CHALMERS | GÖTEBORGS UNIVERSITET

PREPRINT

A Krylov Subspace Method Meets TREC

KATARINA BLOM

Department of Mathematical Statistics
CHALMERS UNIVERSITY OF TECHNOLOGY
GÖTEBORG UNIVERSITY
Göteborg Sweden 2004

Preprint 2004:20

A Krylov Subspace Method Meets TREC

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 2004:20
ISSN 0347-2809

Matematiska Vetenskaper
Göteborg 2004

a Krylov Subspace method meets TREC

Katarina Blom

April 12, 2004

Abstract

We expect a lot from our search engines. We ask them vague questions about topics that we are unfamiliar with ourselves and we anticipate an organized response. In this report we will follow one topic from the TREC collection. The topic is interesting in this way. The user asks for documents relevant to three terms with high search value, and expects the search engine to give back documents from two groups of relevant documents, some documents where these terms appear and some where they do not appear. We show how the Krylov method used for IR is able to indicate a (weak) connection between the groups of relevant documents. We also show how simple modifications of the method can be used to decrease the scoring for irrelevant documents.

keywords TREC, Krylov subspace, Information Retrieval

1 Introduction and Contence

An information retrieval (IR) system matches user queries (formal statements of information needs) to documents stored in a database.

We look at the document collection as a huge term document matrix A, where there is one row for each term that occurs anywhere in the collection and each column represents one document. The value stored in each matrix element defines a nonzero weight if a term occurs in a document. If a term is not present in the document the corresponding value is zero. The queries will be expressed by the same terms as the documents, i.e. as a column vector q with a nonzero value for each term appearing in the query. There are of course several ways to set up the term document matrix (choice of stop words[1], choice of weights for the nonzero elements in the matrix etc.).

Using a term document matrix A, query matching can be viewed as a search in the column space of A, and one of the most common similarity measures for scoring the documents is to measure the angles between the query vector and each document vector in A. In section 2.1 we discuss how we choose to set up the term document matrix for the TREC FT ([6]) set used in this report. We also discuss some properties of this matrix.

The *Krylov method* we use for IR is a subspace method based on Krylov sequences of subspaces reachable from the query vector [3]. The Krylov method is briefly presented in section 3 and some more details about the method can be found in appendix A.

In section 4 we follow one query (topic) from the TREC collection. The topic asks for documents relevant to polygamy, polygyny and polyandry, three terms with high search value. The documents that are relevant to the topic fall into two groups, those where any of the terms appear, and those where none of the terms appear. Clearly scoring the documents only by measuring the angles between the document vectors and the query vector will not capture all relevant documents.

Moreover the document vectors from one group are almost orthogonal to document vectors from the other group, but there is a weak connection between three documents in one of the two groups and one document in the other group. In experiments we show that the Krylov method is able to spot this relation. With the help of relevance feedback we are able to retrieve the relevant documents from a group where all vectors were orthogonal to the query vector.

The topic we follow not only describes what documents are relevant to the topic, it also describes what documents are irrelevant. In section 4.1.2 we show how a modified Krylov method [2] can be used in order to decrease the scoring for such (irrelevant) documents.

1.1 Notation

The notation used in this report is rather standard in numerical linear algebra We use uppercase letters for matrices and lowercase letters for vectors. Low-

[1] A stop word is a term whose frequency and/or semantic use makes it of no value as a searchable word.

ercase 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.

1.2 Measures

The retrieval efficiency of an IR 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 are ranked and we recieve an ordered list 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 ℓ. The *average precision* (non interpolated) for a single query is defined as

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

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

For further details, see Harman [6].

2 Data sets

The Text Retrieval Conferences (TREC) were created by the Defence Advance Research Projects Agency (DARPA) and the National Institute of Standards and Technology (NIST). The goal was to overcome the problems of not having a common base for experimentation and also to provide test sets of a reasonable large size. TREC provides large, diverse test data sets available to anyone interested in using it as a basis for their testing. Since 1992 they also provide a yearly conference to share results between different researchers.

The TREC 4 disc, which we have been using, contains three data collections, the *Financial Times*, 1991–1994 (FT), the *Federal Register*, 1994 (FR94) and the *Congressional Record*, 1993 (CR). The FT collection, FR94 collection and the CR collection consist of 210,158, 55,630 and 27,922 documents respectively. For our experiments we have used the FT collection with the 150 (ad hoc) queries no 301–450 making our experiments comparable with TREC conferences 7&8 [6].

TREC relevance judgments are made through a process known as pooling. The top 100 documents from runs submitted to TREC each year are combined into a single “pool”. The group who created the topic then judges each document in the pool for relevance.

2.1 Term document matrix

The elements of a *term document matrix* 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. Global, local weightings and normalization factors are applied to increase/decrease the importance of terms within and among documents. Often $a_{ij} = g_i l_{ij} d_j$ where l_{ij} is the local weighting for term i in document j, g_i is the global weighting for term i and d_j is the document normalization factor for document j.

Since every term does not normally appear in each document, the term document matrix is (very) sparse. A few terms, however, appear in all (or almost all) documents. These words have no discrimination value during a search and are called stop words. A *stop list* consists of terms whose frequency and/or semantic use make them of no value as searchable words. Eliminating the words appearing on the stop list usually decreases the total amount of words used in the database dramatically.

Preparing the matrix Minimal preprocessing on the raw text of the FT documents were done. All control sequences were removed (i.e. any text within < > delimiters). Upper case characters were replaced by lower case and white space were used to delimit terms. All non-zero length character sequences from (a-zA-z) were used as terms[2]. Defined in this way we found 230, 173 unique terms in the document collection.

A stop list consisting of all terms that appear in more than 10% of all documents from the FT set will have 299 terms and seems to be a good choice. The stop list will in general consist of common terms with no search value such as the and and. Removing the stop words will not decrease the size of the matrix much, but the number of nonzero elements will decrease significantly by 37%[3]. Clearly eliminating 37% of the nonzero elements will improve storage efficiency.

The most expensive operations in the Krylov subspace methods that we use for IR[4] are the matrix vector multiplications involving the term document matrix performed in the BIDIAG procedure. Since the time required for a matrix vector multiplication between a sparse matrix and a vector heavily depends on the number of nonzero elements in the matrix, we will also gain execution efficiency by eliminating the stop words from the term document matrix.

There is little debate on eliminating common words, but there is some discussion on what to do about singletons (words that only appear once or very infrequently in a document or a collection). For the FT collection almost 42%[5] of all terms in the database occur in only one document, thus eliminating these will decrease the size of the term document matrix significantly. For the BIDIAG procedure eliminating the singletons will decrease the length of the basis vectors, thus reducing both storage and time complexity for the procedure.

In this group of terms a lot of misspellings are found, but also a large amount of foregin words and rare names. The misspellings of course have no value as searchable words – but foregin words and names on the other hand might have high search values[6] even if it is unlikely that such a query is ever made.

[2] Since we chose to remove digits as well as special characters queries addressing for example telephone numbers, years, dates, decades etc will have no meaning.

[3] from 40, 687, 915 to 25, 535, 648.

[4] A short description of the methods used are given in section 3. For a more detailed description please see [3].

[5] 96, 549 out of 230, 173 terms

Let $A(t,p)$ be an $m \times n$ term document matrix where the rows of $A(t,p)$ correspond to all terms found in a data set and the columns corresponds to the documents. We will partition $A(t,p)$ such that

$$A(t,p) = \begin{bmatrix} & S(t) & \\ A_1 & & A_2 \\ 0 & & D(p) \end{bmatrix} \tag{1}$$

where the rows of $S(t)$ correspond to the t terms in the stop list. The rows in $D(p)$ correspond to the terms in $A(t,p)$ that appear in at most p documents (and the columns are the document vectors corresponding to the documents where these terms appear). The rows of A_1 and A_2 correspond to the terms in $A(t,p)$ that appear in more than p documents but are not on the stop list. The columns of A_1 correspond to the document vectors from $A(t,p)$ that are not in $D(p)$ and the columns of A_2 correspond to those that are. The 0 is the zero matrix.

In figure 1 the partitions (1) of $A(299,1)$ for the FT set are plotted. The matrix $D(1)$ is large in size (left figure) but the portion of nonzero elements is low (right figure). The $S(299)$ matrix on the other hand is small in size but a large amount of elements are found here.

In the term document matrix used for experiments in this report all 299 stop words are removed and all singletons (in $D(1)$ (1)) are kept. We let the entries l_{ij} in the term document matrix be 1 if term i is present in document j, 0 otherwise. In order to deemphasize common words and long documents first the rows and then the columns of the term document matrix are normalized using the Euclidean norm[7]. The row normalization g_i is a global weighting. Elements in the term document matrix corresponding to rare terms in the data set are given higher values than elements corresponding to common terms. The column normalization d_j will give high weights to rare terms in a particular document. Terms that are common in a particular document will get low weight.

Our way of constructing the term document matrix has several draw

[6] A term has high search value if it is rare among the documents and if a query asking for documents with this term appearing is likely to address this term.

[7] The column normalization will destroy the previous row normalization but not completely. Some deemphazising effect of common terms still remains.

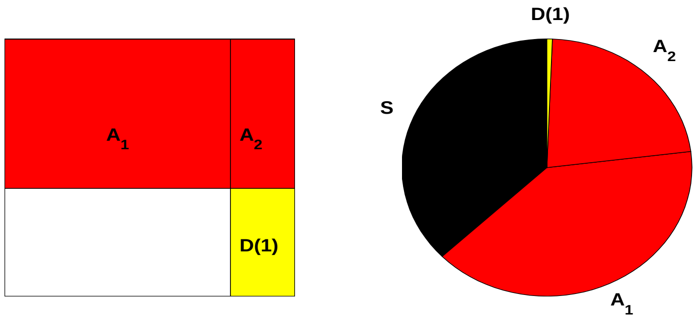

Figure 1: Partitioning of A(299,1) for the FT set. **Left figure** shows sizes of the zero matrix (white), D(1) (light grey), $[A_1A_2]$ (dark grey) and on top $S(299)$ (black). The black field is small compared to the other fields so it is not visible. **Right figure** shows how the nonzero elements are distributed between $S(299)$ (black), $[A_1A_2]$ (dark grey) and $D(1)$ (light grey).

backs. We do not use language specific techniques such as stemming[8], phrases, syntactic or semantic parsing, spell checking or correction, proper noun identification, a controlled vocabulary, a thesaurus[9] or any manual indexing. Our term weighting algorithm[10] is simply a row- followed by one column normalization of the 0/1 term document matrix.

We are aware of that retrieval efficiency for the methods used in this report might increase if the term document matrix is constructed with greater care. For a further discussion on weightings used with the Krylov method please see [1].

[8] In stemming the terms are represented by their stems. For example the stem comput could associate computable, computability, computation, computational, computed, computing, computer etc.

[9] A thesaurus is typically a one level or two level expansion of a term to other terms that are similar in meaning. For example the word computer may be linked to computer hardware and computer software.

[10] A nice summary of different term weighting methods can be found in the book by Frakes and Baeza-Yates [4]

2.2 The Queries

The queries for the TREC collections are called topics and have three parts. The *title*, a *description* that summarizes the topic and a *narrative* that further describes the query.

The titles and the description fields for the topics are often short. For example topic no. 316 used for this report has the title

> Polygamy Polyandry Polygyny

and the description field

> A look at the roots and prevalence of polygamy in the world today.

Titles and descriptions typically give information about which documents are relevant. We refer to such information as *positive information*. A query, consisting of terms from these fields, has a chance of bringing in terms that also would be found in relevant documents.

The narrative part of the queries often specifies what documents will be considered relevant and what documents are irrelevant. For example the narrative field from the same topic

> Polygamy is a form of marriage which permits a person to have more than one husband or wife. Polyandry refers to one woman sharing two or more husbands at the same time. Polygyny refers to one man sharing two or more wives at the same time. Primary focus of the search will be the prevalence of these practices in the world today and societal attitudes towards these practices. Also relevant would be discussions of the roots and practical sources of these customs. A modern development in this area is serial polygamy, a phrase coined to label the practice of men who take a series of wives in sequence as a solution to practical welfare, considerations of child care, housing, etc. Documents discussing serial polygamy will not be considered relevant.

Of course constructing a query vector by taking into account the terms in all three fields would bring in terms such as documents and relevant which are not relevant search terms for this query, it will also bring in terms matching irrelevant documents such as serial polygamy. Information on what documents are irrelevant we will refer to as *negative information*.

For this report we have constructed query vectors in two ways. The first approach is to assume that all information is positive and form *positive query vectors*. The positive query vectors were constructed using all the terms from the title only, using all terms from the title and the description, and using all terms from both the title, description and the narrative respectively. We will refer to these three query vectors as *short*, *middle* and *long* query vectors respectively. The query vectors are constructed as 0/1 vectors, the same way as the documents in the term document matrix, i.e. by letting a 1 in position j denote the presence of term j and 0 in position j denote absence of term j. The query vectors are normalized using euclidean norm (no global weighting is used). We let the terms represented in the term document matrix determine the terms in the collection.

The second approach is to construct *negative query vectors* that take into account the negative information i.e. which documents to leave out. The terms for the query vector with the negative information are picked by hand using information from the narrative field. Again a 1 in position i indicates presence of term i and a 0 absence. In the BIDIAG procedure we will avoid irrelevant documents by orthogonalizing against negative query vectors.

2.3 FT Term document matrix and query vectors

In table 1 the maximum, minimum and mean number of terms in document vectors and query vectors are listed. Although the topics are in general much shorter than the documents there exist topics that are longer than a few documents. However a large majority of the documents are longer than the topics. Common terms tend to appear more frequently in topics than in documents. When removing all terms appearing in more than 10% of the documents[11] the mean number of terms for the documents shrinks by roughly 37%[12]. The mean number of terms in the query vectors will shrink by 56% / 50% / 50% [13] for long/middle/short query vectors.

Rare terms are more common in the documents than in the topics. 22%[14] of the documents have at least one term that appears in only one document. At least one of these terms appears in 7% / 3% / 3% [15] of the

[11] 299 terms appears in more than 10% of the documents

[12] From $40,687,916$ to $25,535,649$.

[13] From 44/16/4 to 25/8/2 terms.

[14] $46,560$ out of $210,158$ documents.

[15] 10/5/4 (out of 150/150/150).

	max/min/mean number of terms	
	10% most common terms removed	no stop list
document vector	3227/3/122	3266/8/194
long query vector	61/8/25	98/18/44
middle query vector	27/1/8	48/6/16
short query vector	19/1/2	34/2/4

Table 1: Maximum, minimum and mean number of terms in document vectors and long, middle and short query vectors.

short/middle/long queries.

Due to the row and column normalizations of the term document matrix these terms are weighted high, and thus the presence of such term in the query vector will rank the corresponding document high. This is exactly what we want if the document is relevant to the topic and exactly what we would like to avoid if the document is irrelevant.

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,

$$c_j = \frac{q^T a_j}{||q||_2 ||a_j||_2}, \quad j = 1, \dots, n. \tag{2}$$

For the Krylov subspace methods we will use the Golub Kahan bidiagonalization procedure [5] 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} satisfying $B_{r+1} = Q_{r+1}^T A P_r$:

$$[Q_{r+1}, B_{r+1}, P_r] = \textsc{Bidiag}(A, q, r) \tag{3}$$

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. The *reached subspace* W forms an orthonormal basis for the column vectors in AP_r. The BIDIAG procedure is further described in appendix A.

Sometimes we want to avoid irrelevant information by making all document vectors in Q_{r+1} orthogonal to some vector $q^{(-)}$. Technically it is simple to rewrite the BIDIAG procedure to incrementally compute vectors q_i in Q_{r+1} orthogonal to $q^{(-)}$. The procedure

$$[Q'_{r+1}, B'_{r+1}, P'_r] = \textsc{Bidiag}_o(A, q, q^{(-)}, r) \tag{4}$$

will compute two basis matrices Q'_{r+1} and P'_r and an $r+1 \times r$ lower bidiagonal matrix B_{r+1}. Both the procedures BIDIAG and BIDIAG$_o$ are further described in appendix A.

3.1 The expanded query measure for document ranking

The reached subspace W, the basis matrices Q_{r+1}, P_r and the B_{r+1} matrix from the BIDIAG are used to score the documents in relevance order to the query (see Blom Ruhe [3]). In this report the *expanded query measure* is used for ranking the documents. An expanded query is

$$\hat{q} = WW^Tq \tag{5}$$

the projection onto the reached subspace. In the *expanded query measure* the documents are sorted measuring the cosine of the angle between the projected query $\hat{q}$ and each document vector in A,

$$c_j^{(2)} = \frac{\hat{q}^T a_j}{||\hat{q}||_2 ||a_j||_2}, \; j = 1, \ldots, n. \tag{6}$$

With the expanded query measure we are able to find document vectors that are orthogonal to the query vector.

Let the $m \times n$ term document matrix $A = \begin{bmatrix} M & X \end{bmatrix}$ where the columns of $M = \begin{bmatrix} m_j \end{bmatrix}$ correspond to d relevant documents and the columns of $X = \begin{bmatrix} x_j \end{bmatrix}$ correspond to the rest of the document vectors in A. Assume q is the query vector and that q is orthogonal to all document vectors in M, thus $q^TM = 0$. (This is precisely the situation we have with the topic we follow in section 4).

Let $\hat{q} = WW^Tq$ be the projected query vector (5). With $r = 1$ in the BIDIAG procedure (appendix A) the reached subspace

$$W = \text{span}(AP_1) = \text{span}\{AA^Tq\} = \frac{XX^Tq}{||X^Tq||_2},$$

where the last equality follows from the orthogonality between q and the columns in M. Let $c = X^Tq$ and $\gamma = \frac{1}{||c||_2}$, the expanded query measure (6) then becomes

$$c_j^{(2)} = \frac{\hat{q}^T a_j}{||\hat{q}||_2 ||a_j||_2} = \begin{cases} \gamma \dfrac{c^T X^T m_j}{||m_j||_2} & \text{if } 0 < j \leq d \\[2ex] \gamma \dfrac{c^T X^T x_{j-d}}{||x_{j-d}||_2} & \text{if } d < j \leq n \end{cases}$$

In order to score the relevant document vectors in M high we want the first d elements in $c^{(2)}$ to be large (and the last $n-d$ elements in $c^{(2)}$ to be small). This is true if there are document vectors in X that are close both to the query vector and to the relevant document vectors in M. So at least in theory document vectors from M could be scored high.

If the relevant document vectors in M are orthogonal to both the query vector and the rest of the document vectors in X (i.e. $M^Tq = 0$ and $M^TX = 0$) the two basis matrices Q_{r+1} and P_r (3) from the BIDIAG procedure will span bases for the two Krylov subspaces

$$\mathcal{K}_{r+1}(AA^T, q) = \{q, XX^Tq, \ldots, (XX^T)^r q\}$$

and

$$\mathcal{K}_r(A^TA, A^Tq) = \{\begin{bmatrix} 0 \\ X^Tq \end{bmatrix}, \ldots, \begin{bmatrix} 0 \\ (X^TX)^{r-1}X^Tq \end{bmatrix}\}.$$

respectively. No directions from the misses documents in M will be in these subspaces, thus we will not find the misses using the bidiagonalization procedure in this case.

4 Experiments

Topic number 316 deals with polygamy, polyandry and polygyny and was presented in section 2.2. According to the relevance judgements 19 docu-

	q	doc_1*	doc_2	doc_3	doc_4*	doc_5*	doc_6*	doc_7*	doc_8
polygamy	0.58	0	0	0	0.22	0.22	0.20	0.17	0.16
polyandry	0.58	0.77	0	0	0	0	0	0	0
polygyny	0.58	0	0.62	0.46	0	0	0	0	0

Table 2: Weights for the terms in the short query vector and for the 8 top scored documents using the vector model (2). The documents marked with a star are relevant to the topic doc_1, doc_4, doc_5, doc_6 and doc_7 correspond to document FT932–4228, FT934–6885, FT943–10242, FT943–5141 and FT942–4193 respectively.

ments from the FT set are relevant to this topic[16]. Two of the relevant documents are identical[17].

The three terms polygamy, polyandry and polygyny appear in the short query vector. These are rare terms of high search value. polygamy appears in 14 documents and 11 of these are relevant. polyandry is in 1 (relevant) document and polygyny is in 2 (irrelevant) documents. (Clearly removing singletons will soon make the short query vector empty and terms with high search value will be removed from the document vectors.).

If using the vector model (2) the short query vector addresses 17 document vectors (corresponding to columns of A_2 (1)) and is orthogonal to the rest of the documents in the set. 12 of the documents addressed by the query vector are relevant to the topic and average precision for the vector model becomes 0.45.

The weights for the terms polygamy, polyandry and polygyny, appearing in the short query vector, for the 8 top scored documents are listed in table 2. The effect of the row normalization of the term document matrix is clearly seen. The highest weight (0.77) corresponds to the most rare of the three terms (polyandry). The weights for the second most rare term (polygyny) are 0.62 and 0.46 respectively. For the least rare term (polygamy) weights are smaller. The effect of the column normalization is not that clear. The weight for the term polygyny in doc_3 is lower because more rare terms appear in doc_3 than in doc_2. doc_3 is shorter than doc_2.

[16] Documents FT922–11381, FT922–12843, FT931–5366, FT931–6791, FT932–1167, FT932–3422, FT932–3656, FT932–4228, FT932–5625, FT933–7689, FT934–6885, FT942–4193, FT943–10242, FT943–2362, FT943–5141, FT944–1831, FT944–2037, FT944–2863 and FT944–8467 are relevant to topic 316.

[17] FT944–1831 and FT944–2037 are identical.

Since only one of the three terms in the short query vector appear in each document, the scoring follows the weights of the terms directly.

The relevant documents can be divided into two distinct groups, *retrieved relevant* and *misses*[18]. In the former group (retrieved relevant) one of the terms polygamy or polyandry appear. All the documents in the later group (misses) are orthogonal to the short query.

The 12 retrieved relevant documents are scored

$$[1, 4, 5, 6, 7, 9, 10, 11, 12, 14, 15, 16]. \tag{7}$$

and the 7 misses were scored[19]

$$[\infty, \infty, \infty, \infty, \infty, \infty, \infty].$$

The challenge is to see whether we can locate the 7 misses[20] using the Krylov subspace method (3).

4.1 Using the Krylov subspace method

Using the short query vector The short query vector is orthogonal to the misses, and using it as a starting vector for the BIDIAG procedure (3) will start a search orthogonal to the documents we want to locate. Unless the document vectors from the two groups of relevant documents are close in angles, iterating in the BIDIAG procedure will not bring in relevant documents from the misses group.

In table 3 terms and weights for the 10 top weighted terms in the projected query vector (5) are listed[21]. The weights for terms in the 8 top scored documents are also shown in the table.

[18] The retrieved relevant documents are the documents that are scored well enough for a human user to judge and the misses are the relevant documents scored so low that they will not be shown to the user (i.e. one cannot expect a user to read through all previous scored documents). In [] two kinds of retrieval failures were used, *false alarms* and misses. False alarms are highly scored irrelevant documents.

[19] We mark the orthogonality by using the scoring ∞.

[20] All relevant documents carry one of the terms polygamy, polygamist, polygamists and polyandry respectively. A search vector addressing all these four terms will score all relevant documents below 23. Thus if stemming had been used for the term document matrix or if the query vector had been expanded with polygamist and polygamists all the relevant documents had been captured using the vector model.

[21] We used the short query vector and constructed projected query vectors $\hat{q}$ (5) for $r = 1, \ldots, 6$ in BIDIAG. The top 10 weighted terms in the projected query vector that were brought in by the procedure dominated all 6 query vectors computed.

The terms polyandry, polygyny and polygamy still show high weights. Naturally all three terms have lower weights in $\hat{q}$ than in q. Also note that the most rare of the three terms (polyandry) has higher weight than polygyny, the second most rare term. The most common of the three terms polygamy has least weight.

The new terms that were brought in by the bidiagonalization scheme (gamy, supergrass, annemarie, telltale, bigamists, spinsters and matings) do not lead us to the wanted documents. doc_1 (relevant) lists different TV-shows and the words supergrass, annemarie and telltale comes from listed TV-shows that have nothing to do with polyandry. None of these terms appear in any other relevant documents. The words gamy, spinsters and matings appear in the non relevant documents doc_2 and doc_3. These terms do not appear in any relevant document.

Relevant documents from the two groups retrieved relevant and misses were scored (sorted as in 7)

$$[2, 7, 4, 5, 6, 9, 10, 21, 27, 31, 20, 33]$$

and

$$[579, 2981, 2965, 5113, 15071, 24559, 24560] \qquad (8)$$

respectively. Even though average precision decreased from 0.45 to 0.33 all documents in the first group are scored below 33. The misses are not orthogonal to the projected query vector – so it is possible to score them. However they still do not belong to the group of retrieved relevant documents. The two document from the misses group that were scored 24559 and 24560 are identical.

Document vectors from the two groups retrieved relevant and misses are almost orthogonal. Let the first 12 columns of A_r be the retrieved relevant document vectors from the term document matrix and let the last columns be the document vectors corresponding to the 7 misses. The orthogonality is clearly seen in figure 2 where all scalar products[22] in $A_r^T A_r$ greater than 0.01 are plotted. Three documents from the retrieved relevant group share some terms with one of the misses. Although the three scalar products are small (between 0.01 and 0.05) the connection between the two groups can be seen in the scoring (8) where this document is scored 579.

[22] Since all document vectors are normalized $A_r^T A_r$ is the cosines of the angles between the relevant document vectors.

	$\hat{q}$	doc_2	doc_1*	doc_3	doc_5*	doc_6*	doc_7*	doc_4*	doc_8
polygyny	0.44	0.62	0	0.46	0	0	0	0	0
polyandry	0.44	0	0.77	0	0	0	0	0	0
monogamy	0.22	0.36	0	0	0	0	0.25	0	0.24
polygamy	0.20	0	0	0	0.22	0.20	0.17	0.22	0.16
supergrass	0.18	0	0.31	0	0	0	0	0	0
annemarie	0.17	0	0.29	0	0	0	0	0	0
telltale	0.16	0	0.27	0	0	0	0	0	0
skews	0.15	0.33	0	0	0	0	0	0	0
bigamists	0.13	0	0	0	0	0	0	0.82	0
matings	0.13	0	0	0.38	0	0	0	0	0

Table 3: The 10 top weighted terms in the projected query vector $\hat{q}$ (5) and their corresponding weights in $\hat{q}$ and in the 8 top scored documents. The documents were scored using the cosine of the angle between the projected query $\hat{q}$ and each document vector in the term document matrix (6), with the short query vector and $r = 3$ in BIDIAG (3). The documents marked with a star are relevant to the topic. The document numbers refer to the numbers in table 2

Also note (see figure 2) that the group of retrieved relevant documents can be divided into two groups having the first document orthogonal to the rest of the documents in the group. In this document the term polyandry appears. Since polyandry appears only once in the whole set the corresponding element is weighted high, and since the term also appears in the query vector the document is scored high (both in the vector model and using the Krylov method). Removing all rows corresponding to terms only appearing once in the set from the term document matrix will make this document vector orthogonal to both the query vector and the rest of the relevant documents. Thus the document will not be captured by the vector model nor by the Krylov subspace method.

False alarms are highly scored irrelevant documents[23]. In some of the false alarms some interesting relations appear.

- A modern phrase serial polygyny labels the practice of women who take

[23] The false alarms are scored 1, 3, 8, 11 – 19, 22 – 26, 28 – 30 and 32.

a series of husbands in sequence as a solution to practical welfare[24]. The two terms serial and polygyny appears in this context in the two irrelevant documents scored 1 and 3. Using the technique discussed in section 4.1.2 (by orthogonalizing against negative query vectors in the BIDIAG procedure) these two irrelevant documents may be moved further down in the ranking list. Let for example the elements in the negative query vector address the terms serial and polygyny.

- The irrelevant document scored 8 lists the today's television. One of the programs listed examines polygamy versus monogamy. The document were brought in because of polygamy appearing in it.
- In the two relevant documents scored 4 and 9 the term polygamy appears. The documents describes a Malaysian sect that uses polygamy. The false alarms scored 11, 12, 13 and 17 are short news telegrams about the Malaysian sect, clearly brought in because of its closeness to the relevant documents scored 4 and 9. In the false alarm scored 29 the sect is mentioned. (None of the terms in the short query vector appear in the irrelevant documents scored 11, 12, 13, 17 and 29.)
- The false alarms scored 14, 15 and 18 are about a Malaysian politician, brought in because of it closeness to the relevant documents scored 4 and 9 and the irrelevant documents scored 11, 12, 13 and 17 above.
- The false alarm scored 26 is about a family running an illegal bomb factory and among the TV shows listed in the (relevant) document scored 2 there is one program about this family.
- The terms skews and polygyny appear in the false alarm scored no. 1. The document were scored high because of the term polygyny appearing in it and it brings in the rare and high weighted term skews[25] to the projected query vector. skews also appears in the false alarms scored 19 and 30.

[24] In the narrative field of the topic it is stated that documents discussing the male equivalence serial polygamy will not be considered relevant.

[25] skews appears in 7 documents.

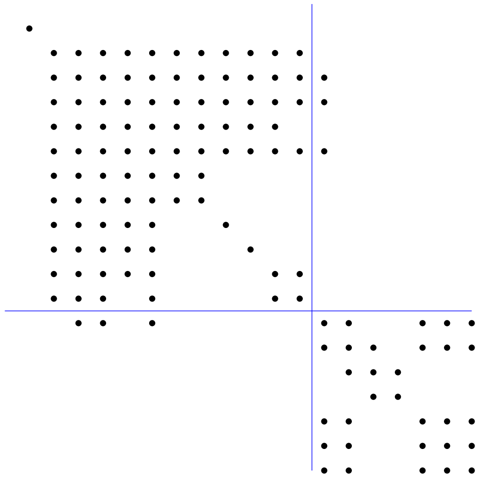

Figure 2: Nonzero elements greater than 0.01 in $A_r^T A_r$ where the 12 first columns of A_r are the retrieved relevant document vectors and the 7 last columns correspond to the misses. Columns in A_r are sorted as (7) and (8).

Using the middle and long query vectors None of the middle or long query vectors are orthogonal to all the misses (but the angles are large). The query vectors are still rather close to at least a few document vectors in the relevant retrieved group. Since the queries are longer the terms polygamy, polyandry and polygyny get lower weights, thus relevant documents from the relevant retrieved group above are scored worse compared to the scoring when the short query vector was used.

Relevant documents from the group retrieved relevant for the middle query vector using the vector model were scored (sorted as in 7)

$$[1, 14, 12, 16, 24, 28, 51, 90, 98, 113, 157, 205]$$

and for the long query vector

$$[3, 385, 169, 416, 430, 887, 728, 3143, 3199, 3989, 5139, 11607]$$

respectively. Iterating a few steps in the BIDIAG procedure will not improve the scoring for the relevant documents. In table 4 the top 10 weighted terms that are brought in after three iterations with the BIDIAG to the projected middle and long query vectors are listed. These differ from the terms that were brought in by the short query vector (listed in table 3). The important search term polygamy is no longer among the top 10 weighted terms. Due to the naive way of constructing the long query vector terms such as discussing and discussions are brought in.

Terms in the intersection In table 5 all terms that appear at least in 6 of the relevant documents and at least once in both groups, retrieved relevant and misses, of relevant documents. The terms appearing in the intersection are weighted low in the relevant documents and thus will not be useful as search terms in the query vector. Also note that the terms appearing in both groups differ completely from the terms appearing in the projected queries.

Summary All three query vectors, short, middle and long, are either orthogonal or almost orthogonal to all documents in the misses group and close in angles to the documents in the relevant retrieved group. The documents from the misses group are almost orthogonal to the documents in the relevant retrieved group. Using the BIDIAG procedure we are able to spot the weak relationship between the two groups but in order to capture documents from the misses group we need either the starting vector (the query) to address document vectors from both groups or the document vectors from the two groups need to be closer in angles. Clearly in order to find the 7 misses we need to choose the starting vector and use the BIDIAG procedure with greater care.

$\hat{q}$(middle)		$\hat{q}$(long)	
look	0.51	housing	0.14
prevalence	0.29	discussions	0.13
roots	0.25	customs	0.11
polygyny	0.11	sources	0.10
polyandry	0.11	practices	0.09
column	0.09	series	0.09
lex	0.07	practice	0.09
elixir	0.07	discussing	0.09
grass	0.06	considered	0.09
monogamy	0.06	welfare	0.09

Table 4: Terms and weights for the 10 top weighted projected query vectors for the middle and long query vectors.

party french old home called clear senior once role family man opposition society century men history children books

Table 5: Terms that appear in at least 6 of the relevant documents and at least once in both groups retrieved relevant and misses.

4.1.1 Relevance feedback

In a relevance feedback cycle, the user is presented a list of retrieved documents, and after examining them, marks those that are relevant. The main idea is to use the information provided by the user to make a new (hopefully) improved search.

Assume a user has given judgements on the documents in the retrieved relevant group (7) from the vector model. Using the term weights from the relevant document vectors we constructed a new (improved) starting vector for the BIDIAG. We let the new query vector be the vector sum of the relevant

	q
polygamy	0.58
bigamists	0.33
polyandry	0.31
taso	0.30
rakai	0.30
magesi	0.25
deviationist	0.24
neziha	0.23
mezhoud	0.23
ashaari	0.21

Table 6: Using the term weights from the relevant document vectors a new query vector was constructed. We let the new query vector be the sum of the row vectors in the document vectors corresponding to the relevant retrieved documents. Terms and weights for the 10 top weighted terms are shown.

retrieved documents.

The terms and the weights (after normalizing) for the top 10 weighted terms in the improved query vector are listed in table 6. These words are rare in the term document matrix and a few differ from the terms for the projected query vector listed in table 3. The five terms taso, magesi, neziha, mezhoud and ashaari comes from the three names Mrs Marble Magesi of Taso, Mrs Neziha Meshoud and Ashaari Muhammad appearing in some of the relevant documents. The names are rare and thus high weighted. Rakai is a southern district in Uganda.

The important search terms polygamy and polyandry appear in the query vector. polygyny appeared in the two irrelevant documents scored 2 and 3 in (7) and is not present. Due to the orthogonality between the two groups misses and retrieved relevant none of the 10 top weighted terms in the improved query vector appear in any of the misses. Also there is no match between the terms in the intersection between the two groups relevant retrieved and misses, listed in table 5, and the most common terms in the improved query vector.

The scorings for the vector model and the Krylov method are listed below

Retrived relevant documents (sorted as in 7)

$$\begin{array}{rl} \text{vector model} & [8, 2, 4, 7, 1, 3, 5, 16, 18, 20, 21, 22] \\ \text{Krylov method} & [16, 9, 4, 5, 7, 12, 10, 31, 33, 36, 24, 37] \end{array}$$

and misses (sorted as in 8)

$$\begin{array}{cl} \text{vector model} & [\infty, \infty, \infty, \infty, \infty, \infty, \infty] \\ \text{Krylov method} & [57, 43346, 116025, 90478, 28075, 46251, 46252] \end{array}$$

The improved query vector manages to capture the document vector in the missing group that is closest to the intersection between the two groups of relevant documents (see figure 2 and scoring (8)) and it is scored 57. If we consider this document to be retrieved we may repeat the process and construct yet an improved query vector by summing the row vectors in the relevant retrieved group and the retrieved document vector from the misses. This will bring in the term polygamist to the query vector.

A query vector with the three terms polygamy, polyandry and polygamist highly weighted will retrieve all but two relevant documents using the vector model. Since the two not retrieved documents are connected to some of the retrieved relevant documents iterating a few steps in the BIDIAG with this query vector will capture at least one of the remaining two relevant documents.

Summary Using the expanded query measure (6) to rank the documents for relevancy and with a carefully picked starting vector for the BIDIAG procedure we are able to retrieve relevant document vectors that are orthogonal to the query vector. In the relevance feedback cycle (section 4.1.1) we used judgements from the user to create a new starting vector. The new starting vector again was orthogonal to the misses, but closer in angles to some non relevant documents that in their turn were close to one of the misses. In this way we were able to steer the Krylov method to further resemble relevant documents.

4.1.2 Using negative information

Documents discussing serial polygamy are not relevant to this topic (see the narrative field for this topic in section 2.2). There is at least one document dealing with serial polygamy in the FT set. This (irrelevant) document is scored 17 using the vector model with the short query vector (7). Clearly removing the term polygamy will decrease retrieval performance since all of the retrived relevant documents were found merely through this term. By using the BIDIAG$_o$ procedure (4) we are able to orthogonalize against non wanted information. We let the negative query vector $q^{(-)}$ have the terms serial and polygamy and q be the short query vector. Average precision will be 0.33 and the document dealing with serial polygamy is scored 120. (The two groups of retrieved relevant documents and misses still remains).

5 Conclusions

There are usually many ways to express a given concept so the terms in a user's query may not match those of a relevant document. The query vector for topic no 316 used in section 4 is orthogonal to some of the relevant document vectors and thus splits the set of relevant documents into those that are retrieved (retrieved relevant) and those that are not retrieved (misses) when using the vector model scoring (2). However some of the misses are connected to some of the retrieved relevant documents through common terms. With the BIDIAG procedure we are able to indicate this (rather weak) connection between the two sets of document vectors. With some relevance feedback from the user some of the misses were ranked well enough to be retrieved.

Many terms have multiple meanings, so terms in a user's query will match terms in documents that are not of interest to the user. Reformulating the BIDIAG procedure slightly it is possible to orthogonalize against unwanted directions and thus avoid subspaces (or vectors) spanned by terms that is of no interest. The projected query vector (5) used in the expanded query measure (6) will be orthogonal to these subspaces (or vectors) and there by the irrelevant retrieved documents is likely to be ranked further down the list.

A 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 [5] 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.
```

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

Define

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

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

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

$\mathcal{K}_r(A^TA, A^Tq)$. The lower bidiagonal matrix $B_{r+1} = Q_{r+1}^T AP_r$ 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.

Technically it is easy to rewrite the BIDIAG procedure to incrementally compute vectors q_i in Q_{r+1} orthogonal to a vector c.

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

The two calls BIDIAG$_o$(A,q,c,r) and BIDIAG($(I - cc^T)A$,q,r) are equivalent. The procedure BIDIAG$_o$ is further discussed in [2].

References

[1] K. BLOM, *Experimenting with different weighting schemes for the Krylov Subspace method used for IR*, tech. rep., Dept. of Mathematics, Chalmers university of Technology, 2003.

[2] ——, *Modified Krylov subspace methods for information retrieval*, tech. rep., Dept. of Mathematics, Chalmers university of Technology, 2003.

[3] K. BLOM AND A. RUHE, *Information Retrieval using a Krylov Subspace method*, submitted for publication, (2003).

[4] W. B. FRAKES AND R. BAEZA-YATES, *Information Retrieval, Data Structures and Algorithms*, Prentice Hall, 1992.

[5] G. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins, 3 ed., 1996.

[6] D. HARMAN, *The Eighth Text REtrieval Conference (TREC-8)*, NIST Special Publication 500–246. http://trec.nist.gov/pubs/trec8/t8_proceedings, (2000), p. A1 (Appendix).