Why Spectral Retrieval Works

Holger Bast
Max-Planck-Institut für Informatik
Saarbrücken, Germany
bast@mpi-inf.mpg.de

Debapriyo Majumdar
Max-Planck-Institut für Informatik
Saarbrücken, Germany
deb@mpi-inf.mpg.de

ABSTRACT

We argue that the ability to identify pairs of related terms is at the heart of what makes spectral retrieval work in practice. Schemes such as latent semantic indexing (LSI) and its descendants have this ability in the sense that they can be viewed as computing a matrix of term-term relatedness scores which is then used to expand the given documents (not the queries). For almost all existing spectral retrieval schemes, this matrix of relatedness scores depends on a fixed low-dimensional subspace of the original term space. We instead vary the dimension and study for each term pair the resulting *curve* of relatedness scores. We find that it is actually the *shape* of this curve which is indicative for the term-pair relatedness, and not any of the individual relatedness scores on the curve. We derive two simple, parameterless algorithms that detect this shape and that consistently outperform previous methods on a number of test collections. Our curves also shed light on the effectiveness of three fundamental types of variations of the basic LSI scheme.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Clustering, Retrieval Models

General Terms

Algorithms, Performance, Theory, Experimentation

Keywords

Spectral Analysis, Latent Semantic Indexing, Document Expansion, Curve of Relatedness Scores

1. INTRODUCTION

Latent semantic indexing (LSI) [4] became famous as one of the first information retrieval techniques for fully automatically and with surprising effectiveness dealing with the problems of synonymy (web, internet) and polysemy (surfing the web, surfing at Waikiki beach), which make searching by pure text matching, where only documents are returned that contain all or at least some of the words in the query, so frequently a frustrating experience.

LSI and its many successors are based on what is usually referred to as the vector space model [18], where documents as well as queries are represented as vectors, with each dimension corresponding to a word or *term* that occurs in the document collection; the similarity between a document and a query is then measured by the similarity between the corresponding vectors, typical measures being the dot product or the cosine of the angle (which coincide if the vectors are normalized).

The key idea of LSI is to map both queries and documents from the m-dimensional space, where m is the number of terms, to a space of significantly lower dimension k, and compute vector similarities in the reduced space instead of in the original one. The hope is that while the dimensions of the original space correspond to the specific terms used, the dimensions of the lower dimensional space correspond more to the (relatively few) *concepts* underlying this term usage. We will therefore refer to the original m-dimensional space as *term space* and to the reduced k-dimensional space as *concept space*.

LSI derives the low-dimensional representation by a spectral decomposition of the term-document matrix, more precisely, by projecting the high-dimensional vectors on the k most significant left singular vectors, i.e., pertaining to the k largest singular values. Following LSI, many other related schemes have been proposed [12] [1] [2] [16] [6] [13] [7], which all do (or can be used for) what we call *spectral retrieval* in this paper: the documents are projected from the original term space to some lower-dimensional space spanned by eigenvectors related to the term-document matrix.[1] All of these schemes can be viewed as to augment the basic LSI method by mere normalizations and rescalings, which can have a significant impact on retrieval quality. This will be explained and discussed in Section 6.

There have been many attempts to explain the success of LSI and spectral retrieval in general, most notably the analyzes of [17], [5], [3], and [2]. The bottom line of all these results is that if the term-document matrix is a slight perturbation of some rank-k matrix — and the results differ in what kind of perturbation is permitted — then spectral retrieval will succeed in recovering that rank-k matrix. On the one hand, this is of course an interesting and non-trivial

[1]The various concept-based retrieval schemes based on probabilistic modelings, e.g. [11], are of a rather different kind, and their investigation lies outside the scope of this paper.

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

property. On the other hand, any statement of this kind is much closer to the mathematics of spectral retrieval (approximating a set of points from a high-dimensional space by a set of points in a lower-dimensional space) than to what real document collections actually look like. For example, *any* of the many term-document matrices we have seen so far — be they small, medium, or large — can be seen as "approximately rank k" for about any k with equal justification. Subtler points are that there is usually more than one reasonable way to divide a set of documents into (a given number of) categories, and that natural categorizations are often not flat but rather hierarchical.

In this paper, we take a novel approach to understanding and analyzing spectral retrieval, and to making use of its potential. We first work out the details of LSI, the most basic implementation of spectral retrieval, as a *document expansion* (and not query expansion) process. This leads us to the insight that spectral retrieval essentially works by assigning a relatedness score to each pair of terms. We then introduce our central notion: the *curve of relatedness scores*, which for a given term pair shows the mentioned relatedness score as the dimension k of the concept space is varied from 1 to the full rank of the term-document matrix. We then provide complementary theoretical and empirical evidence that the relatedness of two terms in a collection corresponds to a certain characteristic *shape* of the curve of relatedness scores and *not* to any of the individual scores on the curve.

Our findings imply a surprising answer to the question of the appropriate choice of dimension, namely that *every fixed choice is inappropriate for a significant fraction of the term pairs*. But our findings also help us out of this dilemma, by naturally leading to two simple, parameterless algorithms that assess the shape of the curve of relatedness scores for each term pair and thus overcome the limitations of methods committing to a concept space of fixed dimension. We find that on three test collections — one small, one medium-sized, and one large — our algorithms consistently outperform LSI and a variety of its successors.

Our findings also shed new light on the effectiveness of the many variations of spectral retrieval. In Section 6, we formulate three fundamental types of variations of the basic LSI scheme, which together capture almost all variants of spectral retrieval which we found in the literature. We comment on the effects of each of these variations from the point of view of our curves of relatedness scores. In particular, we will see that none of these variations can overcome the problems associated with considering only individual scores of the curves, instead of their overall shape.

Parts of our work were inspired by the setup from Dupret [6], who experimentally investigated the influence of the choice of k on particular terms (not term pairs). In a statistician's approach, Efron [8] provided a number of accurate formulas for predicting that k which yields the best retrieval performance (among the possible choices of a *fixed* k). In a very recent work, Dupret [7] presented a heuristic that avoids committing to a fixed k, with promising, yet inconclusive results and without much emphasis on theoretical foundation and explanation.

2. VIEWING LSI AS DOCUMENT EXPANSION

It has been remarked (but never explored further in much detail) by several authors that LSI is related to a process known as *query expansion*, where terms related to those initially present are added to a query. As we show next, LSI is in fact doing *exactly* what could be called *document expansion*. The most widely used variant of LSI maps vectors from the m-dimensional term space to the k-dimensional concept space by the map $x \mapsto U_k^T x$, where U_k is the matrix containing the k left singular vectors pertaining to the k largest singular values. The cosine-similarity of a query q with document A_i in the LSI space is then given by the formula

$$(U_k^T q)^T \cdot U_k^T A_i \,/\, (|U_k^T q| \cdot |U_k^T A_i|).$$

If our goal is to rank the documents by their similarities to a given query, we can drop the division by $|U_k^T q|$ (it is the same for every document), and obtain the $1 \times n$ row vector of all similarities by

$$(U_k^T q)^T \cdot U_k^T A N,$$

where N is an $n \times n$ diagonal matrix, with $N_{ii} = 1/|U_k^T A_i|$. Now a simple but important observation is that

$$(U_k^T q)^T U_k^T A_i = q^T U_k U_k^T A_i,$$

and

$$|U_k^T A_i|^2 = A_i^T U_k U_k^T A_i = A_i^T U_k U_k^T U_k U_k^T A_i = |U_k U_k^T A_i|^2.$$

Instead of mapping both queries and documents to the k-dimensional concept space via U_k^T and computing the cosine similarity there, we may therefore as well transform the documents via the $m \times m$ matrix $U_k U_k^T$, and compute cosine similarities in the original term space. According to the preceding calculations, both procedures will yield exactly the same ranking.

Now the effect of multiplying a document or any vector in the m-dimensional term space by an $m \times m$ matrix, as $U_k U_k^T$ is one, has a simple intuition. Let us call the matrix T and the vector d, and first assume that both have entries either zero or one. Then if $T_{ij} = 1$ and $d_j = 1$, the ith component of Td will also be (at least) 1, that is, the effect of $T_{ij} = 1$ is to "add" term i to the document whenever term j is present. More generally, for arbitrary values in T and d, the entry in T_{ij} determines to which extent the weight for term i should be increased when term j is present. Note that for LSI, T may also have negative values, in which case a term can also effect the "subtraction" of other terms.

3. THE CURVE OF RELATEDNESS SCORES AND ITS MATHEMATICS

As we said, almost all previous spectral retrieval schemes commit to a particular dimension, and from that viewpoint it appears natural to study the entries of the document expansion matrix $T = U_k U_k^T$ for some fixed value of k. The central idea of this paper is now to shift to a different viewpoint, and study how the entry at a fixed position (term pair) changes when the dimension is varied, that is, *instead of looking at all the entries for a fixed dimension, we now look at a fixed entry for all dimensions.*

DEFINITION 1. *Given an $m \times n$ term-document matrix of rank r, let $U\Sigma V^T$ be its singular value decomposition, where $U = (u_{ij})$ is an $m \times r$ matrix and the singular values in Σ are*

sorted in descending order. Then the curve of relatedness scores for terms i and j *is a plot of the function*

$$k \mapsto \sum_{l=1}^{k} u_{il} u_{jl} \quad \textit{for} \quad k = 1, \ldots, r.$$

By looking at the curves of many term pairs, one quickly recognizes three characteristic types of shapes, irrespective of the document collection under consideration.

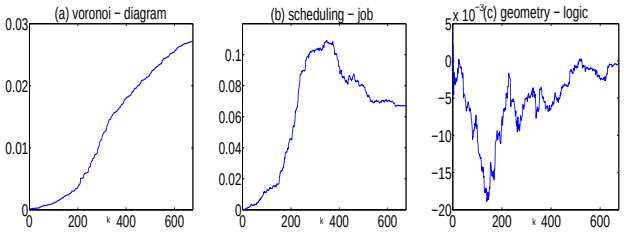

Figure 1: Curves of relatedness scores for pairs of related terms (a) + (b), and for a pair of unrelated terms (c), all from a collection of computer science abstracts.

Figure 1 shows a typical curve for each type. Curves of type (a) go up relatively steadily from beginning to end. Curves of type (b) go up steadily in the beginning, but then from some point on start to go down again. Curves of type (c) are quite different in that they do not have a clear direction upward or downward; they are also less smooth. There are a few intermediate cases, but most curves quite naturally fall into one of these three categories. It turns out that curves of type (a) and (b) typically belong to intuitively more or less related terms ("voronoi" and "diagram", "job" and "scheduling"), while most curves of type (c) belong to intuitively unrelated terms ("geometry" and "logic"). So far, these are empirical observations, but we will soon see theoretical evidence for what we have described here.

It is important to note that for our categorization above *we did not use the order of magnitude* of the relatedness scores. It is true that curves of types (a) and (b) *on average* reach larger scores than those of type (c). But it will become clear by the theory that follows that individual scores on the curves — and almost all previous schemes are based on these — are much less reliable indicators for term relatedness than the overall shape of the curves.

3.1 Perfectly related terms

We next explain why intuitively related terms give rise to curves of the types (a) and (b) that we have seen in Figure 1. To this end, we introduce a notion of *perfectly related terms*, which are terms that have identical co-occurrence patterns in the document collection; this definition was inspired by the experimental setup of Dupret [6].

DEFINITION 2. *Two terms (indexed i and j; without loss of generality assume $i = m-1$, $j = m$) in an $m \times n$ term-document matrix A are called* perfectly related *if, for some permutation of the columns of A,*

$$A = \begin{bmatrix} A_1 & A_1 & A_2 & A_3 \\ a_1 & b_1 & a_2 & 0 \\ b_1 & a_1 & a_2 & 0 \end{bmatrix}$$

where A_1 is a sub-matrix with dimension $(m-2) \times n_1$, A_2 is a sub-matrix with dimension $(m-2) \times n_2$, A_3 is a sub-matrix of dimension $(m-2) \times n_3$, a_1 and b_1 are row vectors of length n_1 each and a_2 is a row vector of length n_2 (consequently, $2n_1 + n_2 + n_3 = n$).

The following lemma says that a pair of perfectly related terms gives rise to a very particular substructure in the left singular vectors, which, as we will explain afterwards, implies an equally particular kind of curve of relatedness scores.

LEMMA 1. *For a matrix A as in Definition 2,*

(a) the vector $v = (0, \cdots, 0, -1/\sqrt{2}, 1/\sqrt{2})$ is a left singular vector of A;

(b) the corresponding singular value is $|a_1 - b_1|$;

(c) for all other left singular vectors u of A, $u_{m-1} = u_m$, that is, the last two entries are equal.

PROOF. If $A = U\Sigma V^T$ is the singular value decomposition of A, and we define $C = AA^T$, we have

$$C = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma(V^T V)\Sigma^T U^T = U\Sigma^2 U^T,$$

because $V^T V$ is the identity matrix. The left singular vectors of A are therefore just the eigenvectors of C, and the singular values of A are the square roots of the eigenvalues of C, so that we may as well consider C instead of A. Now if A is as stated in (2), then

$$C = \begin{bmatrix} C_1 & c_1^T & c_1^T \\ c_1 & x & y \\ c_1 & y & x \end{bmatrix}$$

where $C_1 = 2A_1A_1^T + A_2A_2^T + A_3A_3^T$ is an $(m-2) \times (m-2)$ matrix, $c_1 = a_1A_1^T + b_1A_1^T + a_2A_2^T$ is a $1 \times m$ vector, $x = a_1a_1^T + b_1b_1^T + a_2a_2^T$, and $y = a_1b_1^T + b_1a_1^T + a_2a_2^T$. Then $v = (0, \cdots, 0, -1/\sqrt{2}, 1/\sqrt{2})$ is an eigenvector of C with eigenvalue $x - y$ because $Cv = (x-y)v$, and

$$\begin{aligned} x - y &= a_1a_1^T + b_1b_1^T - a_1b_1^T - b_1a_1^T \\ &= (a_1 - b_1)(a_1 - b_1)^T \\ &= |a_1 - b_1|^2. \end{aligned}$$

Moreover, since all other eigenvectors u of C are orthogonal to v, the dot product $u^T v = -u_{m-1}/\sqrt{2} + u_m/\sqrt{2}$ has to be zero, hence $u_{m-1} = u_m$. □

Lemma 1 implies the following particular shape of the curve of relatedness scores for two perfectly related terms. If k is the rank of the special singular vector from Lemma 1(a), then because of Lemma 1(c) the relatedness scores will steadily increase until k, then at dimension k *fall off* by $-1/\sqrt{2} \cdot 1/\sqrt{2} = -1/2$, and then again increase steadily. For an example, see the left curve of Figure 2. By Lemma 1(b), the dimension k of fall-off is exactly[2] the number of singular values which are greater or equal to $|a_1 - b_1|$.

The dimension of fall-off depends on the co-occurrence pattern of the term pair in an interesting way. For an intuitive explanation, let us assume that $b_1 = 0$, which, according to Definition 2, means that when the terms co-occur, they do so in the same frequency. If then also $|a_1| = 0$, the two terms co-occur whenever they occur. Then $|a_1 - b_1|$ is

[2]assuming that all non-zero singular values are different, which is true for any "real" term-document matrix; see the remark following Lemma 2.

zero, which means that the relatedness scores will increase on the whole range from 1 to the full rank of the matrix. This corresponds to curves of the type shown in Figure 1(a). When $|a_1|$ is non-zero, then the larger it is compared to other terms in the collection, the more singular values will be smaller than $|a_1 - b_1|$, and the earlier the dimension of fall-off will come. This leads to graphs of the type shown in Figure 1(b).[3]

3.2 Adding perturbations

The following lemma shows that our definition of perfectly related terms is robust under small perturbations of the term-document matrix. In view of another application in Section 3.3, the lemma is formulated slightly more generally than would be necessary for this section. In the following we will write m_{ij} for the entries of an arbitrary matrix M, and $|M|_F$ for the Frobenius norm, which is the square root of $\sum m_{ij}^2$.

LEMMA 2. *Let A be a term-document matrix and let k be an integer such that the matrix U of the k most significant left singular vectors of A has two identical rows i and j. Let $\mathcal{E}$ be any matrix with $|\mathcal{E}|_F$ bounded by some fraction $f < 1/4$ of the gap between the singular values σ_k and σ_{k+1}, and let U' be the matrix of the k most significant left singular vectors of $A + \mathcal{E}$. Then,*

$$\left| \sum_{l=1}^{k} u_{il} u_{jl} - \sum_{l=1}^{k} u'_{il} u'_{jl} \right| \leq 8f.$$

Remark. In every real term-document matrix we have seen so far, a plot of the sorted singular values shows a very smooth curve, in particular, there is a gap between each pair of neighboring (non-zero) singular values. Under these circumstances the lemma above implies that sufficiently small perturbations change the curve of relatedness scores only little *at any dimension before it falls off.*

PROOF. By an adaption of Stewart's theorem on the perturbation of symmetric matrices [9, Theorem 8.6.5 on page 450] to arbitrary rectangular matrices, similarly as done in [17] and [3], it can be shown that $U' = UR + H$, where R is an orthogonal $k \times k$ matrix, and $|H|_F \leq 4f$.

Let $U'' = UR$ and let u_i^T, u_j^T, and $u_i''^T$, $u_j''^T$ denote the ith and jth row of U and U'', respectively. Then U'' shares two properties with U. First, by assumption, rows i and j of U are identical, and this property is invariant under right multiplication by any matrix, so that U'' has this property as well. Second, $u_i''^T = u_i^T R$ and $u_j''^T = u_j^T R$, so that $u_i''^T u_j'' = u_i^T R R^T u_j = u_i^T u_j$, that is, the dot product of rows i and j is the same for U'' as for U.

Let further $u_i'^T$, $u_j'^T$, and h_i^T, h_j^T denote the ith and jth row of U' and H, respectively. Then $u_i'^T = u_i''^T + h_i^T$ and $u_j'^T = u_j''^T + h_j^T$, and we now show that because H has small norm, the dot product of rows i and j of U' is close to that for U. For that, first write

$$\begin{aligned} u_i'^T u_j' &= (u_i''^T + h_i^T)(u_j'' + h_j) \\ &= u_i''^T u_j'' + u_i''^T h_j + h_i^T u_j'' + h_i^T h_j. \end{aligned}$$

[3]A generic example for term pairs with a very early dimension of fall-off would be two transcriptions of the same name, e.g., Chernov and Chernoff: these will occur in very related contexts but rarely co-occur together.

With $u_i''^T u_j'' = u_i^T u_j$, as established above, and writing u'' for the identical u_i'' and u_j'', this becomes

$$\begin{aligned} & u_i^T u_j + u''^T (h_i + h_j) + h_i^T h_j \\ \leq\ & u_i^T u_j + \sqrt{|u''||h_i + h_j|} + \sqrt{|h_i||h_j|}, \end{aligned}$$

where the inequality follows from Cauchy's inequality. Now $|u''| \leq 1$ because u'' is part of a row of an orthogonal matrix, and $|h_i|$ and $|h_j|$ are each bounded by $|H|$, and, by the triangle inequality, $|h_i + h_j|$ is bounded by $|H|$, too. This, finally, proves that $|u_i'^T u_j' - u_i^T u_j|$ is bounded by $2|H|_F$, which, in turn, is at most $8f$, as desired. □

Figure 2 gives a typical example of the effect of adding perturbation for two perfectly related terms. It is important to note that the *absolute values of the curve* can and do actually change quite a lot on the way from two perfectly related terms to a pair as it is found in a real collection, however, *the basic shape of the curve remains unchanged*: there is a phase of ascend in the beginning, and exactly one intermediate phase of descend, and drawn on its scale the curve looks rather smooth. (Whether another phase of ascend follows after the phase of descend or not will not be important for us.)

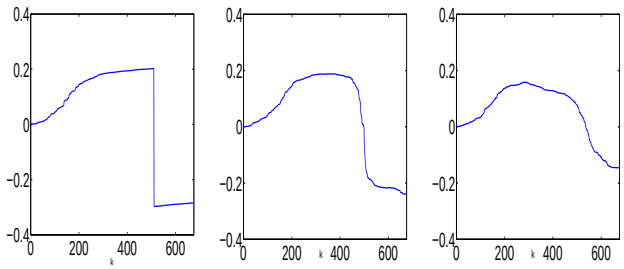

Figure 2: Curves of relatedness scores for two perfectly related terms (left), after a relatively small perturbation of the term-document matrix (middle), and after a relatively large such perturbation (right).

The three curves from Figure 2 were obtained as follows: we took the term-document matrix of the collections of abstracts mentioned in the caption of Figure 1, and modified two of its rows (terms) to obey the perfect-relatedness criterion of Definition 2. The curve for the in such a way modified matrix is shown on the left of Figure 2. For the curves in the middle and on the right, we added to this modified matrix a 0.05 and a 0.5 fraction, respectively, of the difference between the original and the modified matrix (adding the whole difference would give the graph for the original matrix).

3.3 Unrelated terms

In the previous section, we defined an ideal notion of perfectly related terms, found a very characteristic shape of their curves of relatedness scores, and verified that related terms from real collections do actually have curves of a similar shape. It remains to argue that *unrelated terms* give rise to curves of a (measurably) different kind.

For that consider the *co-occurrence graph* of a collection, which is an unweighted undirected graph that has a vertex for each term in the collection, and that has an edge between two vertices if and only if these two terms co-occur in at

least one document (that is, the corresponding entry in the term-document matrix is non-zero).

DEFINITION 3. *Two terms are called* completely unrelated *in a collection, if and only there is no path from the one term to the other in the co-occurrence graph of that collection.*

LEMMA 3. *The curve of relatedness scores for two completely unrelated terms is all zero. A small perturbation of the underlying term-document matrix affects that curve in a way that can be bounded exactly as stated in Lemma 2.*

PROOF. The first part of the lemma is an easy consequence of Theorem 1 in [14]. The second part is implied by Lemma 2. □

Indeed, the curves which we observe for pairs of intuitively unrelated terms (the vast majority of all term pairs) on real collections are of that kind: they vary around zero, change their direction many times and look zig-zagged. An example was given in Figure 1(c).

4. DIMENSIONLESS ALGORITHMS

One consequence of our findings in the last section is a surprising answer to one of the fundamental questions of spectral retrieval, as to what the appropriate choice of dimension (number of concepts) is. We can say that *every choice is inappropriate* for a significant fraction of the term pairs. Figure 3 gives an example to illustrate this point.

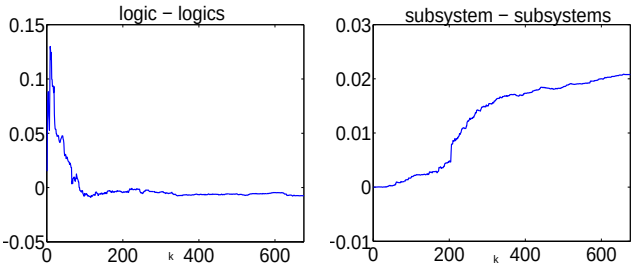

Figure 3: No single fixed dimension can do justice to both term pairs.

Both term pairs are intuitively strongly related (singular and plural of the same word) and both curves have the shape characteristic for related terms. However, the first curve reaches its peak at a very small k, where the relatedness scores of the second curve are still very low, but for larger k, when the scores of the second curve become large, the scores of the first curve actually become negative. We therefore cannot find a single fixed k such that both term pairs get the high relatedness score which they would deserve.

Another problem of any method that (implicitly or explicitly) works with relatedness scores at a fixed dimension is illustrated by an example in Figure 4. With respect to the underlying collection (still the CS abstracts), the two terms on the right are not related, which indeed shows in the shape of the curve of relatedness scores. For a significant and interesting range of dimensions, however, that curve reaches scores of the same order as those reached by curves for strongly related terms, like that on the left hand side of Figure 4.

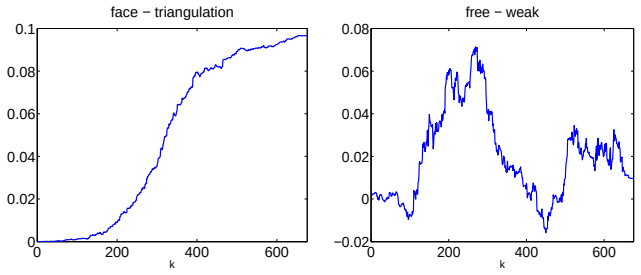

Figure 4: For many fixed dimensions these term pairs look indistinguishable from each other, although the different shapes of the two curves clearly tell them apart.

But our findings from the previous section also suggest a way to overcome these two problems. Algorithms for spectral retrieval should *not build on relatedness scores computed for a fixed dimension*, but *instead assess the overall shape of the curve of relatedness scores* for each term pair.

4.1 Algorithms TN

The following three steps describe one particularly simple such algorithm, where U denotes the matrix of the left singular vectors of the given term-document matrix A.

1. Normalize the rows of A to length 1, compute the SVD of this normalized matrix, and compute the rank r of the number 1 in the sorted list of the singular values.
2. For each pair of terms i, j compute the size z_{ij} of the set $\{1 \leq k \leq r : \sum_{l=1}^{k} u_{il}u_{jl} \leq 0\}$, that is, the number of dimensions at which the relatedness scores are at or below zero, until the potentially earliest dimension of fall-off.
3. Perform document expansion[4] with the $0-1$ matrix T, where $t_{ij} = 1$ if and only if $z_{ij} = 0$, that is, if and only if the corresponding curve of relatedness scores is never negative before the potentially earliest dimension of fall-off.

Let us quickly verify that this algorithm is in accordance with our theoretical findings from Section 3. By Lemma 1, we have that for all perfectly related terms $z_{ij} = 0$ and hence $t_{ij} = 1$. By Lemma 3, completely unrelated terms have all-zero curves, in which case $z_{ij} = r > 0$, and hence $t_{ij} = 0$. By Lemma 2, these assignments to T are invariant under small perturbations of the underlying term-document matrix.[5]

[4] To counter over-expansion effects, we actually split T into its diagonal part (which is the identity matrix) and the rest, do standard expansion as described in Section 2 for each of the matrices (which means no expansion for the diagonal part), and then add the two similarity scores.

[5] Strictly speaking, Lemma 2 does not rule out the possibility that the small change affected in an all-zero curve by a small perturbation is such that all values become slightly above zero, in which case our algorithm would set $t_{ij} = 1$. It is plausible, however, and we have verified it experimentally, that for a random perturbation this happens with negligible probability.

4.2 Algorithm TS

As pointed out already in the paragraph following Figure 1, curves of type (a) and (b) could also be distinguished from those of type (c) by the *smoothness* of the curves. This idea gives rise to the following alternative algorithm.

1. Compute the same matrix U as for TN.
2. For each pair of terms i, j, compute the smoothness s_{ij} of their curve as $(\max - \min)/\sum_{l=1}^{k} |u_{il}u_{jl}|$, where max and min are the maximum and minimum score of that curve, respectively. Observe that s_{ij} is 1 if and only if the scores go only up or only down, and that zig-zags push s_{ij} towards zero.
3. Perform document expansion[4] with the $0-1$ matrix T, where $t_{ij} = 1$ if and only if $s_{ij} \geq s$, for some threshold s. For our experiments we set s such that 0.2% of the entries in T are 1.

In contrast to TN, TS has a parameter: the smoothness threshold s. This can be seen as an advantage as well as a disadvantage. The disadvantage is that we have to find a good value for this parameter. The advantage is that it is a very intuitive parameter: it specifies how many related terms we want to consider. This is in sharp contrast to the choice of dimension in previous methods, which, as we pointed out already in the introduction, has no intuitive or natural setting.

4.3 Computational complexity

The computational complexity of our two dimensionless algorithms TN and TS is essentially that of basic LSI. All three require the computation of a singular value decomposition up to a certain dimension k. After that, LSI needs to map the term-document matrix to the k-dimensional latent space, which takes $O(k \cdot nz)$ basic numerical operations, where nz is the number of nonzero entries in the term-document matrix. The construction of TN/TS requires $O(k)$ operations per term pair, and actually expanding the term-document matrix requires $O(l \cdot nz)$ operations, where l is the average number of related terms of a term, giving a total of $O(k \cdot m^2 + l \cdot nz)$ operation. We can save on the m^2 by discarding pairs of terms that do not co-occur in at least one document, because, in practice, TN and TS never assign a 1 to such term pairs.

5. EXPERIMENTAL EVALUATION

We tested our algorithms on three test collections: the small Time collection [18] (3882×425), the significantly larger Reuters collection [15] (5701×21578), and the still larger Ohsumed collection [10] (99117×233445); in parentheses, the dimensions of the respective term-document matrices are given. In all cases we used stemming (Porter) and removed common stop-words as well as words that occurred in less than a certain number of documents. We measured average precision for 83 queries for Time, 120 queries for Reuters, and 63 queries for Ohsumed. For Time and Ohsumed, the available relevance rankings were used. Reuters comes only with topic labels, and we synthesized a query from each topic by taking the most significant terms of a random sample of documents from that topic, just as done, for example, in [6].

As competitors of our algorithms we chose four major spectral retrieval schemes from the literature: the basic latent semantic indexing scheme (LSI) from [4], the term-normalized variant (LSI-RN) proposed in [12], the correlation method (CORR) from [6], and iterative residual rescaling (IRR) from [2]. These are among the most well-known variants of LSI and moreover, each of the three fundamental types of variation of spectral retrieval, discussed in the next section, is covered by this selection.

As a baseline method we took ranking by plain cosine similarity (COS). The required singular vectors and values were computed from a standard tf.idf matrix for COS, LSI, IRR, and LSI-RN, and from a row-normalized matrix for CORR and our TN and TS, because that is what the theory of the latter methods asks for (note that row-normalization undoes tf.idf normalization). All computations were done in Matlab.

5.1 Results of the experiments

On all three collections, our two dimensionless algorithms consistently outperformed their four competitors. Moreover, the performance of those four varied significantly between different collections and different choices of dimension, while TS and TN consistently gave very similar results, although the numbers they are based on are mathematically quite different (for each term pair, the number of times its curve becomes negative versus the curve's smoothness). This further adds to the evidence built up in the previous sections that spectral retrieval works by identifying pairs of related terms and that the overall shape of the curve of relatedness scores is a more reliable indicator for term-pair relatedness than the score at any fixed dimension can be.

	Time	
COS	**63.2%**	63.2%
LSI	62.8%	61.6%
LSI-RN	58.6%	60.9%
CORR	59.1%	60.5%
IRR	62.2%	**63.5%**
TN	**64.9%**	
TS	**64.1%**	

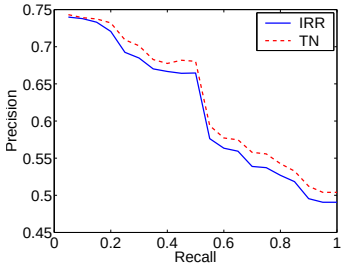

Figure 5: Average precision figures for the Time collection. For the four variants of LSI, figures are given for dimension 300 and 400. The right figure shows the averaged precision-recall graphs of TN versus IRR at its best dimension.

	Reuters		Ohsumed	
COS	36.2%	36.2%	**13.2%**	**13.2%**
LSI	28.0%	32.0%	6.1%	6.9%
LSI-RN	**37.0%**	**36.9%**	12.9%	13.0%
CORR	32.2%	32.3%	10.3%	10.9%
TN	**41.9%**		**14.4%**	
TS	**42.9%**		**15.3%**	

Figure 6: Average precision figures for the Reuters and Ohsumed collections. For the three variants of LSI, figures are for dimensions 800 and 1000 for Reuters and dimensions 1000 and 1200 for Ohsumed. For TN and TS, the relatedness scores were computed up to dimension 1000 for Reuters and up to dimension 1200 for Ohsumed.

We remark that dimension 400 is close to the optimal dimension for each of the four LSI variants in Figure 5. In Figure 6, IRR does not appear because it is computationally too expensive for collections of this size; indeed, all experiments from [2] were done on less than a thousand documents. Note that the low average precisions of around 15% for Ohsumed are actually quite substantial given that on average 40 documents from over 200,000 were relevant for each query. To make the SVD computation for the Ohsumed collection feasible, we kept only the 12464 most significant terms. For the LSI-style methods, all other terms were simply discarded. For TN and TS, we computed the expansion matrices for the restricted number of terms, but then used it to actually expand the originally matrix (which is possible only by the sparseness of our expansion matrices, another advantage of our approach).

5.2 Binary versus fractional relatedness

One surprising aspect of our experimental findings is that algorithms which do a simple binary classification into related and unrelated term pairs outperform schemes which seem to have additional power by giving a fractional assessment for each term pair.

An intuitive explanation is that while in principle it is of course reasonable to deem some term pairs more related than others, it is plausible that deducing such fine distinctions from mere co-occurrence data can add more noise than precision.

A more formal explanation comes from the histogram in Figure 7, which illustrates how many curves have scores at or below zero at how many dimensions (before the earliest possible dimension of fall-off according to our theory). The main observation here is that most curves are at or below zero at either very few dimensions (and the vast majority of these never touches zero) or at quite a lot of dimensions. In fact, this conforms well with our theoretical findings: if all term pairs were either perfectly related or completely unrelated, we would have non-zero counts only at both endpoints of the histogram (and an infinitesimal perturbation would move the count for the unrelated terms to the middle of the histogram).

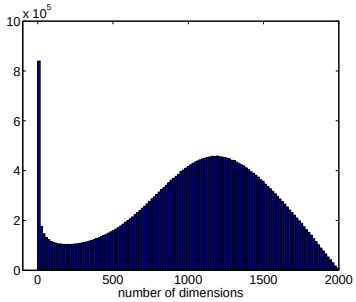

Figure 7: Histogram of the number of dimensions where the curves of relatedness scores are at or below zero (for Reuters).

6. THE MANY VARIATIONS OF LSI

Our experimental results confirmed the observation of several previous authors that the improvements entailed by the various variations of LSI are not consistent over collections of different sizes and types. As the authors of a recent SIGIR publication remark, "to the best of our knowledge, [so far] no study systematically evaluated these fundamental choices" [19].

In this section we will characterize how our curves of relatedness scores are affected by three fundamental types of variations found in the literature: rescaling of the terms or documents, scaling by the singular values, and iterative residual rescaling. This covers, in particular, the selection listed in [19].

The bottom line of this section will be that on the one hand, relatedness scores for individual dimensions can indeed change a lot, which reflects the ability of each of these variations to boost retrieval quality. On the other hand, we will see that the characteristic of the shape of the curves which we found indicative of term relatedness remains basically unaffected by any of these normalizations. This is further evidence that the shapes we identified lie at the bottom of what makes spectral retrieval work in practice.

6.1 Rescaling of the terms or documents

Many authors have normalized the rows or columns of the term-document matrix prior to computing the singular vectors. For the various methods listed in [6], for example, the rows of the term-document matrix are first centered and then normalized to length 1. The following lemma says that our definition of perfectly related terms is invariant under any combination of such normalizations.

LEMMA 4. *Let A be a matrix of the form as in Definition 2. Then A remains a matrix of the same form after normalizing the rows or columns with respect to an arbitrary norm, and after centering of the rows or columns, or any combination of these.*

PROOF. It suffices to observe that the entries in the two rows of the perfectly related terms synonyms are the same up to permutation (hence their norms and means are equal), and that the same applies to columns j and $n_1 + j$ of A, for $j = 1, \ldots, n_1$, where n_1 is the number of documents where the, say, first term occurs without the second. □

6.2 Rescaling of the singular vectors

After the matrix U_k of the top k singular vectors has been computed, there is an option of also rescaling the rows or columns of that U_k, or both. A widely used type of rescaling here is to multiply the columns of U_k by the corresponding singular values raised to some power κ. For example, the original work on LSI [4] sets $\kappa = -1$, while [12] and [6] advocate $\kappa = 1$. The most frequently used setting is $\kappa = 0$.

The effect of κ on our curves can be characterized as follows. Since the singular values are sorted in descending order, positive values of κ will *stretch* the curves towards the beginning but *shrink* it towards the end. For negative values of κ, the opposite happens, which, given the shape of the curve for related terms, seems less favorable for LSI and its variants. Indeed, the inventors of LSI used $\kappa = -1$ only in their very first work [4], while they used $\kappa = 0$ in all of their many follow-up works.

The overall shape of the curves remains basically unchanged by a change of κ, however, and no particular choice of κ can therefore overcome the inherent problems illustrated by Figures 3 and 4. This shows in the inconsistent results reported in [6], as well as in our experimental results from Section 5.1.

Husbands et al. [12] observed that basic LSI gives undue preference to frequent terms and suggested to normalize the

rows of U_k (after scaling its columns by the singular values). The results were again inconclusive, however. In [19], the influence of the size of the collection on this phenomenon was discussed, however without a theoretical underpinning and also without a conclusive answer. From the point of view of our work, row-normalization of U simply brings all curves to the same order of magnitude, which partially overcomes the problem addressed in Figure 4 but does not address the problem addressed in Figure 3.

6.3 Iterative residual rescaling

Ando and Lee [1] [2] proposed *iterative residual rescaling (IRR)* to obtain, from an $m \times n$ term-document matrix A, a sequence of pairwise orthogonal m-vectors $u_1, u_2, \ldots$, where u_1 is just the top left singular vector of A, u_2 is the top left singular vector of a column-normalized version of $A - u_1 u_1^T A$, and so on. The rationale of this method is to balance the bad effects when the assumed underlying concepts are present in the collection in widely different proportions.

Due to the pairwise orthogonality of the $u_1, u_2, \ldots$, IRR can be viewed as document expansion just like any of the other spectral retrieval methods. It is also not hard to see that two perfectly related terms, according to Definition 2, lead to the special singular vector from Lemma 1, and hence to the characteristic shape of the curve from Figure 2. The main observation needed is that if u is any vector with identical entries at the indices corresponding to the two perfectly related terms, then for any matrix $\tilde{A}$ of the form of Definition 2, $\tilde{A} - uu^T\tilde{A}$ will again be of the same form. The problems illustrated in Figures 3 and 4 therefore affect IRR as much as they do affect any of the other methods.

7. CONCLUSIONS AND OUTLOOK

We have introduced the *curves of relatedness scores* as a new angle of looking at retrieval schemes based on spectral analysis. We have given strong evidence, both theoretical and experimental, that identifying related terms on the basis of the shape of these curves is at the heart of what makes spectral retrieval work in practice.

Our new dimensionless algorithms do not only outperform previous schemes that work with relatedness scores computed from individual dimensions, but they are also more intuitive: they do nothing but identify pairs of related terms (in a 0-1 fashion), and expand documents via these relations. Note that this immediately gives a thesaurus-like functionality, which could be used, for example, in an interactive scenario to prompt a user with a list of possible synonyms for each word of her query (possibly ranked by the smoothness scores computed by our algorithm TS).

Our view of spectral retrieval as a document expansion also makes it straightforward to incorporate external knowledge into the retrieval process. In [13], first ad-hoc steps have been taken in that direction. Our findings appear to be a good foundation for a more principled approach.

All spectral retrieval schemes so far, including our new algorithms, give rise to a *symmetric* document expansion matrix T, but word pairs like "nucleic" and "acid" actually call for an *asymmetric* such matrix: whenever there is "nucleic" there is "acid", but not vice versa. This is actually a frequent phenomenon and we would expect algorithms which are able to consider this asymmetry to give a further boost to the effectiveness of spectral retrieval.

8. REFERENCES

[1] R. K. Ando. Latent semantic space: Iterative scaling improves precision of inter-document similarity measurement. In *Proceedings SIGIR'00*, pages 216–223, 2000.

[2] R. K. Ando and L. Lee. Iterative residual rescaling: An analysis and generalization of LSI. In *Proceedings SIGIR'01*, pages 154–162, 2001.

[3] Y. Azar, A. Fiat, A. Karlin, F. McSherry, and J. Saia. Spectral analysis of data. In *Proceedings STOC'01*, 2001.

[4] S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407, 1990.

[5] C. H. Q. Ding. A similarity-based probability model for latent semantic indexing. In *Proceedings SIGIR'99*, pages 58–65, 1999.

[6] G. Dupret. Latent concepts and the number of orthogonal factors in latent semantic analysis. In *Proceedings SIGIR'03*, pages 221–226, 2003.

[7] G. Dupret. Latent semantic indexing with a variable number of orthogonal factors. In *Proceedings RIAO'04*, 2004.

[8] M. Efron. *Eigenvalue-based Estimators for Optimal Dimensionality Reduction in Information Retrieval*. PhD thesis, U. North Carolina, Chapel Hill, 2004.

[9] G. H. Golub and C. F. Van Loan. *Matrix Computations*. Johns Hopkins, third edition, 1996.

[10] W. Hersh, C. Buckley, T. Leone, and D. Hickam. Ohsumed: An interactive retrieval evaluation and new large test collection for research. In *Proceedings SIGIR'94*, pages 192–201, 1994.

[11] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning Journal*, 42(1):177–196, 2001.

[12] P. Husbands, H. Simon, and C. H. Q. Ding. On the use of the singular value decomposition for text retrieval. *Computational Information Retrieval*, pages 145–156, 2001.

[13] S. D. Kamvar, D. Klein, and C. D. Manning. Spectral learning. In *Proceedings IJCAI'03*, 2003.

[14] A. Kontostathis and W. M. Pottenger. Detecting patterns in the LSI term-term matrix. In *Proceedings ICDM'02 Workshop on Foundations of Data Mining and Discovery*, 2002.

[15] D. Lewis. Reuters-21578 text categorization test collection, 1997.

[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *Proceedings NIPS'01*, 2001.

[17] C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: a probabilistic analysis. In *Proceedings PODS'98*, pages 159–168, 1998.

[18] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[19] C. Tang, S. Dwarkadas, and Z. Xu. On scaling latent semantic indexing for large peer-to-peer systems. In *Proceedings SIGIR'04*, pages 112–121, 2004.