Vector Space Ranking: Can We Keep it Simple?

Vo Ngoc Anh *Alistair Moffat*

Department of Computer Science and Software Engineering
The University of Melbourne
Victoria 3010, Australia

{*vo,alistair*}*@cs.mu.oz.au*

Abstract: *The vector-space model is used widely for document retrieval, based upon the TF-IDF rule for calculating similarity scores between a set of documents and a query. One of the drawbacks of this approach is the need to select a specific formulation for the similarity computation. Here we present an initial attempt to simplify the heuristic, by hiding the various detailed calculations, and evaluating the term importance qualitatively rather than quantitatively. A new technique, called local reordering is introduced. Local reordering still relies on the vector-space model, as it employs a scalar vector product for calculating similarity scores. But there is no longer a requirement for precise values of the document or query vectors to be determined. Initial experiments on two data sets shows that it is highly competitive in terms of retrieval effectiveness. As a useful side effect, the method allows extremely fast query processing.*

Keywords Information retrieval, text indexing, vector-space ranking, similarity heuristic.

1 Introduction

Given a collection of documents $D = \{d\}$ and a query q, we wish to determine the documents in D that are most relevant to q, where "relevance" is judged by human assessors. Statistical methods for approximating human relevance use the TF-IDF rule for firstly calculating similarity scores between each document and the query and then choosing the documents with the highest similarity scores as the answers. The TF-IDF rule [Salton, 1989, Witten et al., 1999] asserts that, for each term t in q, the similarity score must vary as a positively correlated function of $f_{d,t}$, and as a negatively correlated function of f_t, where $f_{d,t}$ is the *within-document frequency* of t in d, and f_t is the *document frequency* of term t in the collection D.

Suppose that N and n are number of documents and number of distinct terms, respectively, in D. The vector space model deploys the TF-IDF rule in the following way. Conceptually, a n-dimensional vector space is constructed, with each dimension representing a term that appears in the collection. In the space, a document d is represented as

$$d = (w_{d,t_1}, w_{d,t_2}, \ldots, w_{d,t_n}),$$

and the query q as

$$q = (w_{q,t_1}, w_{q,t_2}, \ldots, w_{q,t_n}).$$

In this framework, the t_i are the distinct terms of the collection, and $w_{x,t}$ is the projection of document or query x in dimension t. That is, $w_{x,t}$ is the "importance" of t in x, and can be calculated by any formulation obeying the TF-IDF requirement. A similarity score $S(d,q)$ between d and q calculated by the cosine measure is of the form:

$$S(d,q) = \frac{\sum(w_{d,t} \cdot w_{q,t})}{\sqrt{\sum w_{d,t}^2} \cdot \sqrt{\sum w_{q,t}^2}}\,, \qquad (1)$$

where the three summations are over all n terms.

In mathematical terms, $S(d,q)$ represents the cosine of the angle between the vectors d and q. The greater the value of $S(d,q)$, the smaller the angle between the vectors d and q, and the more "similar" they can be claimed to be. Note, however, that the calculation $\sum(w_{d,t} \cdot w_{q,t})$ alone can be considered to meet the gross requirements of the TF-IDF rule, and the denominator in equation 1 represents a modification of the TF-IDF value that penalizes long documents or queries, and should be thought of as a component of the cosine rule, but not necessarily of the TF-IDF approach. Note also that $W_d = (\sum w_{d,t}^2)^{0.5}$ is usually referred to as the *document length*, and that the corresponding query length $(\sum w_{q,t}^2)^{0.5}$ is constant for a given query and can be ignored.

Zobel and Moffat [1998] explored a range of similarity score variants. They showed that the number of possible variants is large, and that none of them seems to be an absolute winner when retrieval effectiveness is taken as the criteria. They do, however, suggest the use of the mechanism denoted BD-ACI-BCA, which performed well in their experiments. In this formulation, $w_{d,t} = 1+\log_e f_{d,t}$, and $w_{q,t} = (\log_e(1+f^m/f_t))\cdot(1+\log f_{q,t})$, where f^m is the maximum value of f_t in the collection, and the document length is a normalized function of W_d [Singhal et al., 1996].

The work reported in this paper represents an attempt to escape the need to pick a particular formulation. We introduce a method called *local reordering*, which takes each document in the collection as an (almost) independent scope, and assesses the importance of each term appearing in it without reference to other documents.

Proceedings of the 7th Australasian Document Computing Symposium, Sydney, Australia, December 16, 2002.

Our presentation begins in Section 2 with a examination of the use of document lengths in the cosine measure. That discussion leads to our principal hypothesis: that document length is used as a quantitative surrogate for a more direct requirement, that of qualitatively estimating the importance of each term that appears in a document to the "meaning" of that document.

The idea is developed further in Section 3 where we report our initial attempts to avoid the detailed calculation involved in the cosine computation, and return to the simpler estimation implied by the TF-IDF rule. Section 4 then provides experiment results that show the simple approximation to be perfectly adequate in terms of retrieval effectiveness (hence the title of this paper). Implementation issues are discussed briefly in Section 5. Section 6 then finalizes our presentation with some conclusions and directions for future work.

A brief word on notation is necessary. In this work we concentrate on individual documents d rather than the whole collection D, and it is useful to slightly abuse some of the usual notation. In particular, a document d is supposed to have n_d distinct terms $t_1, t_2, \ldots, t_{n_d}$, and $T_d = (t_1, t_2, \ldots, t_{n_d})$ is referred to as the *term list* of d. In any formulation of $S_{d,q}$, the value derived solely from a certain term t and the document d is called *retrieval contribution*, or *contribution* of term t in d, and is denoted by $\omega_{d,t}$. Thus, in equation 1 we have $\omega_{d,t} = w_{d,t}/W_d$. We also refer to $\Omega_d = (\omega_{d,t_1}, \omega_{d,t_2}, \ldots, \omega_{d,t_{n_d}})$ as the *retrieval contribution list* for d.

2 Document length: Keeping it simple

In this section we examine the possibility of avoiding the document length factor in the cosine similarity computation. The key idea is to define term contribution locally within each document, instead of globally in whole document collection.

Consider equation 1. The document length division is intended to penalize long documents, but it is a rather blunt instrument, and also creates a bias in favor of short documents. That phenomenon was noticed by a range of investigators, and there have been several attempts to ameliorate it. Most recently, Singhal et al. [1996] and Chowdhury et al. [2002] normalized the value of document lengths and were able to significantly improve retrieval effectiveness. Anh and Moffat [2002] obtained further improvement by normalizing the global set of term contributions.

It is possible that further tweaking with the document length might lead to continued incremental gains in retrieval effectiveness. But it is also attractive to think about completely removing it as a factor. In their study Zobel and Moffat [1998] explored mechanisms using a unit document length, but without obtaining competitive effectiveness.

Here we propose a partial removal. Instead of normalizing each term weight by the document length, the term contribution list Ω_d of each document d is re-evaluated in the context of that document so that occurrences of a term may well be treated differently in short documents compared to long ones. For any d, the re-evaluation is done locally in d, without consulting any term contribution values in any other document. This re-evaluation is thus characterized as *local*. By way of contrast, the impact transformation scheme given by Anh and Moffat [2002] is also a re-evaluation, but global.

Now consider Ω_d again. The largest value of Ω_{d_1} in a document d_1 might still be small compared to the largest value of Ω_{d_2} in document d_2. The intention of the local re-evaluation process is to adjust the values in Ω_d so that, over the set of documents D, the largest and smallest values of Ω_d for each document are roughly the same. As in our previous work [Anh et al., 2001, Anh and Moffat, 2002], the contributions are also quantized, so that each *surrogate weight* is used as the retrieval contribution $\omega_{d,t}$ for a whole set of terms. In essence, in each document d the largest component in Ω_d is mapped to a pre-determined integer k, and each other $\omega_{d,t}$ value is represented by an integral surrogate weight in the range 1 to k. Suitable values for k are discussed shortly.

Once it is accepted that it is the relative ordering of $\omega_{d,t}$ values that is important, rather than their actual numeric values, there is no need to persist with a formulation for document length. Instead, document-term weights are calculated as $\omega_{d,t} = w_{d,t}$, and mapped onto the set of surrogates. Requiring that each document d use the full range of k surrogate values guarantees a kind of length-based normalization process, but one in which there is no cross-talk between documents.

Four mapping methods are considered here.

- *By-Value*: The first method is based on the value of the mapped elements. Initial experiments showed that the distribution of $w_{d,t}$ for a single document d is similar to that of the impacts over the whole collection (see [Anh et al., 2001]). That is, for a certain d, the number of high values is much smaller than the number of low values. Following the lead of [Anh and Moffat, 2002], we map the value w to

$$\left\lfloor k \cdot \frac{\log w - \log U_d}{\log U_d - \log L_d + \epsilon} \right\rfloor + 1 \,, \qquad (2)$$

  where U_d and L_d are the maximum and minimum, respectively, of the values $w_{d,t}$ in document d. Note that the use of ϵ ensures that none of the quantized values is higher than k. Note also that the mapping is local, and ensures that surrogate weights of both 1 and k are generated on non-trivial documents.

The other three mappings are based on a strategy called *By-Rank*, in which the fraction of the elements in Ω_d that share each surrogate weight is determined in advance. It seems clear that surrogate weight k should not be used with a greater frequency than surrogate weight 1. Taking x_i to be the number of elements assigned a surrogate weight of i (where i is between 1 and k inclusive), three variants have been considered:

- *By-Rank, Geometric*: The number of elements in a document corresponding to each surrogate weight, in decreasing order of the weights, forms a geometric subsequence. That is, x_i= $x_{i+1} \cdot B$, with B determined in similar way given in equation 2 with L_d and U_d replaced by 1 and n_d respectively.

- *By-Rank, Arithmetic*: The number of elements in a document corresponding to each surrogate weight, in decreasing order of weights, forms a arithmetic subsequence. In particular, x_i= $x_{i+1} + B$, where $B = (n-1)/(k \cdot (k-1))$.

- *By-Rank, Uniform*: Within a document each surrogate weight corresponds to approximately the same number of elements.

For example, when *By-Rank* is combined with the *Uniform* method, approximately n_d/k of the n_d distinct terms in document d are assigned to each of the surrogate ranks from 1 to k.

3 TF-IDF: Keeping it simple

The approach described in the previous section eliminates one of the three parameters in the cosine measure, by removing the need to choose a document length normalization regime. Nevertheless, there are still two factors remaining. One way of trying to establish these would be to repeat a suite of experiments of the kind performed by Zobel and Moffat [1998].

Instead, we prefer to again just make a simplifying assumption, and revert to the underlying basis espoused in the TF-IDF rule.

In its original form, TF-IDF is not a precise definition – it is a philosophy. It is our human desire for precision in computation that has led to overly precise formulations, with their various tuning factors. So, in keeping with the spirit of the "rank is more important than value" claim of the previous section, this section explores simple rules that order the term contributions in way that is consistent with the TF-IDF philosophy. Ideally, we would like to create a sorted list of terms corresponding to each document without doing any detailed computation. The surrogate weights of the n_d terms in document d are then computed for use in the query processing regime, without any further recourse to collection or document statistics.

Again, we focus on one document at a time, and divide the process into two phases. The first *sorting* phase orders the term list T_d of document d in decreasing order of term contribution. In the second *mapping* phase, each value in the term list T_d is converted to an integer in the range 1 to k. As the end of the process, these integer values serve as term contributions, and form the list Ω_d.

There are many ways of ordering the term contributions in the sorting phase. For example, any cosine formulation could be used to calculate a numeric sort key. But our overriding desire in this work is to "keep it simple", and we eschew formula-driven orderings in favor of more primitive methods. In particular, the following variants are explored in the experiments described in Section 4:

- (*IDF*, *TF*): The term list is sorted in increasing order of f_t, with ties on f_t broken by using decreasing order on $f_{d,t}$ as a secondary key. In this arrangement, the *IDF* component is presumed to dominate the *TF* component; the nomenclature reflects the lexicographic sort ordering.

- (*TF*, *IDF*): The *IDF* component is a statistic of the collection, rather than of a particular document. The (*IDF*, *TF*) method, in that sense, does not reflect our "keep it simple" theme. From the point of view of a single document, the *TF* component should dominate. In this arrangement, the term list is sorted in decreasing order of $f_{d,t}$, with ties broken by using increasing order on f_t as a secondary key.

- (*TF*, *IDF*, *stopped*): One possible drawback of the (*TF*, *IDF*) proposal is that it emphasizes common words in English that are devoid of meaning, such as "the", "and", and "at". These words tend to have high $f_{d,t}$ values, and get mapped by the (*TF*, *IDF*) ordering onto the highest surrogate weights. To correct for this anomaly, the (*TF*, *IDF*, *stopped*) mechanism resets the $f_{d,t}$ value of a set of *stop words* to 1, to guarantee their positions at the tail of each sorted list. There is, however, a consequent issue as to how to decide whether a word should be stopped [Fox, 1992]. To keep it simple, here we classify words appearing in 20% or more of the documents (that is, $f_t \geq N/5$) as being stop words.

The number of different f_t values is high in any non-trivial document collection, so in any of these three sorting rules there is only a small chance for two different terms in the same document have the same sort key value.

A fourth method is also included in the experiments described shortly:

- (*TF* × *IDF*): The value $w_{d,t}$ calculated by the BD-ACI-BCA similarity computation is used as the sort key, with the term list sorted into decreasing order.

Once the term list has been sorted, it is partitioned using one of the four mechanisms described in Section 2, except that the *By-Value* partitioning process can only be coupled with the (*TF* × *IDF*) ordering. The resulting surrogate weights are used as the basis of a similarity computation.

4 Experiments

The aim of the first experiments is to ensure that the reported approaches result in acceptable retrieval effectiveness, and to choose appropriate parameters or methods. The dataset *WSJ2* – a subset of *Disk2* of the *TREC* corpus [Harman, 1995], is employed here for that purpose. The collection *WSJ2* is a homogeneous text collection, containing the text of the *Wall Street Journal* for the period 1990–1992. The collection has around $75{,}000$ documents totaling approximately 240 MB. To measure effectiveness, 150 queries were formed by taking the "title" fields of *TREC* topics 051–200.

To control the complexity of the experimental process, a set of initial experiments was undertaken with the *By-Rank* process to determine the best of the three alternative partitioning regimes. As discussed in Section 2, the three mechanisms are *Geometric*, *Arithmetic*, and *Uniform*. They are similar in many ways to the methods discussed by Anh and Moffat [2002], and the preliminary experiments led to the expected results – *Geometric* outperforms *Arithmetic*, and *Arithmetic* outperforms the *Uniform* method. Those results will appear in a more detailed version of this paper. In the remainder of this presentation, whenever a mapping is used to convert a sorted list of objects into a set of integers, the *Geometric* method is employed.

A second set of experiments was then undertaken using the various proposed methods to sort the term list of a document. The BD-ACI-BCA cosine mechanism, which includes document-length pivoting, was used as a baseline in these experiments, and is denoted in the results by "cosine, baseline". The same similarity formulation, but without document-length normalization, and thus without pivoting, was used as the basis of the $w_{d,t}$ values required by the *By-Rank* and *By-Value* surrogate assignment mechanisms. In the results in Figure 1 these two are denoted as "(*TF* × *IDF*), *By-Rank*" and "(*TF* × *IDF*), *By-Value*" respectively.

Note that the decision to continue only with the *Geometric* partitioning means that the value of k (the number of distinct surrogate weights) is the only tunable parameter that remains; and that the retrieval achieved by the "baseline" mechanism is independent of k.

Figure 1 plots retrieval effectiveness as a function of k for the five methods, and compares them to the BD-ACI-BCA baseline. Three different metrics for assessing retrieval effectiveness are shown. The most reliable metric is average precision [Buckley and Voorhees, 2000]. However, precision at 10 documents retrieved and reciprocal rank are also included, as they are important for many people, especially in situations in which number of identified-as-relevant documents is unknown – when web searching, for example.

To rank the documents with respect to a query, the same process was applied to each query as was applied to each document – namely, the terms in the query were ordered using the same rule as the terms in the documents; and then the ordered terms were mapped to surrogate weight values using the same mapping as was used for the documents. The inner-product of the two resulting integer vectors was then calculated, and used as a similarity score without any further adjustment.

There are a number of interesting points that can be drawn from the results.

First, except for the (*IDF*, *TF*) term ordering regime, all of the methods using surrogate weights are capable of giving excellent retrieval performance and of outperforming the BD-ACI-BCA baseline by a compelling margin. Indeed, the good methods have surprisingly similar effectiveness curves when plotted as a function of k, and all are relatively stable when k is greater than about 6. Of the four good methods, (*TF*, *IDF*, *stopped*) has a slight edge.

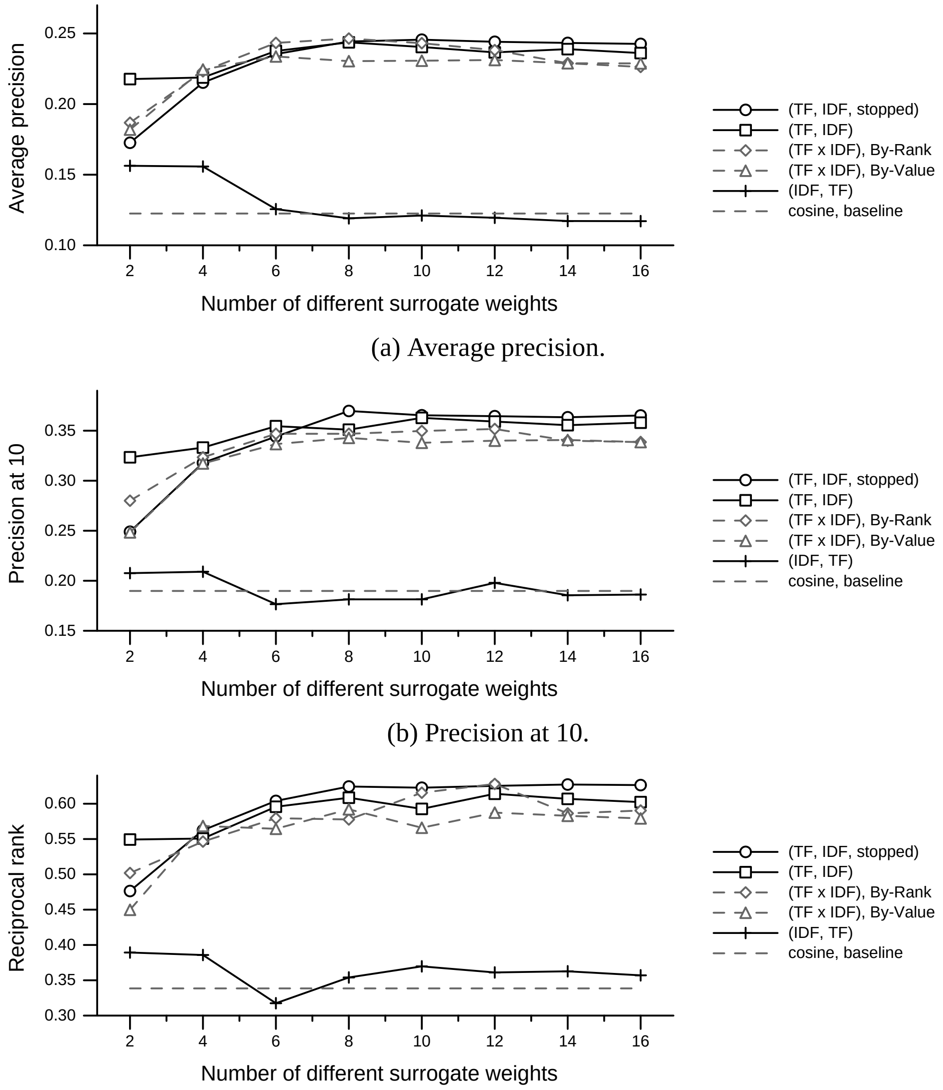
Second, it should be noted that even with $k = 2$ the new methods give better effectiveness than the baseline. This outcome was completely unexpected, and means that just one bit can be used to code term importance, instead of the several bits required in conventional cosine implementations to store the within-document frequency value associated with each document pointer [Witten et al., 1999]. The two different surrogate weights in the binary case can be interpreted as "high relevance" and "some relevance" of the term to the document.

Third, (*IDF*, *TF*) performs poorly in all situations. This outcome is also somewhat surprising, since at face value it means that the *IDF* factor performs only a minor role in determining the contribution made by a term. One possible reason for the poor performance is that numerically the *IDF* takes on many more distinct values than does the *TF*, and so there is less opportunity for *TF* to be used in a tie-breaking role. On the other hand, if the primary sort key is given by *TF*, then the *IDF* is likely to be used relatively often to break ties. That is, in (*TF*, *IDF*) both of *TF* and *IDF* are likely to influence the ordering more often than they do in (*IDF*, *TF*). If this is the case, then there may be benefit to be gained by quantizing either or both of the *TF* and *IDF* components before applying the ordering step.

An issue that may be puzzling the reader is that in the (*TF*, *IDF*) regime, almost certainly the highest surrogate weights are assigned to the most common words; yet this does not appear to impact retrieval effectiveness. The key to understanding this paradox is to note that if a word such as "the" is assigned the same surrogate weight of k in every document in the collection, then appearance of "the" in a query (or non-appearance) has no net effect on the eventual ranking, since the similarity score for every document is adjusted in the same way. That is, the sheer ubiquity of common words ensures that they have only a small impact upon the document ranking.

The third set of experiments takes the best method – ordering mechanism (*TF*, *IDF*, *stopped*), with *Geometric* partitioning, and $k = 10$ – and compares it to other retrieval heuristics on a different document collection, against which a wide range of retrieval mechanisms have been applied.

For this phase of the experimentation, collection *wt10g* is employed, as was used in *TREC-9*. The collection contains around 1.6 millions documents with a total size of about 10 GB. The documents were



(a) Average precision.

(b) Precision at 10.

(c) Reciprocal rank.

Figure 1: Retrieval effectiveness as a function of the number of distinct surrogate weights used, using three different effectiveness metrics, and dataset *WSJ2*. The queries are taken from the title field of *TREC* topics 051–200, and effectiveness values are averaged over the set of queries.

crawled from the Internet and can be considered as heterogeneous. The collection is accompanied by 50 ad-hoc queries (topics 451–500). The title fields of these topics are taken as queries. Hawking [2001] and Soboroff [2002] give information about the dataset *wt10g*.

The results of the third experiment are summarized in Figure 2. The effectiveness of the new mechanism is compared with the baseline implementation of BD-ACI-BCA, and with the performance reported by Anh and Moffat [2002] for their impact-transformation technique. The new method also performs well on this data set. Note that with the same data set, impact transformation was shown to perform well compared with other methods participating in *TREC-9* [Anh and Moffat, 2002], including systems employing additional heuristics such as relevance feedback, query expansion, and phrase indexing. (However we also note that it is inappropriate to include *TREC-9* results in Figure 2, as we manually edited the queries to correct spelling mistakes, whereas the *TREC* systems were permitted to only employ automatic correction techniques.)

5 Implementation and efficiency

Local reordering is straightforward to implement, and represents a relatively minor variation from the impact sorted indexes we have described previously [Anh et al., 2001, Anh and Moffat, 2002]. Each index list contains *k* blocks of sorted document numbers, and can be represented using the standard integer coding techniques [Witten et al., 1999]. Query processing is performed using a

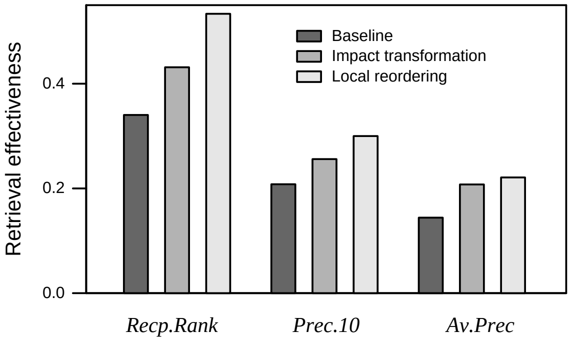


Figure 2: Relative comparison of effectiveness performance of three approaches: the standard BD-ACI-BCA cosine measure (as a baseline), the impact transformation mechanism described by Anh and Moffat [2002], and the local reordering technique described in this paper, for the collection *wt10g*. The three metrics shown are average values calculated over 50 queries taken from *TREC* topics 451–500, with spelling mistakes manually corrected. For the local reordering approach, the number of different surrogate weights is $k = 10$, the sorting method is (*TF*, *IDF*, *stopped*), and the *By-Rank* partitioning method is *Geometric*.

look-up table as a priority queue, and because all of the value manipulated are small integers, is fast.

To date we have not explored any pruning heuristics, and the results presented in Section 4 are for full evaluation. An extended version of this paper will investigate pruning operations, and quantify the tradeoff they offer between execution time and retrieval effectiveness.

6 Conclusion

Local reordering has three distinct benefits. First, it is largely free of the tunable parameters and "knobs" that plague other similarity heuristics. Our intention throughout was to "keep it simple", and we have succeeded in that aim.

Second, it achieves excellent retrieval effectiveness. In the experiments conducted to date, the local reordering matches the best systems that contributed to the *TREC-9* experiments in 2001. Experiments to validate this claim in other test environments are currently being planned.

Third, it allows an extremely efficient implementation. All query-time operations are on small integers, and query processing is rapid. There is little or no overhead cost in terms of index space.

As well as further experiments with the current implementation, we plan to incorporate a number of other "simple" extensions, including the appropriate indexing of anchor text and any meta-data, and perhaps the use of a simple rules based upon term location and context to adjust the surrogate weights. With these changes we hope to create a simple, efficient, and effective, web searching tool.

Acknowledgement This work was supported by the Victorian Partnership for Advanced Computing and the Australian Research Council.

References

V. N. Anh, O. de Kretser, and A. Moffat. Vector-space ranking with effective early termination. In W. B. Croft, D. J. Harper, D. H. Kraft, and J. Zobel, editors, *Proc. 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 35–42, New Orleans, LA, September 2001. ACM Press, New York.

V. N. Anh and A. Moffat. Impact transformation: Effective and efficient web retrieval. In M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, and K. Järvelin, editors, *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 3–10, Tampere, Finland, August 2002. ACM Press, New York.

C. Buckley and E. M. Voorhees. Evaluating evaluation measure stability. In N. J. Belkin, P. Ingwersen, and M.-K. Leong, editors, *Proc. 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 33–40, Athens, Greece, September 2000. ACM Press, New York.

A. Chowdhury, M. C. McCabe, D. Grossman, and O. Frieder. Document normalization revisited. In M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, and K. Järvelin, editors, *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 381–382, Tampere, Finland, August 2002. ACM Press, New York.

C. Fox. Lexical analysis and stoplists. In W.B. Frakes and R. Baeza-Yates, editors, *Information Retrieval: Data Structures and Algorithms*, chapter 7, pages 102–130. Prentice-Hall, Englewood Cliffs, New Jersey, 1992.

D. K. Harman. Overview of the second text retrieval conference (TREC-2). *Information Processing & Management*, 31(3):271–289, May 1995.

D. Hawking. Overview of the TREC-9 Web Track. In E. M. Voorhees and D. K. Harman, editors, *The Ninth Text REtrieval Conference (TREC-9)*, pages 87–102, Gaithersburg, MD, November 2001. NIST Special Publication 500-249.

G. Salton. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Addison-Wesley, Reading, Massachusetts, 1989.

A. Singhal, G. Salton, M. Mitra, and C. Buckley. Document length normalization. *Information Processing & Management*, 32(5):619–633, 1996.

I. Soboroff. Does WT10g look like the web? In M. Beaulieu, R. Baeza-Yates, S. H. Myaeng, and K. Järvelin, editors, *Proc. 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 423–424, Tampere, Finland, August 2002. ACM Press, New York.

I. H. Witten, A. Moffat, and T. C. Bell. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann, San Francisco, second edition, 1999.

J. Zobel and A. Moffat. Exploring the similarity space. *ACM SIGIR Forum*, 32(1):18–34, Spring 1998.