CHALMERS | GÖTEBORGS UNIVERSITET

PREPRINT

Experimenting with Different Weighting Schemes for the Krylov Subspace Method Used for Information Retrieval

KATARINA BLOM

Department of Mathematics
CHALMERS UNIVERSITY OF TECHNOLOGY
GÖTEBORG UNIVERSITY
Göteborg Sweden 2004

Preprint 2004:19

Experimenting with Different Weighting Schemes for the Krylov Subspace Method Used for Information Retrieval

Katarina Blom

CHALMERS | GÖTEBORGS UNIVERSITET

Mathematics
Department of Mathematics
Chalmers University of Technology and Göteborg University
SE-412 96 Göteborg, Sweden
Göteborg, April 2004

NO 2004:19
ISSN 0347-2809

Matematiska Vetenskaper
Göteborg 2004

Experimenting with different weighting schemes for the Krylov subspace method used for Information Retrieval

Katarina Blom

April 12, 2004

Abstract

In a previous report we have described how simple Krylov subspace methods can be used for information retrieval. We used the Golub Kahan bidiagonalization procedure to generate an approximation to a low rank representation of the documents. The process is query based and a new approximation is made for every new query. The Krylov method often shows better retrieval performance than the raw vector model (where documents are scored measuring angles between the query and the original documents).

In this report we explore the effects of 107 different combinations of term weighting schemes for the term document matrix together with 27 different weighting schemes for the queries in each of four test collections. Also, for each weighting performance of three similarity measures for the Krylov method are compared to performance of the vector model, and for the best and worst performing weighting combination for each set also with the LSI.

Our results are rather consistent with results from similar experiments carried out previously.

There is a large difference in performance between the best performing weighting scheme and the worst performing weighting scheme.

There is no overall best weighting, but in general using a term weighting based on the distribution of a term within the whole collection improved performance.

A weighting that is bad for the Krylov subspace method is also bad for the vector model and the LSI.

keywords Krylov subspace, Information Retrieval, weighting experiment, LSI.

1 Introduction and Summary

In a previous report [5] we have described how simple Krylov subspace methods can be used for information retrieval. We used the Golub Kahan bidiagonalization procedure to generate an approximation to a low rank representation of the documents. The process is query based and a new approximation is made for every new query. The Krylov method often shows better retrieval performance than the raw vector model (where documents are scored measuring angles between the query and the original documents).

This report investigates the effect on retrieval performance when different term weightings are used. Simple weighting schemes are constructed using the one-norm, euclidean norm and max-norm. These simple weightings are compared to more sophisticated weighting schemes such as inverse document frequency and the entropy weighting. The weighting schemes used for this report are presented in section 2.

For the Krylov subspace method three similarity measures for scoring documents, the LSI-like measure ($c^{(1)}$), the expanded query measure ($c^{(2)}$) and the subspace projection measure($c^{(3)}$) are compared to the vector model (c). For the best and worst performing weighting combination for each set (performance is measured in average precision) we compare the LSI [4],[9] with the expanded query measure and the vector model. A short summary of the similarity measures and of the Krylov subspace method is given in section 3.

In order to make our experiments comparable to several other similar weighting experiments in the past, we use the four data sets Adi, Cisi, Cran and Med. These sets are old and rather small. The data sets are presented in Appendix B.

In section 4 the numerical results are presented. We explore the effects of 2889 different weighting combinations for the term document matrix and the query vectors in each of four test collections.

Our results are in general consistent with similar experiments carried out previously by Dumais [8], Salton et al [18], Kolda et al. [16],[15], see also Harman [13]. A few trends can be observed.

1. There is a large difference in performance between the best performing weighting scheme and the worst performing weighting scheme. Performance is measured in average precision.

2. There is no overall best weighting for all similarity measures and all four test sets, but in general using a term weighting based on the distribution of a term within the whole collection improves performance.

3. It seems important which query weighting is chosen (or at least the combination of term weighting and query weighting seems to be important).

4. In general the more sophisticated weighting schemes give better performance compared to the simpler vector norm weightings. But the euclidean norm is not far behind. The one-norm weighting in general decreases performance and should not be used.

5. A weighting that is bad for any of the Krylov subspace similarity measures (the LSI-like measure, the expanded query measure and the subspace projection measure) is also bad for the vector model and the LSI.

Notations The notations used in this report are rather standard in the Numerical Linear Algebra community. We use upper case letters for matrices and lower case letters for vectors. Lower case Greek letters usually denotes scalars. Component indices are denoted by subscript. For example, a vector c and a matrix M might have entries c_i and m_{ij} respectively. On the occasions when both an iteration index and a component index are needed, the iteration is indicated by a parenthesised superscript, as in $c_j^{(r)}$ to indicate the jth component of the rth vector in a sequence. Otherwise c_j may denote either the jth component of a vector c or the jth column of a matrix C. The particular meaning will be clear from its context.

The pseudo inverse of a matrix B is denoted B^+.

All of the vector norms we will use are instances of p-norms, which for a real $p \geq 1$ and a vector x of dimension n are defined by

$$\|x\|_p = (\sum_{i=1}^{n} |x_i|^p)^{1/p}.$$

The special cases we use are
one-norm:

$$\|x\|_1 = \sum_{i=1}^{n} |x_i|$$

euclidean norm:

$$\|x\|_2 = (\sum_{i=1}^{n} |x_i|^2)^{1/2}$$

and *max-norm*

$$\|x\|_\infty = \max_i |x_i|.$$

All norms on $\mathcal{R}^n$ are equivalent, i.e. if $\|\cdot\|_\alpha$ and $\|\cdot\|_\beta$ are p-norms on $\mathcal{R}^n$, then there exist positive constants c_1 and c_2 such that

$$c_1\|x\|_\alpha \leq \|x\|_\beta \leq c_2\|x\|_\alpha \tag{1}$$

A *Krylov subspace* of a square matrix M, starting at the vector v, is a subspace of the form

$$\mathcal{K}_r(M, v) = \text{span}\{v, Mv, M^2v, \ldots M^{r-1}v\}.$$

Measures The retrieval efficiency of an information retrieval system depends on two main factors. The ability of the system to retrieve relevant information and the ability to dismiss irrelevant information. The ability to retrieve relevant information is measured by *recall*, the ratio of relevant documents retrieved over the total number of relevant documents for that query. A systems ability to reject irrelevant documents is measured by *precision*, the ratio of the number of relevant documents retrieved for a given query over the total number of documents retrieved. Precision and recall are usually inversely related (when precision goes up, recall goes down and vice versa).

When we evaluate a query q all documents in the set are ranked and we receive an ordered list $\mathcal{L}$ of documents. Assume t documents are relevant to the query and let ℓ_i, $i = 1 \ldots t$ be the position for the ith relevant document in $\mathcal{L}$. The *average precision* (non interpolated) for a single query is defined as

$$\frac{1}{t}\sum_{i=1}^{t} \frac{i}{\ell_i}.$$

The *mean average precision* for multiple queries is defined as the mean of the average precisions for all queries.

Precision can be computed at any *actual recall level* $\frac{i}{t}$, $i = 1 \ldots t$ (where t is the number of relevant documents to the query). Let r_j be the jth recall level from the 11 *standard recall levels* $0, 0.1, 0.2 \ldots 1$. The *interpolated average precision* for a query at standard recall level r_j is the maximum precision obtained for any actual recall level greater that or equal to r_j.

The *Recall level precision averages* for multiple queries are the means of the interpolated average precision values at each (standard) recall level for the queries. Recall level precision averages are used as input for plotting the recall-precision graphs.

For further details, see Harman [14].

2 The term document matrix

In vector space models both queries and documents are encoded as vectors in m-dimensional space. The choice m is the number of unique terms in the collection. The documents are stored as columns in a $m \times n$ *term document matrix* A. The elements in A are the occurrences of each word in a particular document, i.e.

$$A = [a_{ij}]$$

where a_{ij} is nonzero if term i occurs in document j, zero otherwise.

A term weight has three components; local, global and normalization [18]. *Local weights* are used to transform the term's frequency within the document. Each term in the collection is assigned a *global weight* to indicate its importance as an indexing term. A *normalization factor* is used to normalize the documents. We let

$$a_{ij} = g_i l_{ij} d_j$$

where l_{ij} is the local weight for term i in document j, g_i is the global weight for term i and d_j is the document normalization factor.

Specifically we can write

$$[a_{ij}] = A = GLD \tag{2}$$

where the elements in $L = [l_{ij}]$ are the local weights. G and D are diagonal matrices and g_{ii} in G is the global weight for term i and d_{jj} in D is the normalization factor for document j. The global weighting correspond to a *row scaling* of the term document matrix and the normalization corresponds to a *column scaling*.

There are several local and global weightings that can be used. For nice summaries see for example Frakes and Baeza-Yates [11], Salton and McGill [19] or Kolda [15].

The queries are stored the same way as the document vectors, that is

$$q = [q_i]$$

where q_i is nonzero if term$_i$ appears in the query. As for the elements in the term document matrix local and global weightings are used. The local weights are computed using the term frequency within the query vector and the global weights are computed from the frequency counts in the documents. Normalizing the query makes no difference when ranking the documents and is not used [1].

For convenience [16], let

$$\chi(x) = \begin{cases} 1 & \text{if } x > 0 \\ 0 & \text{ir } x = 0 \end{cases}$$

Various combinations of weights are used for the documents in the term document matrix and for the queries. Each term weight combination is described using two three letter strings, representing the weightings for the term document matrix (first triple) and the query terms (second triple). The letters in each string represent the local, global and normalization component respectively.

Formulas and symbols for the weightings used for this report are shown in tables 1 - 3.

For example the classical idf weight [18] is described by the string

$$\texttt{bfx} \cdot \texttt{bfx},$$

which implies the local, global and normalization components

$$\begin{aligned} l_{ij} &= \chi(\mathrm{tf}_{ij}) \\ g_i &= \log_2(\frac{n}{\mathrm{df}_i}) \\ d_j &= 1 \end{aligned}$$

for the elements in the term document matrix. The *term frequency* tf_{ij} is the the number of times term i appears in document j, and the *document frequency* df_i is the number of documents to which term i is assigned. The local global and normalization components for the query vector elements are

$$\begin{aligned} l_i &= \chi(\mathrm{tf}_i) \\ g_i &= \log_2(\frac{n}{\mathrm{df}_i}) \\ d_j &= 1. \end{aligned}$$

Here tf_i is the term frequency for the terms in the query (i.e. the number of times term i appears in the query) and df_i is the document frequency for term i in the collection.

The binary local weighting (`b`) and the local frequency weighting (`t`) listed in table 1 are simple but with some major drawbacks. The binary weighting gives every word that appear in a document equal relevance. (This might be useful when the number of times a word appears is not considered important.)

The local frequency weighting give more credit to words that appear more frequently which might serve the recall function. For example a term such as melon appearing with reasonable frequency in some documents indicates that they deal with melons. The assignment of the term melon with high weight will then help to retrieve these documents in response to appropriate queries.

On the other hand, high precision implies high ability to distinguish individual documents from each other (to be able to prevent unwanted retrievals), therefore when common terms are not concentrated in a few documents but instead are spread out in the whole collection, precision is likely to drop.

More concretely, if the whole document collection deals with melons, almost all documents will contain the term melon many times, giving high credit to melon, will not help to identify the wanted subset of documents.

For a more detailed discussion see for example Salton [17] (or Salton and Buckley [18]).

The (local) augumented normalized term frequency (`n`) will give basic credit (0.5) to any word that appears and then give additional credit to words that appear more frequently.

The logarithmic weight (`l`) will deemphasize the effect of high frequency.

The choice for local weightings depends on the vocabulary used for the collection. Some general recommendations can be made [3]. Local binary term weighting schemes are recommended for sets where the term list (the number of rows in the term document matrix) is short. The local frequency

[1] In the Krylov subspace method used in this report (section 3 gives a short summary) the query vector is always normalized using euclidean norm before the bidiagonalization procedure is started.

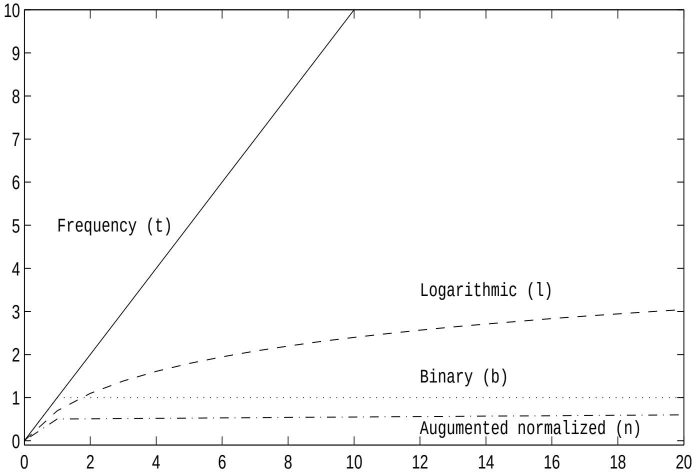

Figure 1: Comparison of local term weighting schemes. The term frequency range from 0 to 20 (x-axis) and the weights (y-axis) range from 0 to 10 in the figure.

weighting is recommended for varied vocabularies, eg. popular magazines, and the augumented normalized term frequency is recommended for technical or scientific vocabularies.

The four local weightings are compared in figure 1. The term frequency range from 0 to 20. The raw frequency grows very quickly compared to the other local weightings grow more slowly.

As mentioned above, precision might be better served by using very specific terms that will match the most relevant documents in the collection, because such terms are able to distinguish the few documents in which they appear from the many from which they are absent. All of the global weighting schemes in table 2 (except `x`) give less weight to frequent terms. So in order to fulfill both the requirements of high recall and of high precision, i.e. to credit those terms that occur frequently in individual documents but rarely in the remainder of the collection, the combination of local term frequency weighting and any of the global weightings may be used.

The global weightings `n`, $\mathtt{n}_1$ and $\mathtt{n}_\infty$ are based on simple vector norms and will normalize the length of each row in the term document matrix in some norm. This has the effect of giving high weight to infrequent terms. If a few rare terms appear frequently in only a few documents the max-norm is giving the most credit to these terms, followed by the euclidean norm and then the one-norm.

The entropy global weighting (`e`) uses concepts from information theory. In information theory the least predictable terms in a running text, those exhibiting the smallest probabilities, carry the greatest information value. The weighting assign weights between zero and one. Zero for a term appearing with the same frequency in every document and one for a term that appears only once.

The weights given by the different global schemes to two different terms in a collection are compared in figure 2. For both terms the local term frequency (`t`) was used. The term in the upper plot appears once in one document and three times in another. The term is rare in the set and all of the global weighting schemes give high credit to the term in the two documents where it appears. In the lower plot a term appears in all but one document. This term is common in the set. The global schemes will not emphasize the appearance of the term as they did for the rare term. All the weighting schemes give slightly more credit to the term in the document where it appears three times.

The normalizing factors will normalize the length of each column in the term document matrix. This has the effect of giving higher weights to all terms in short documents and giving lower weights to all terms in long documents. If using the angles between the query and the document vectors in the therm document matrix when ranking documents for relevancy there is a tendency that shorter documents will be ranked more relevant than longer documents. In order to retrieve documents of a certain length with the same probabilities the pivoted cosine normalization scheme has been proposed for indexing the TREC collection [6], [20].

In this report we always apply first the local weighting, then the global weighting and at last the normalization factor. For example, the matrix weighting `tnc` corresponds to first normalizing the rows in the term document matrix using euclidean norm, then normalizing the columns (using euclidean norm). Note that the column normalization might destroy the previous row normalization, but not completely. Some deemphasizing effect on common terms still remain.

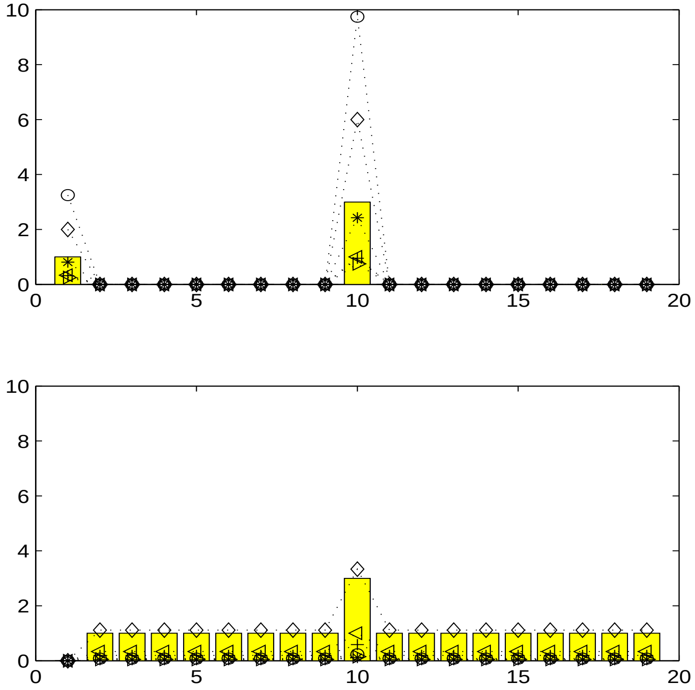

Figure 2: Comparison of global term weighting schemes when the local term frequency weighting (`t`) is used. The bars are the term frequencies. The global weighting schemes are: no weighting `x` (the bars), inverse document frequency `f` (○), GfIdf `g` (◇), entropy `e` (∗), normal `n` (+), one-norm $\mathtt{n}_1$ (▷) and max-norm $\mathtt{n}_\infty$ (◁).

Local Weighting		Description
`b`	$\chi(\text{tf}_{ij})$	*Binary weight* [18] equal 1 for terms present in vector, zero otherwise. The term frequency tf_{ij} is the number of times term i appears in document j.
`t`	tf_{ij}	*Raw frequency weight* [18] is number of times a term appears in a document or a query.
`l`	$\log_2(1+\text{tf}_{ij})$	*Logarithmic weight* [8][13] takes the log of the term frequency, thus dampening effects of large differences in frequencies.
`n`	$\frac{1}{2}\left(\chi(\text{tf}_{ij})+\frac{\text{tf}_{ij}}{\max\limits_k(\text{tf}_{kj})}\right)$	*Augumented normalized term frequency* [18][13]. The term frequency tf_{ij} is normalized by maximum appearance of term in document j and further normalized to lie between 0.5 and 1.0[2].

Table 1:

[2] A more general formula was proposed by Croft [7]. The formula was parameterized by a value K (a sliding importance factor), $t_{ij} = \chi(\text{tf}_{ij})\text{K} + (1-\text{K})\frac{\text{tf}_{ij}}{\max\limits_k(\text{tf}_{kj})}$. It is suggested that K be low for large documents and high for short documents.

[3] In [8] $l_{ij} = \text{tf}_{ij}$

[4] In [18] $l_{ij} = \text{tf}_{ij}$.

Global Weighting		Description
x	1	No change in weight [18].
f	$\log_2(\frac{n}{\text{df}_i})$	*Inverse document frequency* (Idf) [18] where n is number of documents in collection and df_i is the document frequency (the number of documents to which term i is assigned).
g	$\frac{\text{gf}_i}{\text{df}_i}$	*GfIdf* [8]. gf_i is the global frequency (the total number of times term_i appears in the whole collection). df_i is the document frequency.
e	$1 - \sum_{j=1}^{n} \frac{p_{ij}\log(\frac{1}{p_{ij}})}{\log(n)}$	*Entropy* [8][13]. n is number of documents in collection and $p_{ij} = \frac{\text{tf}_{ij}}{\text{gf}_i}$ where tf_{ij} is the raw term frequency and gf_i is the global frequency.
n	$\frac{1}{\sqrt{\sum_j l_{ij}^2}}$	*Normal* [8], where l_{ij} is received after applying any of the local weightings presented in table 1 [3].
$\mathtt{n}_1$	$\frac{1}{\sum_j l_{ij}}$	where l_{ij} is received after applying any of the local weightings presented in table 1.
$\mathtt{n}_\infty$	$\frac{1}{\max_j l_{ij}}$	where l_{ij} is received after applying any of the local weightings presented in table 1.

Table 2: .

Normalization factor		Description
x	1	No normalization factor is used [18].
c	$\frac{1}{\sqrt{\sum_i (g_i l_{ij})^2}}$	*Cosine normalization* [18] [4].
$\mathtt{n}_1$	$\frac{1}{\sum_i g_i l_{ij}}$	
$\mathtt{n}_\infty$	$\frac{1}{\max_i g_i l_{ij}}$	

Table 3: The local weightings l_{ij} and global weightings g_i are received after applying any of the local and global weightings respectively presented in tables 1 and 2

3 The Krylov subspace method for Information retrieval

Query matching can be viewed as a search in the column space of the term document matrix A. One of the most common similarity measures used for query matching is to measure the angle between the query vector and the document vectors in A. The smaller the angle is the more relevant the document is. In the *vector model* the cosines between the query vector q and document vectors a_j are used to score the documents in relevance order,

$$c_j = \frac{q^T a_j}{||q||_2 ||a_j||_2}, \quad j = 1, \ldots, n. \tag{3}$$

For the Krylov subspace methods we will use the Golub Kahan bidiagonalization procedure [12] applied to the term document matrix A starting with the query vector q to receive the two *basis matrices* Q_{r+1} and P_r and the $r+1 \times r$ lower bidiagonal matrix B_{r+1}:

$$[Q_{r+1}, B_{r+1}, P_r] = \textsc{Bidiag}(A, q, r) \tag{4}$$

The column vectors in the basis matrices Q_{r+1} and P_r span bases for the two Krylov subspaces $\mathcal{K}_{r+1}(AA^T, q)$, in the document space (spanned by the query q and the columns of A) and $\mathcal{K}_r(A^TA, A^Tq)$, in the term space (spanned by the rows of A) respectively. We let the *reached subspace* W form an orthonormal basis for the column vectors in AP_r.

The Bidiag procedure is further described in section 3.1.

The reached subspace W, the basis matrices Q_{r+1}, P_r and the B_{r+1} matrix are used to score the documents in relevance order to the query (see Blom Ruhe [5]). A few examples of similarity measures are:

- For the *subspace projection measure* the documents in A are sorted according to their closeness measured in angles to the Krylov subspace $\mathcal{K}_{r+1}(AA^T, q)$. The closer the document is the more relevant the document is. The cosine of the angle between the basis matrix Q_{r+1} for the Krylov subspace $\mathcal{K}_{r+1}(AA^T, q)$ and each document vector in A

  $$c_j^{(3)} = ||Q_{r+1}^T a_j||, \; j = 1, \ldots, n, \tag{5}$$

  is used to sort the documents. Note that for $r = 0$ in the Bidiag procedure the subspace projection measure is simply the vector model scoring (3).

- A projected query vector

$$\hat{q} = WW^T q \tag{6}$$

is constructed using the reached subspace. In the *expanded query measure* the documents are sorted measuring the angle between $\hat{q}$ and each document vector in A,

$$c_j^{(2)} = \frac{\hat{q}^T a_j}{||a_j||}, \; j = 1, \ldots, n. \tag{7}$$

In the *LSI-like measure* we mimic the LSI [5] and the documents are scored measuring the angle between $\hat{q}$ and each projected document vector in A

$$c_j^{(1)} = \frac{\hat{q}^T a_j}{||W^T a_j||}, \; j = 1, \ldots, n. \tag{8}$$

The smaller the angle the more relevant the document is. Note that if the starting vector $q \in \mathcal{R}(A)$ then the projected query $\hat{q} = q$ and the cosines (7) is simply the vector model scoring (3).

3.1 The Golub Kahan bidiagonalization procedure

The Golub Kahan bidiagonalization procedure is a variant of the Lanczos tridiagonalization algorithm and it is widely used in the numerical linear algebra community.

The Golub Kahan algorithm starts with the normalized query vector $q_1 = q/||q||$, and computes two orthonormal bases P and Q, adding one column for each step k, see [12] in section 9.3.3.

```
ALGORITHM BIDIAG($A$,$q$,$r$):
Start with $q_1 = q/||q||$, $\beta_1 = 0$
for $k = 1, 2, \ldots r$ do
    $\alpha_k p_k = A^T q_k - \beta_k p_{k-1}$
    $\beta_{k+1} q_{k+1} = A p_k - \alpha_k q_k$
end.
```

[5] In LSI, Berry et al [4], Dumais et al [9], see also Berry and Brown [3], the $m \times n$ term document matrix is represented using a rank-k approximation, $k < \min(m, n)$, from the singular value decomposition of A. Documents are scored measuring the angles between the query and the column vectors in the approximation.

The scalars α_k and β_k are chosen to normalize the corresponding vectors.

Define

$$\begin{aligned} Q_{r+1} &= \begin{bmatrix} q_1 & q_2 & \ldots & q_{r+1} \end{bmatrix}, \\ P_r &= \begin{bmatrix} p_1 & p_2 & \ldots & p_r \end{bmatrix}, \end{aligned}$$

$$B_{r+1} = \begin{bmatrix} \alpha_1 & & \\ \beta_2 & \alpha_2 & \\ & \ddots & \alpha_r \\ & & \beta_{r+1} \end{bmatrix}.$$

After r steps k we have the basic recursions

$$\begin{aligned} A^T Q_r &= P_r B_r^T \\ A P_r &= Q_{r+1} B_{r+1}. \end{aligned}$$

The columns of Q_r will be an orthonormal basis of the Krylov subspace $\mathcal{K}_{r+1}(AA^T, q)$ and the columns of P_r forms an orthonormal basis for the Krylov subspace
$\mathcal{K}_r(A^T A, A^T q)$. The lower bidiagonal matrix $B_{r+1} = Q_{r+1}^T A P_r$ is the projection of A onto these Krylov subspaces and some of the singular values of B_{r+1} will be approximations of those of A.

With r large enough the bidiagonalization procedure BIDIAG(A,q,r) can be used to compute a solution $x_L = P_r B_{r+1}^+ e_1$ for the least squares problem

$$\min_x ||Ax - q||_2.$$

Let $k < r$. The projected query vector (6) $\hat{q} = Ax^{(k)}$ where $x^{(k)} = P_k B_{k+1}^+ e_1$ is an approximation to x_L received after k iterations in the BIDIAG procedure.

3.2 Numerical aspects of using weighting schemes

Let $A = GLD$ be the term document matrix defined in (2). If no global weighting or normalization factor is used (i.e. global weighting and normalization factor `x` respectively is used) then $A = L$. Consider the least squares problem

$$\min_x ||Lx - q||_2 \tag{9}$$

A solution x_L to this problem can be obtained by using the Bidiag procedure with L and starting at q (see for example [12] or [2]).

If no global weighting (i.e. global weighting x from table 2) is used for the term document matrix then $A = LD$ where D is the $n \times n$ diagonal matrix defined in (2). Assume D is nonsingular (i.e. assume all documents has at least one term) and consider the least squares problem

$$\min_{y} ||LDy - q||_2. \tag{10}$$

Multiplying L by a diagonal matrix from the right corresponds to a column scaling of L and the solution x_L to problem (9) can be obtained by finding the minimum 2-norm solution y_L to problem (10). If rank(L)$= n$ then $x_L = Dy_L$ otherwise Dy_L is the minimum D-norm solution [6] to (9).

However it is well known that column scaling affects singular values and that the number of iterations needed in the Bidiag procedure before the solution is reached heavily depend on the distribution of singular values in the matrix that is used. When we use the Bidiag procedure for IR purposes we stop iterating after $r \leq 10$ steps, that is long before a solution to any of the least squares problems (9) and (10) is reached. This means that we cannot use the relations between x_L and y_L directly when computing scorings $c^{(1)}$and $c^{(2)}$.

Assume only global weighting is used (i.e. normalization factor x) and consider the weighted least squares problem

$$\min_{s} ||G(Ls - q)||_2 \tag{11}$$

where G is the $m \times m$ diagonal matrix with global weights defined in (2). (In equation (11) we have assumed that the terms in the query vector are weighted using the same global weighting as for the term document matrix [7]). Multiplying L by a diagonal matrix G from the left correspond to a row scaling of the matrix L (and query vector q). It is well known that row scaling affects the solution to a least squares problem [8] and there is no simple relation between the solutions to (9) and (11).

[6] D-norm is defined by $||z||_D = ||D^{-1}z||_2$.

[7] In the experiments performed (see section 4) we have also tried combinations when L and q have different global weights.

[8] An exception occurs when $q \in \mathcal{R}(L)$. In this case the solution to (9) and (11) are equal. In all test sets we tried the query vector q is never completely in $\mathcal{R}(\mathcal{A})$.

4 Experiments

For our experiments four test sets were used, Adi, Cisi, Cran and Med. The sets are further described in appendix B .

We have tried all possible combinations of local, global and normalization factors from tables 1, 2 and 3 for the term document matrix. In tables 5 and 6 all weighting combinations we used are listed. For each weighting on the term document matrix the queries were weighted using all combinations of local and global weightings listed in table 6. (The document frequencies and the global frequencies are taken from the term document matrix.). In total we explored the effect of $105 * 27 = 2889$ different weighting combinations.

Using these data sets and weighting schemes makes our experiments comparable for example with the LSI experiments made by Dumais in [8], some of the experiments made by Salton et al [18] and with the LSI and LDD experiments made by Kolda et al [16].

For each weighting combination four similarity measures were used to score the documents, the vector model c (3), the LSI-like measure $c^{(1)}$(8), the expanded query measure $c^{(2)}$ (7) and the subspace projection measure $c^{(3)}$ (5). For the Krylov subspace methods the iterations in the Bidiag procedure were stopped when maximum average precision before the number of steps $r = 10$ for each query was reached.

For the best and worst weighting combination for the expanded query measure we computed recall level average precisions for the LSI [4],[9]. For the LSI we need to chose a rank k (the number of singular vectors to use) for the low rank approximation of the term document matrix. We chose $k \leq 100$ (for Adi we let $k \leq 60$) to be the rank where where maximum mean average precision was found.

Computational Results For each weighting the number of times each of the four similarity measures gave best mean average precision is shown in figure 3. The expanded query measure $c^{(2)}$ generally give best average precision in Cran and Med. In Adi and Cisi the LSI-like measure $c^{(1)}$ gave best average precision in a little more than half of the weighting combinations. The vector model c is never the best one. Observe that since Bidiag is stopped when best average precision before the number of steps $r = 10$ is reached the subspace projection measure $c^{(3)}$ never score worse than the vector model.

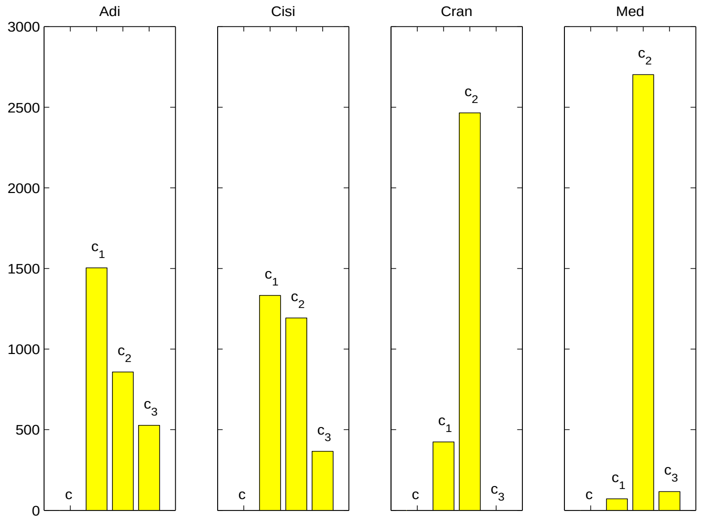

Figure 3: For each weighting the number of times each of the four similarity measures c(3) $c^{(1)}$ (8), $c^{(2)}$ (7), and $c^{(3)}$ (5) gave best mean average precision.

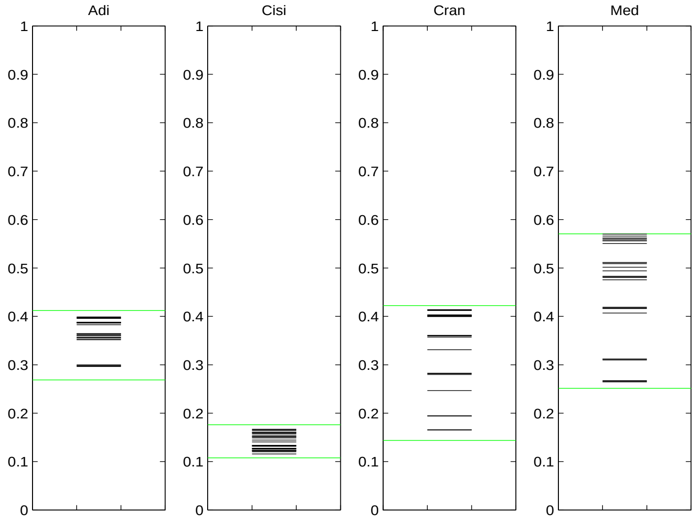

Figure 4: Comparison of mean average precisions for the vector model c (3). The weighting scheme `nxx` was used for the term document matrix. Mean average precision for each query weighting from table 6 is marked (the black lines). The grey lines are maximum and minimum mean average precision respectively for the vector model scoring in each set.

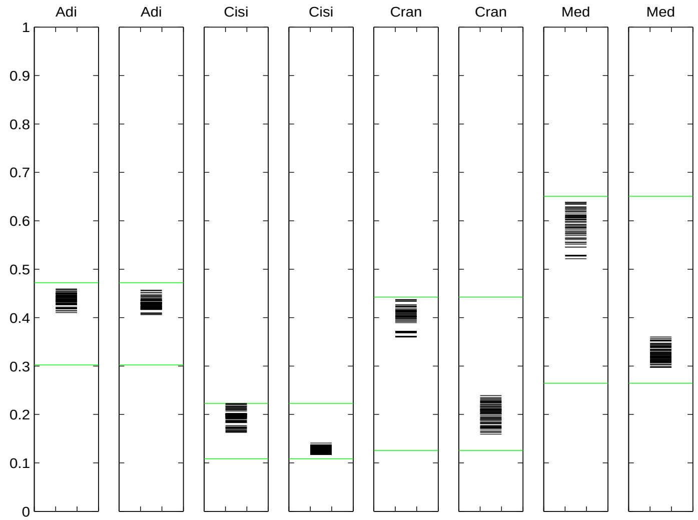

Figure 5: Comparision of mean average precisions for the LSI-like measure $c^{(1)}$ (8). In the left plot of each pair mean average precisions for weighting schemes where global entropy weighting (e) are used for both the term document matrix and the queries are marked (the black lines). In the right plot of each pair mean average precisions for weighting schemes where global one-norm weighting ($\mathtt{n}_1$) is used for both the term document matrix and the queries are marked (the black lines). The grey lines are maximum and minimum mean average precision respectively for the LSI-like measure in each set.

Adi							
c		$c^{(1)}$		$c^{(2)}$		$c^{(3)}$	
ngx·l1x	0.41	bgc·t1x	0.47	bgc·t1x	0.49	lfc·lnx	0.48
lxx·l1x	0.41	te1·n1x	0.47	bgn_∞·l1x	0.48	tec·t1x	0.48
lex·lnx	0.41	bgc·l1x	0.47	ngn_∞·t1x	0.48	tfc·t1x	0.48
lgx·l1x	0.41	bgc·tnx	0.47	len_∞·nnx	0.47	lec·lnx	0.48
nxx·n1x	0.41	txx·tfx	0.47	bgn_∞·t1x	0.47	nfc·n1x	0.47
lfx·lnx	0.41	te1·b1x	0.47	bgx·t1x	0.47	nec·l1x	0.47
nxx·l1x	0.41	nxx·n1x	0.47	nfn_∞·bnx	0.47	nfc·l1x	0.47
lgx·t1x	0.41	nxn_∞·n1x	0.47	tx1·t1x	0.47	lfc·l1x	0.47
tex·t1x	0.41	bgn_∞·lnx	0.46	bg1·t1x	0.47	lec·l1x	0.47
Cisi							
c		$c^{(1)}$		$c^{(2)}$		$c^{(3)}$	
ngx·tfx	0.17	lex·tex	0.22	tgx·l1x	0.23	nfx·tgx	0.20
ngx·lfx	0.16	lex·lex	0.22	tgx·n1x	0.23	nfx·lgx	0.20
ngx·tex	0.16	lfx·tex	0.22	tgx·b1x	0.23	nex·tgx	0.20
lgx·tfx	0.16	lex·tfx	0.22	lex·tfx	0.23	nex·lgx	0.20
ngx·nfx	0.16	lex·nex	0.22	lex·tex	0.23	nfx·ngx	0.20
lgx·lfx	0.16	lfx·lex	0.22	lex·lfx	0.23	nnx·tgx	0.19
ngx·lex	0.16	lex·lfx	0.22	lfx·tex	0.23	nex·ngx	0.19
lgx·nfx	0.16	lfx·tfx	0.22	lex·lex	0.23	nnx·lgx	0.19
ngx·nex	0.16	lfx·nex	0.22	lex·nfx	0.23	lgx·lfx	0.19
Cran							
c		$c^{(1)}$		$c^{(2)}$		$c^{(3)}$	
ngx·lfx	0.42	lfc·bgx	0.44	ngx·lnx	0.51	ngc·lfx	0.43
ngx·bex	0.42	lfc·ngx	0.44	ngx·nnx	0.51	ngc·bex	0.43
ngx·nfx	0.42	lfc·lgx	0.44	ngn_∞·nnx	0.51	ngc·lex	0.43
ngx·lex	0.42	lec·bgx	0.44	ngn_∞·lnx	0.50	ngc·nfx	0.43
ngx·bfx	0.42	lec·ngx	0.44	ngc·nnx	0.50	ngc·bfx	0.43
ngx·nex	0.42	lfc·tgx	0.44	lgx·lnx	0.50	ngc·nex	0.43
lgx·bfx	0.42	lec·tex	0.44	ngc·lnx	0.50	ngc·tex	0.43
lgx·nfx	0.42	lec·lgx	0.44	lgc·lnx	0.50	ngc·tfx	0.43
ngx·tex	0.42	lec·lex	0.44	ngc·bnx	0.50	lgc·nfx	0.43
Med							
c		$c^{(1)}$		$c^{(2)}$		$c^{(3)}$	
ngx·bex	0.57	lec·bgx	0.65	ngn_∞·bnx	0.68	lfc·bgx	0.62
ngx·bfx	0.57	lfc·bgx	0.65	ngc·bnx	0.68	ngc·bnx	0.61
ngx·nex	0.57	lec·ngx	0.65	ngx·bfx	0.68	lec·bgx	0.61
ngx·nfx	0.57	lfc·ngx	0.65	lgn_∞·bnx	0.68	ngc·bfx	0.61
lgx·bex	0.56	lec·lgx	0.64	ngx·lnx	0.68	lfc·ngx	0.61
lgx·bfx	0.56	lfc·lgx	0.64	lgc·nnx	0.68	lfc·bfx	0.61
ngx·lfx	0.56	tfc·bfx	0.64	ngc·nnx	0.68	lfc·bex	0.61
ngx·lex	0.56	tfc·bex	0.64	ngx·bnx	0.68	lec·ngx	0.61
nex·bgx	0.56	tfc·nfx	0.64	ngx·nnx	0.68	lec·bfx	0.61

Table 4: The nine best performing weighting schemes for each set and each similarity measure. Performance is measured in mean average precision. Since the vector norms used as normalization factors are equivalent (1) they have no effect for the vector model (3) and are not listed in the table.

Tables 4 show numerical results for the Adi, Cisi, Cran and Med data sets. For each test we report the mean average precision for all queries in the set.

As we can see there is no overall best weighting for all similarity measures, however a few trends can be seen. We observe that for the vector model c (3) the matrix weightings `ngx` and `lgx` give the best results for all test sets.

The binary matrix weighting `bxx` and the raw term frequency weighting `txx` combined with no global weighting for the query vector tend to be ranked towards the bottom.

For the LSI-like measure $c^{(1)}$ (8) the global entropy weighting (`e`) is good. The matrix weightings $\texttt{ten}_1$ and `lex` are good for Adi and Cisi respectively and the weighting `lec` is good for Cran and Med. But also the global inverse frequency weighting (`f`) and the GfIdf weighting (`g`) are good. The weightings `bgc` and $\texttt{bgn}_\infty$ are good for Adi. Weighting `lfc` is good for Med and Cran and weighting `lfx` is good for Cisi.

Also for the expanded query measure $c^{(2)}$ (7) the global inverse frequency weighting (`f`) and the GfIdf weighting (`g`) work well. Matrix weightings `bgc` and $\texttt{bgn}_\infty$ are good for Adi and weighting `tgx` is good for Cisi. For Cran and Med weightings `ngx` and `ngc` are good. But also the global entropy weighting (`e`) seems to work well.

Among the poor performing weighting combinations the global one-norm weighting ($\texttt{n}_1$) is frequent. And for Adi also the global max-norm weighting ($\texttt{n}_\infty$) is bad.

For the subspace projection measure $c^{(3)}$ (5) the global inverse frequency weighting (`f`) and the GfIdf weighting (`g`), but also the entropy global weighting (`e`) works well.

One trend found in weighting experiments is that the use of global weights improves performance (or at least does not hurt performance) [13]. In our experiments in general the use of global weights improves performance except when the global one-norm weighting ($\texttt{n}_1$) is used. The global one-norm weighting is bad for all sets but the Adi. In figure 5 the mean average precision for the global entropy weightings and the global one-norm weightings are compared.

For the local weightings in general the binary weighting (`b`) appear among the poor performers, however the weighting works well for the Adi. This might be due to the small size of the term document matrix

It seemed to be important which query weighting was chosen. In figure 4 we plotted mean average precision for the matrix weighting `nxx` and all

27 different query weightings listed in table 6 for all test sets. As we can see the differences in mean average precisions are large. In general using a global weight for the query vector, preferably any of entropy (`e`), inverse document frequency (`f`), GfIdf (`g`) or normal (`n`), seems to improve retrieval performance.

Our results for the vector space model are quite consistent with those reported by Kolda [15]. Somewhat surprisingly she found that it makes little difference which query weighting is chosen.

In [8] Dumais report good performance for the `lec` matrix weighting on the matrix and Salton's *best weighting* reported in [18] was `tfc`·`tfx`. In general these weightings also work well in our experiments.

The weighting `tnc`·`txx` used by Blom Ruhe in [5] is among the average (sometimes above average) performing weighting schemes.

Figure 6 are recall-precision graphs for the best and worst performing weighting combinations for the expanded query measure $c^{(2)}$ (7) in each set. For each set interpolated average precision for the vector model c (3), the expanded query measure $c^{(2)}$ and the LSI are compared. We observe that a weighting combination that is bad for the expanded query measure $c^{(2)}$ also performs poorly for the c and the LSI.

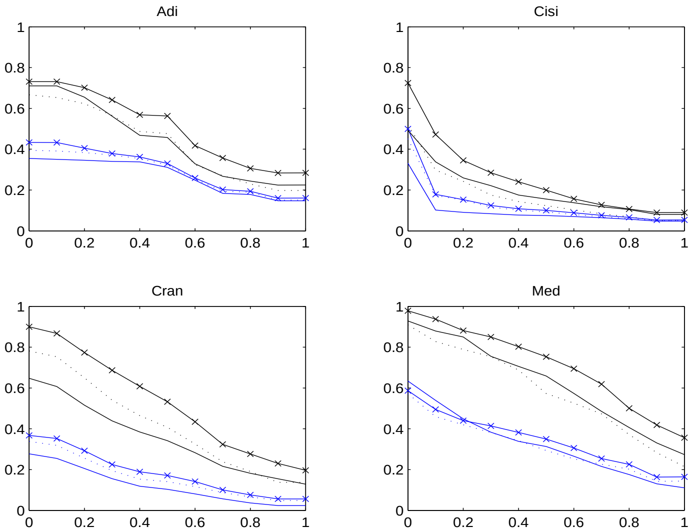

Figure 6: Recall-precision graphs for the best and worst performing weighting combinations for the expanded query measure $c^{(2)}$ (7) in each set. In each plot interpolated average precision for the vector model c (3) $(\ldots)$, the expanded query measure (x-) and the LSI [4],[9] (–) are shown.

A Weighting combinations

Local and global weigthing combinations						
bxx	bfx	bgx	bex	bnx	bn$_1$x	(bn$_\infty$x)
txx	tfx	tgx	tex	tnx	tn$_1$x	tn$_\infty$x
lxx	lfx	lgx	lex	lnx	ln$_1$x	ln$_\infty$x
nxx	nfx	ngx	nex	nnx	nn$_1$x	nn$_\infty$x
Local and normalization factor combinations						
bxc	bxn$_1$	(bxn$_\infty$)	txc	txn$_1$	txn$_\infty$	
lxc	lxn$_1$	lxn$_\infty$	nxc	nxn$_1$	nxn$_\infty$	
Local, global and normalization factor combinations						
bfc	bfn$_1$	bfn$_\infty$	bgc	bgn$_1$	bgn$_\infty$	
bec	ben$_1$	ben$_\infty$	bnc	bnn$_1$	bnn$_\infty$	
bn$_1$c	bn$_1$n$_1$	bn$_1$n$_\infty$	(bn$_\infty$c)	(bn$_\infty$n$_1$)	(bn$_\infty$n$_\infty$)	
tfc	tfn$_1$	tfn$_\infty$	tgc	tgn$_1$	tgn$_\infty$	
tec	ten$_1$	ten$_\infty$	tnc	tnn$_1$	tnn$_\infty$	
tn$_1$c	tn$_1$n$_1$	tn$_1$n$_\infty$	tn$_\infty$c	tn$_\infty$n$_1$	tn$_\infty$n$_\infty$	
lfc	lfn$_1$	lfn$_\infty$	lgc	lgn$_1$	lgn$_\infty$	
lec	len$_1$	len$_\infty$	lnc	lnn$_1$	lnn$_\infty$	
ln$_1$c	ln$_1$n$_1$	ln$_1$n$_\infty$	ln$_\infty$c	ln$_\infty$n$_1$	ln$_\infty$n$_\infty$	
nfc	nfn$_1$	nfn$_\infty$	ngc	ngn$_1$	ngn$_\infty$	
nec	nen$_1$	nen$_\infty$	nnc	nnn$_1$	nnn$_\infty$	
nn$_1$c	nn$_1$n$_1$	nn$_1$n$_\infty$	nn$_\infty$c	nn$_\infty$n$_1$	nn$_\infty$n$_\infty$	

Table 5: Weighting combinations used for the term document matrices. The weightings surronded by parentheses have no effect and are not used.

Local and global weighting combinations						
bxx	bfx	bgx	bex	bnx	$\mathtt{bn}_1\mathtt{x}$	$(\mathtt{bn}_\infty\mathtt{x})$
txx	tfx	tgx	tex	tnx	$\mathtt{tn}_1\mathtt{x}$	$\mathtt{tn}_\infty\mathtt{x}$
lxx	lfx	lgx	lex	lnx	$\mathtt{ln}_1\mathtt{x}$	$\mathtt{ln}_\infty\mathtt{x}$
nxx	nfx	ngx	nex	nnx	$\mathtt{nn}_1\mathtt{x}$	$\mathtt{nn}_\infty\mathtt{x}$

Table 6: Weighting combinations used for the queriy vectors. The weighting surrounded by parentheses has no effect and is not used.

B Data sets

E.A. Fox at the Virginia Polytechnic Institute and State University has assembled nine small test collections in a CD-ROM. These test collections have been used heavily throughout the years for evaluation of information retrieval systems and they provide a good setting for preliminary testing. Among these nine sets we used four for our evaluation.

Adi Adi is a very small test collection of document abstracts from library science and related areas.

Cisi The data set consist of document abstracts in library science and related areas extracted from Social Science Citation Index by the Institute for Scientific Information.

Cran The Cranfield collection is a small collection with a large number of queries. The data set consist of document abstracts in aerodynamics originally used for tests at the Cranfield Institute of Technology in Bedford, England.

Med The Medline set is a small collection with a small number of queries. It has been extensively used in the past. The documents are abstracts in biomedicine received from the National Library of Medicine.

For a further summary on test sets see [1]. See also Fox [10].

Documents and queries are represented as vectors. Before the representation can be constructed a list of index terms must be compiled for each set. A list of all words (non-zero length strings of characters(A-Z,a-z) delimited by white space) found in the documents was constructed. Each word occurring on the SMART [19] stop list was removed. The remaining words form the set of index terms.

Table 7 summarizes some characteristics of the data sets and queries. All of these sets are rather small in size. For all the sets a large portion of the documents are relevant to some query. For all but the Medline set there are documents that are relevant to more than one query. All sets have more terms than documents and in general there are more terms per document than documents per term. All document vectors are longer than the query vectors.

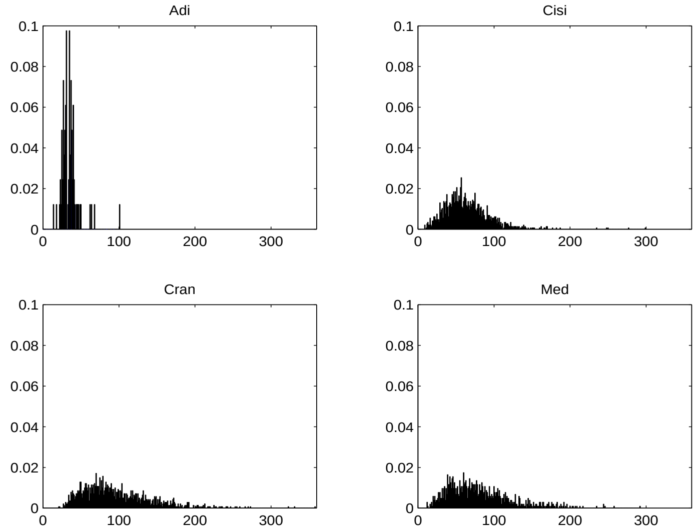

Figure 7: Portion of documents (y-axis) versus length of documents (x-axis) for the data sets.

In Cisi, Cran and Med the length of the documents (length is measured by number of terms) are more spread out than for the Adi set (see figure 7). This is probably due to the small size of the set. A few zero length documents appear in the Cranfield set.

	Adi	Cisi	Cran	Med
no of docs	82	1460	1400	1033
no of indexing terms	1311	10325	7776	12194
no of queries	35	35	225	30
no of relevant documents	72	467	924	696
no of <query,relevant doc> pairs	170	1742	1838	696
max/min/avr no of terms in docs	101/14/35	299/9/65	358/0/95	292/12/80
max/min/avr no of docs per term	44/1/2	644/1/7	703/0/11	262/1/5
max/min/avr no of terms in queries	13/3/7	18/3/8	21/4/9	23/2/11
nonzero elements in matrix (%)	2.1	0.48	0.79	0.46

Table 7: Some characteristics of the data sets Adi, Cisi, Cran and Med.

References

[1] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, Addison Wesley, 1999.

[2] Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide*, SIAM, Philadelphia, 2000.

[3] M. W. Berry and M. Browne, *Understanding Search Engines, Mathematical modeling and text retrieval*, SIAM, 1999.

[4] M. W. Berry, S. Dumais, and G. W. O'Brien, *Using linear algebra for intelligent information retrieval*, SIAM Review, 37 (1995), pp. 573–595.

[5] K. Blom and A. Ruhe, *Information Retrieval using a Krylov Subspace method*, submitted for publication, (2003).

[6] C. Buckley, A. Singhal, M. Mitra, and G. Salton, *New retrieval approaches using SMART: TREC 4*, in Proceedings of the Fourth Text Retrieval Conference (TREC-4), D. Harman, ed., Department of Commerce, National Institute of Standard and Technology. NIST special Publication, 1996, pp. 500–236.

[7] W. B. Croft, *Experiments with representation in a document retrieval system*, Information Technology: Research and Development, 2(1) (1983), pp. 1–21.

[8] S. T. Dumais, *Improving the retrieval of information from external sources*, Behavior Research Methods, Instruments, & Computers, 23 (1991), pp. 229–236.

[9] S. T. Dumais, G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, *Indexing by latent semantic analysis*, Journal of the American Society for Information Science, 41 (1990), pp. 391–407.

[10] E. A. Fox, *Characterization of two new experimental collections in computer and information science containing textual and bibliographical concepts*, Tech. Rep. 83-561, `http://www.ncstrl.org`, 1983.

[11] W. B. Frakes and R. Baeza-Yates, *Information Retrieval, Data Structures and Algorithms*, Prentice Hall, 1992.

[12] G. Golub and C. F. van Loan, *Matrix Computations 3rd edition*, Johns Hopkins, 1996.

[13] D. Harman, *Ranking algorithms*, in Information Retrieval, data structures and algorithms, W. B. Frakes and R. Baeza-Yates, eds., Prentice Hall, 1992, pp. 363–392.

[14] ——, *Appendix*, in The Eighth Text REtrieval Conference (TREC-8), D. Harman, ed., Department of Commerce, National Institute of Standard and Technology. NIST special Publication, 2000, p. A1.

[15] T. G. Kolda, *Limited-memory matrix methods with applications*, PhD thesis, Applied Mathematics University of Maryland, 1997.

[16] T. G. Kolda and D. P. O'Leary, *A semi-discrete matrix decomposition for latent semantic indexing in information retrieval*, ACM Transactions on Information Systems, 16 (1998), pp. 322–348.

[17] G. Salton, *Automatic Text Processing, The Transformation, Analysis and Retrieval of Information by Computer*, Addison-Wesley publishing company, 1989.

[18] G. Salton and C. Buckley, *Term-weighting approaches in automatic text retrieval*, Information Processing & Management, 24 (1988), pp. 513–523.

[19] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*, McGraw-Hill, 1983.

[20] A. Singhal, G. Salton, M. Mitra, and C. Buckley, *Document length normalization*, Tech. Rep. TR95-1529, Department of Computer Science, Cornell University, Ithaca, NY, 1995.