer to 'Run DMC.'

Previous applications of machine learning to information retrieval have often suffered from inadequate test sets. For most test collections, relevance judgements are only provided for a small number of queries, and these queries are carefully chosen to overlap only slightly if at all. Redundancy can be achieved by including some, but not all of the training queries in the test set, but even this approach is not entirely satisfactory. Part of the redundancy should be due to queries that are related, but not identical. For example, 'Bach organ music' and 'organ works j s bach' are a pair of similar queries that are not identical. Generating such pairs would be a difficult task.

A better approach is to take samples of training and testing data directly from the domain under consideration. If these samples are independent, then the training and test sets should automatically contain just the right amount and type of redundancy. We have followed this approach by taking our training and test sets directly from the collection of authentic queries stored in the referer log.

### 4.3 The Training Set

Clickthroughs were collected over a seven week period beginning on September 6. After removing 100 clickthroughs to create the test set, a little over 4000 were left. We took the first 4000 of these for our training set.

Our training data is noisy, since a clickthrough does not necessarily indicate that a page is relevant to a query. As far as we know, all previous studies have used human-generated relevance judgements as a basis for Document Transformation. Previous studies have also used training sets much smaller than 4000 queries. The largest training set used by [5] or [20] contained only 125 queries. For both of these reasons, our work models the realities faced by web search engines more accurately than any previous study of Document Transformation.

### 4.4 Performance Measure

The need to generate relevance judgements for this collection and test set ruled out most of the standard metrics for assessing the performance of an IR system. We used the precision(10) metric: the average number of relevant pages

**Table 2.** Three strategies for Document Transformation. $Q$ is a query relevant to document $D$. $|D|$ is the 1-norm of $D$: that is, the sum of the term weights in $D$

| Strategy | Formula |
|---|---|
| DT1 | $D = (1-\alpha)D + \alpha\frac{\|D\|}{\|Q\|}Q$ |
| DT2 | $D = D + \beta Q$ |
| DT3 | $D = D_{original} + D_{learned}$, where $D_{learned} = \begin{cases} D_{learned} + \beta Q & \text{if } \|D_{learned}\| < l \\ (1-\alpha)D_{learned} + \alpha\frac{\|D_{learned}\|}{\|Q\|}Q & \text{otherwise} \end{cases}$ |

among the top ten documents returned. [1] have argued that precision(10) is an appropriate metric for web search engines, since users rarely proceed past the first page of results. This metric, however, is less stable than most of the other standard metrics: if the test set is small, performance assessed using this metric may not accurately predict performance for other collections [7].

### 4.5 Relevance Judgements

Relevance judgements were made for the top ten documents returned by each system. The results for all experimental runs were shuffled before making these judgements. That is, when deciding whether a document was relevant to a query, we did not know whether the document had been returned by the control method, or a system that had supposedly learned, or both.

## 5 Systems Compared

A control system was implemented to provide a baseline for comparison. The control system has no access to the referer log, and carries out a ranked keyword search using the BD-ACI-BCA similarity measure. Other similarity measures were tried, but none led to superior results.

The remaining systems perform Document Transformation using clickthroughs from the training set. Document Transformation can be implemented in many ways: the three tested here are given in Table 2.

Strategy DT1 ensures that the length of a document vector remains constant, where the length of a vector is defined as the sum of its weights. [5] and [20] both describe experiments where DT1 was found to improve retrieval performance.

Strategy DT2 is the simplest of the three: the terms in $Q$ are weighted by $\beta$, then added to $D$. This strategy allows the length of a document vector to grow without bound.

Both of these strategies are susceptible to saturation of the document vectors. If enough clickthroughs are associated with a document, the effect of the terms

found in the original document becomes negligible. Previous studies of Document Transformation have not discussed this problem, probably because the training sets they use are small.

Document vector saturation, however, is a real problem for a search engine attempting to learn from thousands of queries each day. Strategy DT3 is designed to rule out this problem. Each document vector is the sum of two parts: the original part, and the learned part. The original part is the original document vector, and remains unchanged throughout the experiment. The learned part starts out as the zero vector, and is built up using a mix of strategies DT1 and DT2. DT2 is used until the vector has attained length $l$. After this point, DT1 is used and the length of the learned part remains constant.

This parameter $l$ corresponds roughly to the ambition of a system. Large values of $l$ mean that a system is prepared to modify document vectors by a large amount, and small values indicate a more conservative approach. When $l$ is extremely large, DT3 is equivalent to DT2, and when $l$ is zero, DT3 is equivalent to the control system.

All three strategies are computationally cheap, and could be used over large collections without a problem. [5] and [20] present several alternative strategies that we might have implemented, but we tested none of these for two main reasons. First, none of these alternative strategies has been shown to perform consistently better than DT1. Second, any properties of these strategies that have been identified in the past are unlikely to carry over to an experiment using a training set of several thousand queries. New strategies are needed to cope with large amounts of training data.

Each strategy can be applied before or after the normalisations required for the BD-ACI-BCA similarity measure (a linear transformation before normalisation is equivalent to a non-linear transformation of the normalised vectors). All of the results presented here will be for systems that perform Document Transformation first. Since the normalisation involves taking logarithms of term frequencies, these systems were less sensitive to small changes in the learning parameters than systems that normalised first.

### 5.1 Implementation

All of our systems were created by modifying the source code for version 11 of the SMART system. The SMART system is a public-domain implementation of the vector space model [6]. It is suitable for collections up to a few hundred megabytes in size, and has been designed for flexibility rather than efficiency. SMART performs stemming using Porter's algorithm, and allows stop words to be removed. All of our systems use both of these features.

## 6 Experimental Results

Figure 1(a) shows the performance of our three strategies as a function of the number of clickthroughs in the training set. $\alpha$ and $\beta$ were chosen to maximise the performance of DT1 and DT2 for a training set of size 2000.

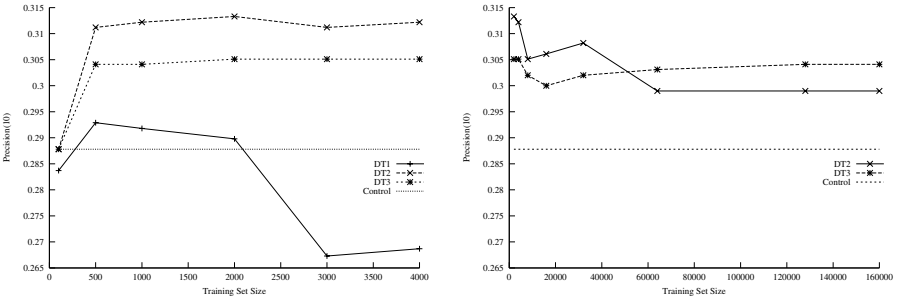


**Fig. 1.** Performance of the Document Transformation strategies when (a) training sets are small, and (b) for some large, artificial training sets. For DT1, $\alpha = 0.03$. For DT2, $\beta = 1.5$. For DT3, $\alpha = 0.03$, $\beta = 1.5$, and $l = 10$

DT1 has performed well over the small training sets used by previous studies, but suffers here from saturation of the document vectors. It is a little better than the control strategy for a training set of size 2000, which is not surprising since $\alpha$ was chosen to give the best possible performance in this situation. After this point, however, the performance of DT2 drops well below the control. Further evidence for saturation of the document vectors is provided by the first part of the DT1 curve. Even though $\alpha$ was chosen to maximise performance for a training set of size 2000, this value of $\alpha$ leads to superior performance for smaller training sets.

DT2 appears unaffected by document vector saturation, and achieves a stable improvement of around 8% over the control system.

When $l$ is large — 1000, say — DT3 reduces to DT2 for training sets of several thousand clickthroughs. In Figure 1(a), $l$ was set to 10 to limit the changes that could be made to the document vectors. DT3 is nearly as good as DT2 in this situation, showing that small changes to the document vectors are enough to account for most of the improvement achieved by DT2.

Even though a training set of size 4000 is larger than those considered by previous studies, it is still much smaller than the number of clickthroughs generated in a day by a popular search engine. To see how DT2 and DT3 would cope with large quantities of training data, we created some artificial training sets by concatenating copies of the 4000 genuine clickthroughs (the largest set contained 160,000 clickthroughs). The results of these experiments are presented in Figure 1(b).

Both DT2 and DT3 level out above the control system, but this time DT3 is the superior strategy. For systems learning from large numbers of clickthroughs, this result suggests that it is worth ensuring that document representations cannot be altered too drastically by the learning process.

Even though DT2 is out-performed by DT3, it does not appear to suffer from document vector saturation, probably because we are performing Docu-

ment Transformation before normalisation, and the normalisation involves logarithms of term frequencies. There should be some point after which DT2 will be susceptible to saturation, but a demonstration of this claim might require truly gigantic training sets.

## 7 Discussion

For large training sets, strategy DT3 achieved a stable improvement of around 6% over the control system. This improvement might have been even greater were it not for a serious problem with our test collection. Many of the pages mentioned most often in the referer log had been taken down for violating faculty guidelines about appropriate web use. Eight of the twenty most popular pages fall into this category, including the two most popular pages overall. Had they been part of the collection, these popular pages might have been expected to contribute most to the success of Document Transformation.

As a rule-of-thumb, [23] has suggested that a difference of more than 5% between IR systems is noticeable, and a difference of more than 10% is material. Our 6% improvement therefore suggests that Document Transformation is a viable strategy for improving the performance of web search engines.

### 7.1 Collaboration with a Commercial Search Engine

Further work in this area would profit from a large collection of web pages and a log recording interactions with that collection. The companies that own the major search engines are in the best position to meet this need. There is a good chance one of these companies would be prepared to collaborate in a large-scale study of Document Transformation: Lycos, Excite and AltaVista have all released query logs to selected researchers [2,13,21].

### 7.2 Better User Histories

There are many extensions of the approach presented here that would be worth investigating. Clickthroughs were the only user histories used for this study, but more complete histories would be valuable. For example, it would be useful to know how long a user spent reading a page, whether she saved it to disk or printed it, and whether she returned to the search engine to try a similar query. All of these factors would allow a more accurate judgement about whether her query was relevant to the page she found.

### 7.3 Dynamic Collections

We used a static collection for our experiments, but the web, of course, is dynamic. Working with a dynamic collection would require some adjustments to the strategies for Document Transformation investigated here. A clickthrough from five months ago, for example, is a less reliable guide to the contents of

a page than a clickthrough generated yesterday. A clickthrough's effect on a document representation should therefore fade with time.

Of the three strategies tested here, DT3 alone should adapt well to a dynamic collection. Once the 'learned part' of a document vector is full, later clickthroughs will reduce the impact of earlier clickthroughs. Other strategies, however, should also be tried.

If Document Transformation is successful, positive feedback loops will be created: the documents most likely to be returned will be those that have been looked at before. New documents may have little chance of breaking into one of these loops unless special measures are taken.

One way of dealing with this problem is to design a similarity measure that is sensitive to the age of a document. All other things being equal, newer documents should be regarded as more similar to a query than older documents.

### 7.4 Space Requirements

If Document Transformation is to be applied to large collections, the space occupied by the modified document vectors will need to be considered. To minimise storage requirements, it would be useful to set a limit on the number of terms that can be added to a vector (Strategy DT3 sets an upper bound on the length of the learned part, but the number of terms in a vector of fixed length can grow arbitrarily large as the weight of each becomes arbitrarily small). It would be worth investigating whether limiting the number of terms that can be added to a vector affects the success of Document Transformation.

## 8 Conclusion

Information retrieval systems may improve their performance by collecting and analysing user histories. Document Transformation is one instance of this approach based on the idea of moving a document vector towards a query known to be relevant to that document. Although Document Transformation was proposed many years ago, it has received little attention, probably because it was difficult to study in the pre-Internet era. This paper has shown how studies of Document Transformation can now easily be carried out using the logs kept by web search engines.

Our experiments have suggested that Document Transformation can improve retrieval performance over large collections of webpages. All of the user histories considered in this study were genuine: they were taken from a referer log kept by a web server at the University of Melbourne, and reflect the actions of actual web searchers rather than experimental stooges. To our knowledge, no other study of Document Transformation has attained this level of realism.

Previous strategies for Document Transformation have not been suitable for long-term use. They are susceptible to saturation of the document vectors: a process where the terms in the original document vector are gradually overpowered by terms that have later been associated with the document. This study has introduced a new strategy that overcomes this problem.

The ultimate test for Document Transformation will be whether it can improve the performance of one of the major search engines. All of the experiments described here could be repeated on a much larger scale using the logs collected by one of these search engines. Our results suggest that it would be worth collaborating with the developers of a commercial search engine on a large-scale study of Document Transformation.

## References


1. Vo Ngoc Anh and Alistair Moffat. Improved retrieval effectiveness through impact transformation. In *Proceedings of the Thirteenth Australasian Database Conference*, Melbourne, Australia, in press. 268
2. Doug Beeferman and Adam Berger. Agglomerative clustering of a search engine query log. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pages 407–416, Boston, 2000. ACM Press. 271
3. Richard K. Belew. Adaptive information retrieval: Using a connectionist representation to retrieve and learn about documents. In *Proceedings of the Twelfth International Conference on Research and Development in Information Retrieval*, pages 11–20, Cambridge, MA, 1989. ACM Press. 264
4. Justin Boyan, Dayne Freitag, and Thorsten Joachims. A machine learning architecture for optimizing web search engines. In *Proceedings of the AAAI Workshop on Internet-Based Information Systems*. 1996. 264
5. T. Brauen. Document vector modification. In Gerard Salton, editor, *The SMART Retrieval System : Experiments in Automatic Document Processing*, pages 456–484. Prentice Hall, NJ, 1971. 265, 266, 267, 268, 269
6. Chris Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Department of Computer Science, Cornell University, Ithaca, NY, 1985. 269
7. Chris Buckley and Ellen M. Voorhees. Evaluating evaluation measure stability. In *Proceedings of the Twenty Third Annual International Conference on Research and Development in Information Retrieval*, pages 33–40, Athens, Greece, 2000. ACM Press. 268
8. Hsinchun Chen. Machine learning for information retrieval: Neural networks, symbolic learning, and genetic algorithms. *Journal of the American Society of Information Science*, 46(3):194–216, 1995. 264
9. The Direct Hit popularity engine technology: A white paper, 1999. Available from `www.directhit.com/about/products/technology_whitepaper.html`. 263
10. S. Friedman, J. Maceyak, and S. Weiss. A relevance feedback system based on document transformations. In Gerard Salton, editor, *The SMART Retrieval System: Experiments in Automatic Document Processing*, pages 447–455. Prentice Hall, NJ, 1971. 265
11. Norbert Fuhr and Chris Buckley. A probabilistic learning approach for document indexing. *Information Systems*, 9(3):223–248, 1991. 264
12. M. Gordon. Probabilistic and genetic algorithms for document retrieval. *Communications of the ACM*, 31(10):1208–1218, 1988. 264
13. B. Jansen, A. Spink, J. Bateman, and T. Saracevic. Real life information retrieval: A study of user queries on the web. *SIGIR Forum*, 32(1):5–17, 1998. 271
14. K. L. Kwok. A neural network for probabilistic information retrieval. In *Proceedings of the Twelfth Annual International Conference on Research and Development in Information Retrieval*, pages 21–30, Cambridge, MA, 1989. 264

15. David D. Lewis. Learning in intelligent information retrieval. In Lawrence A. Birnbaum and Gregg C. Collins, editors, *Machine Learning: Proceedings of the Eighth International Workshop*, pages 235–239, Evanston, IL, 1991. Morgan Kaufmann. 264
16. M. Maron and J. Kuhns. On relevance, probabilistic indexing and information retrieval. *Journal of the Association for Computing Machinery*, 7(3):216–244, July 1960. 265
17. Benjamin Piwowarski. Learning in information retrieval: a probabilistic differential approach. In *Proceedings of the Twenty Second Annual Colloquium on Information Retrieval Research*, Cambridge, England, April 2000. 264
18. J. Rocchio, Jr. Relevance feedback in information retrieval. In Gerard Salton, editor, *The SMART Retrieval System : Experiments in Automatic Document Processing*, pages 313–323. Prentice Hall, 1971. 265
19. Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, Reading, MA, 1989. 264
20. J. Savoy and D. Vrajitoru. Evaluation of learning schemes used in information retrieval. Technical Report CR-I-95-02, Faculty of Sciences, University of Neuchâtel, 1996. 266, 267, 268, 269
21. Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. Analysis of a very large AltaVista query log. Technical Report 1998-014, Systems Research Center, Digital Equipment Corporation, Palo Alto, California, October 1998. 271
22. Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In H-P Frei, D. Harman, and P. Schäuble, editors, *Proceedings of the Nineteenth International Conference on Research and Development in Information Retrieval*, pages 21–29, New York, 1996. ACM Press. 265
23. Karen Sparck Jones. Automatic indexing. *Journal of Documentation*, 30:393–432, 1974. 271
24. Justin Zobel and Alistair Moffat. Exploring the similarity space. *SIGIR Forum*, 32(1):18–34, 1998. 264, 265