On the Effectiveness of Relevance Profiling

David J Harper
Smart Web Technologies Centre
The Robert Gordon University
Aberdeen AB25 1HG UK
d.harper@rgu.ac.uk

David Lee
Smart Web Technologies Centre
The Robert Gordon University
Aberdeen AB25 1HG UK
david.lee@smartweb.rgu.ac.uk

Abstract

Relevance profiling is a general process for within-document retrieval. Given a query, a profile of retrieval status values is computed by sliding a fixed sized window across a document. In this paper, we report a series of bench experiments on relevance profiling, using an existing electronic book, and its associated book index. The book index is the source of queries and relevance judgements for the experiments. Three weighting functions based on a language modelling approach are investigated, and we demonstrate that the well-known query generation model outperforms one based on the Kullback-Leibler divergence, and one based on simple term frequency. The relevance profiling process proved highly effective in retrieving relevant pages within the electronic book, and exhibits stable performance over a range of sliding window sizes. The experimental study provides evidence for the effectiveness of relevance profiling for within-document retrieval, with the caveat that the experiment was conducted with a particular electronic book.

Keywords

relevance profiling; within-document retrieval; language modelling; information retrieval experimentation.

1 Background

Increasingly, long electronic documents are being published and delivered using web and other technologies, and users need to find information within these long documents. Various approaches have been proposed for within-document retrieval, including passage retrieval [1], and user interfaces supporting content-based browsing of documents [2]. We have developed a tool, ProfileSkim, for within-document retrieval based on the concept of relevance profiling [3], and we reported on a comprehensive user-centred evaluation of this tool [4] [5]. ProfileSkim integrates passage retrieval and content-based document browsing. The key concept underpinning the tool is relevance profiling, in which a profile of retrieval status values is computed across a document in response to a query. Within the user interface, an interactive bar graph provides an overview of this profile, and through interaction with the graph the user can select and browse in situ potentially relevant passages within the document. The reader is referred to [3] for a description of, and detailed rationale for, relevance profiling and the ProfileSkim user interface.

In the user evaluation study [4] [5], we compared the performance of ProfileSkim against a tool based on the sequential "Find" command delivered with most browsers and word processing packages. We concluded that:

- Relevance profiling**,** as implemented and presented by the ProfileSkim, was effective in enabling users to identify relevant passages of a document;
- Using ProfileSkim, users were able to select and browse relevant passages more efficiently, because only the best matching passages needed to be explored; and
- Users found the tool satisfying to use for within-document retrieval, because of the overview provided by the relevance profile.

That study was based on a simulated work task, namely recreating part of the (back of book) index for an electronic book. On the basis of the results, we tentatively concluded that relevance profiling in combination with a set of pre-defined index entries, might prove to be an effective replacement for the typical intellectually derived book index.

In this paper, we wish to investigate the underlying relevance profiling mechanism, through controlled bench experiments. Specifically, we want to investigate different forms of the weighting function used for relevance profiling, and different settings for parameters of the process. Additionally, we wanted to investigate whether relevance profiling could provide an alternative to the intellectually generated book index. Consequently, we based our experiments on an electronic book, with an existing book index, that formed the basis of the experimental corpus.

Proceedings of the 9th Australasian Document Computing Symposium, Melbourne, Australia, December 13, 2004.

2 Relevance Profiling and Weighting Functions

Relevance profiling is a process whereby a profile of retrieval status values is computed across a document in response to a query. This profile is presented to the user in the form of a bar graph, and by interacting with this bar graph the user can identify, and navigate to, relevant sections of a document. In this paper each section, and hence bar in the graph, corresponds to a page of the document (electronic book).

The process is sketched in on object-oriented pseudo-code in Figure 1. Effectively, a retrieval status value (RSV) is computed for each word position in the document. The relevance profiling procedure operates by sliding a window of fixed size across the document (lines 7-18, Figure 1), and at each word position a window weighting function is applied (line 12). The weighting function is a function of the set of words in the document, query and the window, starting at the given position and of the given size. The window weights are aggregated for each page of the document, and the maximum weight achieved by a window starting in the page is returned (lines 13-16). Note, that a filter is applied so that only pages containing at least one term from the query are retrieved, i.e. assigned a score (line 10).

```
 1 process relevanceProfiling
          ( Document, Query, weightingFunction,
          WindowSize, scoresArray )
 2 begin
 3   for Page = 1 to Document.countPages()
 4   begin
 5     scoresArray[ Page ]= minimumWindowScore
 6   end
 7   for windowPosition = 1 to Document.length()
 8     begin
 9       PageInDoc= Document.pageOf( windowPosition )
10       if (Document.containsTermOf (Query, PageInDoc)
11         begin
12           windowWeight = weightingFunction
                   ( Document, Query, windowPosition,
                   WindowSize )
13           if (windowWeight > scoresArray [ PageInDoc ])
14           begin
15             scoresArray [ PageInDoc ]= windowWeight
16           end
17       end
18   end
19 end
```

Figure 1: Pseudo-code describing the relevance profiling process

The main purpose of the experimental study is to investigate the behaviour of the relevance profiling process, and specifically the effect on retrieval effectiveness of the choice of window size and weighting function. Before describing the experimental study, we introduce the various weighting functions that are investigated.

The weighting functions are based on a language modelling approach in which we model a document, as a probability distribution over the terms of the vocabulary, and similarly for each (sliding) window. We define the document model to be the probability distribution *P(t|D)*, where for every term *t* in the vocabulary (in our case, of the document), the model gives the probability that we would observe term *t* if we randomly sampled from the document *D*. Similarly, we define the window model, *P(t|W)* for window *W*.

We will define three weighting functions (line 12, Figure 1), which we refer to as the query generation model (GEN), the Kullback-Leibler model (KL), and a simple term frequency model (FREQ).

2.1 Query Generation Model (GEN)

We adapt the language modelling approach proposed for document retrieval in [6] [7] for relevance profiling. Language modelling is used to construct a statistical model for a text window, and based on this model; we compute the window RSV as the probability of generating a query (denoted *Q*). We model the distribution of terms (actually stemmed words) over a text window, as a mixture of the text window and document term distributions as follows:

$$\mathrm{P}(Q \mid W) = \prod_{t_i \in Q} p_{mix}(t_i \mid W) \qquad \textbf{(1)}$$

where: $p_{mix}(t_i \mid W) = \lambda * p(t_i \mid W) + (1-\lambda) * p(t_i \mid D)$

The estimates are smoothed by the document word statistics using the mixing parameter, λ [a]. The individual word probabilities are estimated in the obvious way using maximum likelihood estimators:

$$p(t_i \mid W) = n_{iW} / n_W \quad p(t_i \mid D) = n_{iD} / n_D \qquad \textbf{(2)}$$

where n_{iW} (n_{iD}), and n_W (n_D), are the number of term occurrences of term *i* in the window (document), and total term occurrences in the window (document) respectively.

Equivalently, we can rank windows by taking the log of both sides of (1), yielding:

$$\log \mathrm{P}(Q \mid W) = \sum_{t_i \in Q} \log p_{mix}(t_i \mid win) \qquad \textbf{(3)}$$

[a] We have investigated the best value for the mixing parameter empirically, and similar results were obtained in the range 0.8 through 0.999. In the experiments reported, we use 0.8.

2.2 Kullback-Leibler Model (KL)

The Kullback-Leibler divergence has been used extensively in applications of language modelling in information retrieval [8]. In this approach, we compute the Kullback-Leibler divergence between the window model and the document model, and we restrict the computation to the query terms, as follows:

$$KL(W \parallel D) = \sum_{t_i \in Q} p(t_i \mid W) \log\left(p(t_i \mid W) \big/ p(t_i \mid D)\right) \quad \textbf{(4)}$$

We use the strength of the divergence as a measure of window relevance with respect to a query. Intuitively, the larger the divergence of the window model from the (common) document model, the more likely the window is relevant to the query. Unlike the GEN model, the KL term weights have both a representation component (outside the *log*), and a discriminative component (the *log* term).

We are unable to use maximum likelihood estimates for the component probabilities, as we must avoid estimating zero for $p(t_i|W)$. We use the estimation rule attributed to Jefferys [9, p. 127] to estimate the parameters as follows:

$$p(t_i \mid W) = \left(n_{iW} + 0.5\right) \big/ \left(n_W + 1.0\right)$$
$$p(t_i \mid D) = \left(n_{iD} + 0.5\right) \big/ \left(n_D + 1.0\right) \quad \textbf{(5)}$$

2.3 Term Frequency Weighting Model (FREQ)

In this approach, we simply weight the window by summing the total occurrences of all query terms within the window. This approach can be given a language modelling interpretation, which will be useful in our later analysis. Using the query generation approach, we compute the window RSV as the probability of generating **any term** of the query as follows:

$$\mathrm{P}(any\ Q\ term \mid W) = \sum_{t_i \in Q} p(t_i \mid W) \quad \textbf{(6)}$$

This is equivalent to summing all the term occurrences in a window over the query terms.

3 Experimental Study

3.1 Research Questions

In this study we investigate three main research questions, namely:

- Is there an optimal window size for the sliding window, and is this optimal size dependent on the particular weighting function used?
- What is the relative effectiveness of the three weighting functions we have described?
- Could relevance profiling be used as an aid in generating a book index, through intellectual effort by a human, or indeed as a complete replacement for the book index?

In respect of window size, we conjecture that smaller window sizes will tend to be precision-enhancing as they are less likely to discover unrelated term occurrences, and that large window sizes will tend to be recall enhancing. We investigate a range of window sizes from a few sentences in length (50 words) up to 2-3 paragraphs or half a page (250 words).

For the weighting functions, we conjecture that the simple term frequency weighting (FREQ) will perform worst, given that it exploits less of the available information than the other functions. Both the query generation approach and the Kullback-Leibler approach have proved effective in conventional document retrieval. However, in this application, we believe that comparatively small size of sample text (namely the samples of window text) is likely to be a major limiting factor, and specifically in estimating probabilities. The simpler query generation model may prove more effective due to the smoothing provided by the mixture approach.

3.2 Corpus

We wish to investigate relevance profiling for within-document retrieval, and especially for long documents. And, in order to measure relative effectiveness, we need to establish a so-called "ground truth" for the experiments. We used van Rijsbergen's classic textbook, "Information Retrieval" [9], which is arguably long (191 pages, 60000 words). It is available in an electronic format, and it possesses a comprehensive book index. We had used this corpus successfully in previous end-user experiments [4] [5]. Here, we use the corpus in a similar way to a conventional test collection: the pages are the units of retrieval, the queries are provided by the book index entries, and the relevance assessments are the relevant pages associated with each index entry. We note that the book index was created by the book's author, and it is possible that the indexing is not exhaustive. Consequently, the absolute performance of the techniques we investigate is likely to be higher than we report. However, the main purpose of the experiments is to compare techniques, and in this respect all are evaluated against the same "ground truth".

Table 1 summarises the main characteristic of the corpus.

In the main, we conducted the experiments using the set of multi-term queries as all the weighting func-

tions are monotonic with respect to single term queries. We note that a high proportion of the multi-word queries are in fact phrasal queries rather than sets of words.

Number of pages:	191		
Number of words:	60035		
Index Entry Type	**No. of Entries[b] (queries)**	**Total relevant pages**	**Average #rels/entry**
single word	46	144	3.1
multi-word	232	766	3.3

Table 1: Details of corpus: "Information Retrieval" textbook by van Rijsbergen [9]

3.3 Experiment Design

The experiments were conducted as follows. Each query is used to generate a relevance profile for the document. Then, we simply rank the pages according to the retrieval status value, and evaluate the ranking based on the relevance judgements provided by the book index. In fact, this is not an ideal way of evaluating relevance profiling, which is designed to identify relevant groups (or clumps) of pages, and not simply individual pages. It would have been better to assess the effectiveness in identifying the same groups of pages identified in the book index. Given the availability of measures and tools for evaluating ranked output, we opted to do page ranking.

3.4 Measures

We used a range of single-valued measures averaged over the set of queries to assess effectiveness, namely: non-interpolated precision, R-precision (R-prec.), and three variations of the F-measure. R-precision is the precision achieved at rank R, where R is the number of relevant documents for a given query. This measure combines elements of both precision, proportion of retrieved R documents that are relevant, and recall, proportion of R relevant documents retrieved. The F-measure is simply the complement to the E-measure [9], and is given by:

$$F = R\,P/\left(\alpha R + (1-\alpha)P\right) \qquad \textbf{(7)}$$

where R[c] is recall, P is precision, and α enables one to bias the measure in favour of precision (α -> 1) or recall (α ->0). We use three variants with α set to 0.8 (precision-oriented), 0.5 (balanced harmonic mean of R and P), and 0.2 (recall-oriented).

To compute F-values for each query, we compute precision and recall at each different score in the ranking (note, not at each rank position), and then compute the optimal values F-values for the different settings.

Statistical significance testing was performed using the sign test, and p-values less than 0.05 were considered significant.

4 Experiments

We investigate each research question in turn, and present and discuss the results. Given the possible interaction between window size and weighting function, we decided to simply explore the window size initially with the query generation model (GEN), and we later confirmed the validity of the findings for the other two weighting functions.

4.1 Window Size Experiments

In the first set of experiments, we investigated effect of the size of the sliding window on the performance achievable with the query generation model (GEN). The size of the window was varied from 50 words through 200 words in steps of 25, and for 250 words. The experiment was run with two sets of queries, those for which the queries contained more than one word (MULTI), and those comprising a single word (SINGLE). In the context of document skimming, we think it likely that multi-word queries will be more usual that single word queries. However, we were also interested in the performance achievable with single word queries, as they form a significant proportion of entries in our book index. If ProfileSkim were used as a replacement for the book index then it would have to provide good performance for this type of query.

We present the window size results in Tables 2 and 3, where the **bold figures** are the best values achieved for each measure, and the *italicised figures* are values close to the best.

The results for the multi-term queries (Table 2) show that for F-measures, the results are broadly comparable over a wide range of window sizes from 50 through 250. Interestingly, the recall-oriented F (α=0.2) results are better than the other F results, indicating that relevance profiling may be a recall-oriented device. For the (averaged) non-interpolated precision and for R-Precision, the results are better for the smaller window sizes (50, 75), and significantly so compared with the results for window sizes 100 and up.

[b] There were a number of entries that we not used in the experiments, namely the 'see also' entries.

[c] This is a different use of 'R' and we trust the use is clear from the context.

Window size	**Prec.**	**R-prec.**	**F (α=0.8)**	**F (α=0.5)**	**F (α=0.2)**
50	*0.656*	0.559	0.699	0.695	*0.755*
75	**0.662**	**0.579**	*0.704*	*0.702*	**0.757**
100	0.648	0.541	*0.706*	*0.699*	*0.754*
125	0.650	0.551	*0.705*	*0.700*	**0.757**
150	0.645	0.541	**0.709**	**0.703**	**0.757**
175	0.647	0.546	*0.705*	*0.699*	**0.757**
200	0.643	0.539	0.701	0.695	0.752
250	0.628	0.514	0.691	0.686	0.749

Table 2: Averaged effectiveness of query generation weighting (GEN) for 232 multi-term queries for different window sizes

The performance for the single term queries (Table 3) is reduced by 5-10% compared with the multi-term queries, except for the R-Precision measure. Interestingly, for all measures, the maximum effectiveness is achieved at a window size of 200.

Window size	**Prec.**	**R-prec.**	**F (α=0.8)**	**F (α=0.5)**	F **(α=0.2)**
50	0.602	0.559	0.604	0.604	0.650
75	*0.618*	0.573	0.607	0.609	0.654
100	0.609	0.566	0.638	0.630	0.664
125	0.607	0.568	0.626	0.625	0.664
150	0.606	0.567	0.642	0.634	*0.672*
175	*0.620*	*0.593*	0.642	0.634	*0.672*
200	**0.621**	**0.598**	**0.660**	**0.647**	**0.678**
250	*0.617*	**0.598**	**0.654**	*0.642*	*0.673*

Table 3: Averaged effectiveness of query generation weighting (GEN) for 46 single term queries for different window sizes

The results of the window size experiments demonstrate the robust nature of the relevance profiling process, in that high and broadly comparable levels of performance are achieved across a range of window sizes. However, statistically speaking, small window sizes (50, 75) yield significantly better results for non-interpolated precision and R-precision, for multi-term queries. Nevertheless, it is clear that relevance profiling works well for windows of a few sentences in length up to half a page. For single term queries, the best effectiveness is achieved for larger window sizes (200, 250). A possible explanation for this is that, in order to establish relevance for a short query, we may require a larger sample of text

Contrary to expectations, small window sizes did not seem to be precision-enhancing, nor large window sizes recall-enhancing. If one looks in detail at the relevance profiling process (Figure 1), one can observe both precision- and recall-enhancing devices. By insisting that pages can only achieve a score if at least one query term is actually present in the page, we are effectively boosting precision. On the other hand, the fact that any window beginning in a page can contribute an RSV to that page is a recall-enhancing device. We would argue that these devices operate to mitigate the effects of window size, and particularly the possible deleterious effect of large window sizes on precision.

4.2 Weighting Function Experiments

In this second set of experiments, we investigate the comparative effectiveness of the three weighting functions described in section 2.

In Table 4, we report the best levels of effectiveness achieved for each weighting function, where the window size is chosen optimally for each function (and in some cases measure).

	Standard		
Function	GEN	KL	FREQ
Window Size	75	50	75
Average Precision	**0.662**	0.575	0.536
Average R-precision	**0.579**	0.460	0.430
F (α=0.8)	**0.704**	0.630	0.542
F (α=0.5)	**0.702**	0.626	0.550
F (α=0.2)	**0.757**	0.698	0.636

Table 4: Averaged effectiveness of three weighting functions for 232 multi-term queries (using optimal window setting for each function)

	With coordination filtering		
Function	GEN	KL	FREQ
Window Size (unless stated otherwise)	200	75	75
Average Precision	**0.607**	*0.595*	*0.598*
Average R-precision	**0.544 (75)**	*0.533*	*0.539*
F (α=0.8)	**0.677**	*0.669 (100)*	0.646
F (α=0.5)	**0.661**	*0.654 (100)*	0.638
F (α=0.2)	**0.696**	*0.690 (200)*	*0.685 (200)*

Table 5: Averaged effectiveness of three weighting functions with co-ordination level filtering for 232 multi-term queries (using optimal window setting for each function)

Let us consider the 'standard' results first. Clearly, the performance of the query generation model (GEN) exceeds that of the other two weighting functions by a large and statistically significant margin. It is approximately 10%-15% better than the Kullback-Leibler model, and above 20% better than the term frequency weighting model. Although KL weighting outperforms FREQ, it is not significantly better. It would appear that the ostensibly simple query generation model is better than the arguably more information-rich KL model, and we surmised earlier the problem of estimating parameters from small samples may be important in relevance profiling.

If we look at the individual term weights in the KL model (eqn 4), it is clear that within the *log* term, $p(t|D) << p(t|W)$, and thus the contribution of $p(t|D)$ dominates the expression. Given the nature of a book index, we believe that many terms in the index will be of similar frequency, and thus the $P(t|D)$ will be both small and relatively constant for query terms. Consequently, the term outside the *log*, namely $p(t|W)$ will provide the overall "shape" of the weighting function, and this is the same basis for the term frequency weighting (eqn 6). Hence, the performance of KL and FREQ should be, and indeed are, broadly similar.

How then do we explain why the performance of GEN is so much better than the other two weighting functions? We think it is the treatment of the non-occurrence of query terms in a window that is the distinguishing hallmark of GEN. In GEN, if a query term does not appear in the window, then the weighting function penalises this harshly. Consider that in the probability mixture (eqn 1), the contribution from the document model will always be very small. Whereas, in the other two functions the weighting is more or less linear in the number of term occurrences. To test this conjecture, we ran a third set of experiments, in which we added an additional filter, which we dubbed the coordination level filter. In these experiments we insist that all query terms are present in a window (or "coordinated" as it used to be known), prior to a retrieval status value being calculated for that window.

The results of this third set of experiments are presented in Table 5. In some cases, there was a different optimal window size for some measures, and this is included in brackets with the measure value. We expected that co-ordination filtering will reduce the number of pages retrieved, and hence likely to depress recall, and possibly boost precision. For GEN, performance was reduced by about 5-10% for all measures. However, the co-ordination filtering boosted the performance of both KL and FREQ to comparable levels to the filtered GEN performance. This provides strong evidence that it is the treatment of the non-occurrence of query terms which is the key to the performance of the unfiltered (or standard) query generation model. It may be that the performance of the KL model could be improved by experimenting with the parameter estimation rules. Interestingly, incorporating co-ordination filtering, results in generally larger optimal window sizes for GEN and to some extent KL.

4.3 Relevance Profiling for Book Indexing

Here, we draw together the evidence that relevance profiling might be used as an aid in generating a book index, namely through intellectual effort by a human, or indeed as a complete replacement for the book index.

First, the absolute effectiveness achieved when using the query generation model was very high, for both the multi-term and single term sets of queries. For the multi-term queries (Table 3), it achieved optimal F-values of 0.703 (F, α=0.5), which effectively means that at this optimum point, on average both Precision and Recall achieved levels around 70%. Moreover, the R-precision value of 0.58 is quite high, and means if a user inspects down to the rank corresponding the exact number of relevant pages for each query, then on average 58% of these relevant pages will be retrieved.

High recall is clearly important in the context of book indexing. We also computed the number of queries that achieved recall of 100% at a cutoff of 20 retrieved pages. For GEN (window size = 75), 175 of the 232 multi-term queries achieved maximum recall, and only 9 of the 232 queries failed to retrieve any relevant documents. Further, GEN (window size = 75) retrieves 587 of the possible 766 relevant pages for the multi-term queries, and 100 of the possible 144 relevant pages for the single term queries. It seems reasonable to conclude that a tool such as ProfileSkim, which implements relevance profiling, would prove a useful aid to a human indexer. Indeed, given a set of index entries, relevance profiling might even replace the traditional book index.

It is worth noting that careful inspection of the existing book index showed conclusively that the index is not completely exhaustive. In particular, there is inconsistent treatment of page ranges, where frequently not all relevant pages in a range are included in the index. Relevance profiling was able to reliably identify many of these unidentified "relevant" pages.

5 Conclusions and Future Work

In this paper we have reported a series of bench experiments of relevance profiling. The experiments were conducted with an electronic book and its associated book index. We investigated three alternative

weighting functions, and various parameters of the relevance profiling process, most notably window size.

The major findings of the experiments are:

- relevance profiling is a robust process yielding acceptably high levels of performance over a range of window sizes from 75-200 words in length;
- the query generation model (GEN) is significantly more effective than the Kullback-Leibler (KL) and the term frequency (FREQ) models;
- a key factor in effectiveness of query generation model is its treatment of non-occurrences of query terms in the sliding window, which we attribute to the smoothing provided by the probability mixture model; and
- the absolute performance of relevance profiling indicates that it might be usefully employed as an aid to human book indexing, and indeed may provide a viable alternative to the book index (providing that a good set of index entries can be generated).

We would emphasise that these experiments have been conducted with a single book, and associated index, and it would be highly desirable to repeat the experiments with a number of such books. Nevertheless, we believe that the general conclusions would likely be confirmed by such experiments, albeit the absolute levels of effectiveness may differ, depending on the exhaustivity of the (human) indexing.

The results reported here suggest a number of possible avenues for future work. First, given the evident, and indeed widely recognised, importance of parameter estimation and smoothing in language modelling applications, and given the specialised nature of relevance profiling, further experiments should be conducted on both GEN and KL using different parameter estimation rules or smoothing approaches. Second, all the weighting functions we investigated assumed that query terms are distributed independently. Given that, in general, this is unlikely to be the case, and especially for phrasal queries, it would be desirable to investigate models based on term dependence. However, we would suggest only first-order models be investigated given the paucity of sample data.

Finally, it would be interesting to explore how to generate "useful" queries automatically, where by useful we mean the query identifies coherent chucks of relevant text. We note that such queries would not necessarily be just phrasal. This would pave the way for both automating the process of book indexing, and perhaps more tantalizingly, enabling query-less retrieval within documents.

Acknowledgements

This work was done within the Smart Web Technologies Centre that was established through a Scottish Higher Education Funding Council (SHEFC) Research Development Grant (No. HR01007). David Lee was supported by a Summer Studentship provided by The Robert Gordon University. We thank the following colleagues for their feedback on this work: Ian Pirie, Ivan Koychev, Bicheng Liu, Ralf Bierig and Murat Yakici. I also thank Margery Murray for her invaluable assistance in preparing the manuscript. Finally, we would like to thank the anonymous referees for their insightful comments on the submitted paper.

6 References

[1] Kaszkiel, M. and Zobel, J.: Passage Retrieval Revisited. In: *Proceedings of the Twentieth International ACM-SIGIR Conference on Research and Development in Information Retrieval*, Philadelphia, July 1997. ACM Press, 178-185, 1997.

[2] Hearst, M. A.: TileBars: visualization of term distribution information in full text information access. *Proc. CHI'95*, 56-66, 1995.

[3] Harper, D. J., Coulthard, S., and Sun, Y. A language modelling approach to relevance profiling for document browsing. In *Proceedings of the Joint Conference on Digital Libraries*, pages 76-83, Oregon, USA, July 2002.

[4] Harper, D.J., Koychev, I. and Sun, Y. Query-based document skimming: A user-centred evaluation of relevance profiling. In: *Proceedings of 25th European Conference on Information Retrieval.* Lecture Notes in Computer Science, Springer-Verlag, Berlin, pp. 377-392, 2003.

[5] Harper, D.J., Koychev, I., Sun, Y. and Pirie, I. Within-Document Retrieval: A User-Centred Evaluation of Relevance Profiling. In: *Information Retrieval,* Kluwer Academic Publishers, The Netherlands, 7, 265-290, 2004.

[6] Ponte, J. and Croft, W. B.: A language modeling approach to information retrieval. In: *Proceedings of the 1998 ACM SIGIR Conference on Research and Development in Information Retrieval*, 275-281, 1998.

[7] Song, F. and Croft, W.B.: A general language model for information retrieval. In: *Proceedings of the 1999 ACM SIGIR Conference on Research and Development in Information Retrieval*, 279-280, 1999

[8] Croft, W.B. and Lafferty, J. (Eds.): Language modeling for information retrieval. Kluwer Academic Publishers, The Netherlands, 2003.

[9] Van Rijsbergen, K. *Information Retrieval*. Butterworths, London, 1979.