# Detecting Patterns in the LSI Term-Term Matrix

April Kontostathis
*Lehigh University*
*19 Memorial Drive West*
*Bethlehem, PA 18015*
*apk5@lehigh.edu*

William M. Pottenger, Ph.D.
*Lehigh University*
*19 Memorial Drive West*
*Bethlehem, PA 18015*
*billp@cse.lehigh.edu*

## Abstract

*Higher order co-occurrences play a key role in the effectiveness of systems used for text mining. A wide variety of applications use techniques that explicitly or implicitly employ a limited degree of transitivity in the co-occurrence relation. In this work we show use of higher orders of co-occurrence in the Singular Value Decomposition (SVD) algorithm and, by inference, on the systems that rely on SVD, such as LSI. Our empirical and mathematical studies prove that term co-occurrence plays a crucial role in LSI.*

*This work is the first to study the values produced in the truncated term-term matrix, and we have discovered an explanation for why certain term pairs receive a high similarity value, while others receive low (and even negative) values. Thus we have discovered the basis for the claim that is frequently made for LSI: LSI emphasizes important semantic distinctions (latent semantics) while reducing noise in the data*

## 1. Introduction

Term clustering is an approach that researchers have used to convert the original words of a document into more effective content identifiers for use in text mining applications. Unsupervised term clustering algorithms generally consist of two phases. In the first phase term-term similarity is determined. The second phase uses the term-term similarities to develop clusters of terms. Latent Semantic Indexing (LSI) [4] is a text mining algorithm that is based on Singular Value Decomposition (SVD). The values in the truncated term-term matrix produced by SVD can be treated as similarity measures for input to a clustering algorithm. In this work we present simple, highly intuitive theoretical framework for understanding the values in the LSI term-term matrix. This framework is a critical step towards our goal of detecting correlations between the values in the term-term matrix and the higher order co-occurrence implicit in the data. This work also presents preliminary work toward detecting patterns in the data.

LSI has been applied to a wide variety of learning tasks, such as classification [16] and filtering [6,7]. LSI is a vector space approach for modeling documents, and many have claimed that the technique brings out the 'latent' semantics in a collection of documents [4,5].

LSI is based on well known mathematical technique called Singular Value Decomposition (SVD). The algebraic foundation for Latent Semantic Indexing (LSI) was first described in [4] and has been further discussed in [2][3]. These papers describe the SVD process and interpret the resulting matrices in a geometric context. The SVD, truncated to k dimensions, gives the best rank-k approximation to the original matrix. In [15], Wiemer-Hastings shows that the power of LSI comes primarily from the SVD algorithm.

Other researchers have proposed theoretical approaches to understanding LSI. [17] describes LSI in terms of a subspace model and proposes a statistical test for choosing the optimal number of dimensions for a given collection. [14] discusses LSI's relationship to statistical regression and Bayesian methods. [8] constructs a dual probability model for LSI using the cosine similarity measure.

Although other researchers have explored the SVD algorithm to provide an understanding of SVD-based information retrieval systems, to our knowledge, only Schütze has studied the values produced by LSI [13]. We expand upon this work, showing here that SVD exploits higher order term co-occurrence in a collection, and providing insight into the origin of the values produced in the term-term matrix.

This work, for the first time, provides a model for understanding LSI semantically – a goal that has been quite elusive for over 12 years since the technique was first introduced. Our framework is based on the concept of term co-occurrences. Term co-occurrence data is implicitly or explicitly used for almost every advanced application in textual data mining.

This work provides the theoretical foundation for understanding the use of limited transitivity in LSI. Eventually, the patterns we are detecting will be used to approximate the SVD algorithm, which is resource intensive [5,6,7], at much lower cost. In [10], we describe an unsupervised learning algorithm that develops clusters of terms, using the LSI term-term matrix to define the similarity between terms. Use both similarity and anti-similarity measures to develop clusters. Our results show a significant improvement in performance when the positive and negative clusters are used in a search and retrieval application.

The goal of the current line of work is a theoretically sound, effective and efficient unsupervised clustering algorithm that can be applied to a wide variety of textual data mining applications. The theoretical framework we provide here is a crucial step in this research project.

In section 2 we present a simple example of the use of third-order co-occurrence in LSI. Section 3 presents a mathematical proof of term transitivity within LSI. In section 4 we identify some initial patterns in the data.

## 2. Co-occurrence in LSI – An Example

The data for the following example is taken from [4]. In this paper, the authors describe an example with 12 terms and 9 documents. The term-term matrix for this collection is shown in table 1.

The SVD process used by LSI decomposes the matrix into three matrices: T, a term by dimension matrix, S a singular value matrix, and D, a document by dimension matrix. The number of dimensions is min (t, d) where t = number of terms and d = number of documents. The original matrix can be obtained, through matrix multiplication of $TSD^T$. The reader is referred to [4] for the T, S, and D matrices. In the LSI system, the T, S and D matrices are truncated to k dimensions. Dimensionality reduction reduces "noise" in the term–term matrix resulting in a richer word relationship structure that reveals latent semantics present in the collection. After dimensionality reduction the term-term matrix can be approximated using the formula $T_kS_k(T_kS_k)^T$. The term-term matrix, after reduction to 2 dimensions, is shown in table 2.

We will assume that the value in position (i,j) of the matrix represents the *similarity* between term i and term j in the collection. As can be seen in table 2, *user* and *human* now have a value of .94, representing a strong similarity, where before the value was zero. In fact, *user* and *human* is an example of second order co-occurrence. The relationship between *user* and *human* comes from

**Table 1: Deerwester Term by Term Matrix**

| | human | interface | computer | user | system | response | time | EPS | Survey | trees | graph | minors |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| human | x | 1 | 1 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| interface | 1 | x | 1 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| computer | 1 | 1 | x | 1 | 1 | 1 | 1 | 0 | 1 | 0 | 0 | 0 |
| user | 0 | 1 | 1 | x | 2 | 2 | 2 | 1 | 1 | 0 | 0 | 0 |
| system | 2 | 1 | 1 | 2 | x | 1 | 1 | 3 | 1 | 0 | 0 | 0 |
| response | 0 | 0 | 1 | 2 | 1 | x | 2 | 0 | 1 | 0 | 0 | 0 |
| time | 0 | 0 | 1 | 2 | 1 | 2 | x | 0 | 1 | 0 | 0 | 0 |
| EPS | 1 | 1 | 0 | 1 | 3 | 0 | 0 | x | 0 | 0 | 0 | 0 |
| Survey | 0 | 0 | 1 | 1 | 1 | 1 | 1 | 0 | x | 0 | 1 | 1 |
| trees | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | x | 2 | 1 |
| graph | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 2 | x | 2 |
| minors | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 2 | x |

**Table 2: Deerwester Term by Term Matrix, truncated to two dimensions**

| | human | interface | computer | user | system | response | time | EPS | Survey | trees | graph | minors |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| human | 0.62 | 0.54 | 0.56 | 0.94 | 1.69 | 0.58 | 0.58 | 0.84 | 0.32 | -0.32 | -0.34 | -0.25 |
| interface | 0.54 | 0.48 | 0.52 | 0.87 | 1.50 | 0.55 | 0.55 | 0.73 | 0.35 | -0.20 | -0.19 | -0.14 |
| computer | 0.56 | 0.52 | 0.65 | 1.09 | 1.67 | 0.75 | 0.75 | 0.77 | 0.63 | 0.15 | 0.27 | 0.20 |
| user | 0.94 | 0.87 | 1.09 | 1.81 | 2.79 | 1.25 | 1.25 | 1.28 | 1.04 | 0.23 | 0.42 | 0.31 |
| system | 1.69 | 1.50 | 1.67 | 2.79 | 4.76 | 1.81 | 1.81 | 2.30 | 1.20 | -0.47 | -0.39 | -0.28 |
| response | 0.58 | 0.55 | 0.75 | 1.25 | 1.81 | 0.89 | 0.89 | 0.80 | 0.82 | 0.38 | 0.56 | 0.41 |
| time | 0.58 | 0.55 | 0.75 | 1.25 | 1.81 | 0.89 | 0.89 | 0.80 | 0.82 | 0.38 | 0.56 | 0.41 |
| EPS | 0.84 | 0.73 | 0.77 | 1.28 | 2.30 | 0.80 | 0.80 | 1.13 | 0.46 | -0.41 | -0.43 | -0.31 |
| Survey | 0.32 | 0.35 | 0.63 | 1.04 | 1.20 | 0.82 | 0.82 | 0.46 | 0.96 | 0.88 | 1.17 | 0.85 |
| trees | -0.32 | -0.20 | 0.15 | 0.23 | -0.47 | 0.38 | 0.38 | -0.41 | 0.88 | 1.55 | 1.96 | 1.43 |
| graph | -0.34 | -0.19 | 0.27 | 0.42 | -0.39 | 0.56 | 0.56 | -0.43 | 1.17 | 1.96 | 2.50 | 1.81 |
| minors | -0.25 | -0.14 | 0.20 | 0.31 | -0.28 | 0.41 | 0.41 | -0.31 | 0.85 | 1.43 | 1.81 | 1.32 |

the transitive relation: *user* co-occurs with *interface* and *interface* co-occurs with demonstrates the value of the LSI system, since queries on the term *user* would correctly result in documents containing the term *human* in the context of this collection.

A closer look reveals a value of 0.15 in the relationship between *trees* and *computer*. Looking at the co-occurrence path gives us an explanation as to why these terms received a positive (although weak) similarity value. From table 1, we see that *trees* co-occurs with *graph*, *graph* co-occurs with *survey,* and *survey* co-occurs with *computer*. Hence the *trees/computer* relationship is an example of third order co-occurrence. In section 4 we present trend data that explains term-term matrix values in terms of the number of connectivity paths between terms.

## 3. Transitivity and the SVD

In this section we present mathematical proof that the SVD algorithm encapsulates term co-occurrence information. Specifically we show that a connectivity path exists for every nonzero element in the truncated matrix.

We begin by setting up some notation. Let A be a term by document matrix. The SVD process decomposes A into three matrices: a term by dimension matrix, T, a diagonal matrix of singular values, S, and a document by dimension matrix D. The original matrix is re-formed by multiplying the components, $A = TSD^T$. When the components are truncated to k dimensions, a reduced representation matrix, $A_k$ is formed as $A_k = T_kS_kD_k^T$ [4].

The term-term co-occurrence matrices for the full matrix and the truncated matrix are [4]:

(3a) $B = TSST^T$

(3b) $Y = T_kS_kS_kT_k^T$

We note that elements of B represent term co-occurrences in the collection, and $b_{ij} >= 0$ for all i and j. If term i and term j co-occur in any document in the collection, $b_{ij} > 0$. Matrix multiplication results in equations 4a and 4b for the ij$^{th}$ element of the co-occurrence matrix and the truncated matrix, respectively. Here $u_{ip}$ is the element in row i and column p of the matrix T, and $s_p$ is the p$^{th}$ largest singular value.

(4a) $$b_{ij} = \sum_{p=1}^{m} s_p^2\, u_{ip} u_{jp}$$

(4b) $$y_{ij} = \sum_{p=1}^{k} s_p^2\, u_{ip} u_{jp}$$

$B^2$ can be represented in terms of the T and S:

$$B^2 = (TSST^T)(TSST^T) = TSS(T^TT)SST^T = TSSSST^T = TS^4T^T$$

An inductive proof can be used to show:

(5) $B^n = TS^{2n}T^T$

And the element $b_{ij}^n$ can be written:

(6) $$b_{ij}^n = \sum_{p=1}^{m} s_p^{2n}\, u_{ip} u_{jp}$$

To complete our argument, we need two lemmas related to the powers of the matrix B.

*Lemma 1*: *Let i and j be terms in a collection, there is a transitivity path of order <= n between the terms, iff the ij$^{th}$ element of $B^n$ is nonzero.*

*Lemma 2: If there is no transitivity path between terms i and j, then the* ij$^{th}$ *element of $B^n$ ($b_{ij}^n$) is zero for all n.*

The proof of these lemmas can be found in [11]. We are now ready to present our theorem.

*Theorem 1: If the ij$^{th}$ element of the truncated term by term matrix, Y, is nonzero, then there is a transitivity path between term i and term j.*

We need to show that if $y_{ij} \neq 0$, then there exists terms q1, … , qn n >= 0 such that $b_{i\,q1} \neq 0$, $b_{q1\,q2} \neq 0$, …. $b_{qn\,j} \neq 0$. Alternately, we can show that if there is no path between terms i and j, then $y_{ij} = 0$ for all k.

Assume the T and S matrices have been truncated to k dimensions and the resulting Y matrix has been formed. Furthermore, assume there is no path between term i and term j. Equation (4b) represents the $y_{ij}$ element. Assume that $S_1 > S_2 > S_3 > … > S_m > 0$. By lemma 2, $b_{ij}^n = 0$ for all n. Dividing (6) by $s_1^{2n}$, we conclude that

$$u_{i1}u_{j1} + \sum_{p=2}^{m} \left(\frac{s_p}{s_1}\right)^{2n} u_{ip} u_{jp} = 0$$

We take the limit of this equation as n → ∞, and note that $(s_p/s_1) < 1$ when $2 <= p <= m$. Then as n→ ∞, $(s_p/s_1)^{2n} \rightarrow 0$ and the summation term reduces to zero. We conclude that $u_{i1}u_{j1} = 0$. Substituting back into $b_{ij}^n$ we have:

$$\sum_{p=2}^{m} s_p^{2n}\, u_{ip} u_{jp} = 0$$

Dividing by $s_2^{2n}$ yields:

$$u_{i2}u_{j2} + \sum_{p=3}^{m} \left(\frac{s_p}{s_2}\right)^{2n} u_{ip} u_{jp} = 0 \text{ for all n.}$$

Taking the limit as n→ ∞, we have that $u_{i2}u_{j2} = 0$. If we apply the same argument m times we will obtain $u_{ip}u_{jp} = 0$ for all p such that $1 <= p <= m$. Substituting back into (4b) shows that $y_{ij} = 0$ for all k.

The argument thus far depends on our assumption that $S_1 > S_2 > S_3 > \ldots > S_m$. When using SVD it is customary to truncate the matrices by removing all dimensions whose singular value is below a given threshold [5]; however, for our discussion, we will merely assume that, if $s_1 > s_2 > \ldots > s_{z-1} > s_z = s_{z+1} = s_{z+2} = \ldots = s_{z+w} > s_{z+w+1} > \ldots > s_m$ for some z and some w >= 1, the truncation will either remove all of the dimensions with the duplicate singular value, or keep all of the dimensions with this value.

We need to examine two cases. In the first case, $z > k$ and the z … z+w dimensions have been truncated. In this case, the above argument shows that either $u_{i\,q} = 0$ or $u_{j\,q} = 0$ for all $q <= k$ and, therefore, $y_{ij} = 0$.

To handle the second case, we assume that $z < k$ and the z … z+w dimensions have not been truncated and rewrite equation (6) as:

$$b_{ij}^{n} = \sum_{p=1}^{z-1} s_p^{2n}\, u_{ip} u_{jp} + \sum_{p=z}^{z+w} s_z^{2n}\, u_{ip} u_{jp} +$$

$$\sum_{p=z+w+1}^{m} s_p^{2n}\, u_{ip} u_{jp} = 0$$

The argument above can be used to show that $u_{ip}\, u_{jp} = 0$ for $p <= z-1$, and the first summation can be removed. After we divide the remainder of the equation by $s_z^{2n}$ :

$$b_{ij}^{n} = \sum_{p=z}^{z+w} u_{ip} u_{jp} +$$

$$\sum_{p=z+w+1}^{m} \left(\frac{s_p}{s_z}\right)^{2n} u_{ip} u_{jp} = 0$$

Taking the limit as $n \rightarrow \infty$, we conclude that $\sum_{p=z}^{z+w} u_{ip} u_{jp} = 0$, and $b_{ij}^{n}$ is reduced to:

$$b_{ij}^{n} = \sum_{p=z+w+1}^{m} s_p^{2n}\, u_{ip} u_{jp} = 0$$

Again using the argument above, we can show that $u_{ip}\, u_{jp} = 0$ for $z+w+1 <= p <= m$. Furthermore,

$$y_{ij} = \sum_{p=1}^{z-1} s_p^{2}\, u_{ip} u_{jp} + \sum_{p=z}^{z+w} s_z^{2}\, u_{ip} u_{jp} +$$

$$\sum_{p=z+w+1}^{k} s_p^{2}\, u_{ip} u_{jp} = 0$$

And our proof is complete.

## 4. Experimental Results

We chose the MED, CRAN, CISI and LISA collections for our study of the higher order co-occurrence in LSI. MED is a commonly studied collection of medical abstracts. It consists of 1033 documents and 5823 terms. CISI is a set of 1460 information science abstracts containing 5609 terms. The MED and CISI collections were used for the landmark Deerwester paper [4]. CRAN is the Cranfield collection, one of the earliest collections available to information retrieval researchers. CRAN consists of 1398 documents and 4612 terms.

Experiments on a larger collection, LISA, were also performed. The LISA (Library and Information Science Abstracts) test collection was developed by the University of Sheffield, England. The LISA collection was processed in two ways. The first was an extraction of words only, resulting in a collection with 6004 documents and 18429 terms. We will refer to this collection as LISA Words. Next the HDDI™ Collection Builder [12] was used to extract maximal length noun phrases. This collection contains 5998 documents (no noun phrases were extracted from several short documents) and 81,879 terms. The experimental results for the LISA Noun Phrase collection were restricted to 1000 randomly chosen terms (due to processing time considerations). However, for each of the 1000 terms, all co-occurring terms (up to 81,879) were processed, giving us confidence that this data set accurately reflects the scalability of our result.

### 4.1 Methodology

Our experiments captured four main features of these data sets: the order of co-occurrence for each pair of terms in the truncated term-term matrix (shortest path length), the order of co-occurrence for each pair of terms in the original (not truncated) matrix, the distribution of the similarity values produced in the term-term matrix, categorized by order of co-occurrence, and the number of second-order co-occurrence paths between each set of terms.

In order to complete these experiments, we needed a program to perform the SVD decomposition. The SVDPACK suite [1] that provides eight algorithms for decomposition was selected because it was readily available, as well as thoroughly tested. The singular values and vectors were input into our algorithm.

### 4.2 Results

The order of co-occurrence summary from the TraceCo-occurrence program for all of the collections is shown in table 3. Fifth order co-occurrence was the highest degree of transitivity observed. In is interesting to note that the noun phrase collection is the only collection that resulted in a co-occurrence order higher than 3. It is important also to note that the order 2 and

Table 3: Order of Co-occurrence Summary Data (k=100 for all Collections)

| Collection | First | Second | Third | Fourth | Fifth | Sixth |
|---|---|---|---|---|---|---|
| MED Truncated | 1,110,485 | 15,867,200 | 17,819 | - | - | |
| MED Original | 1,110,491 | 15,869,045 | 17,829 | - | - | |
| CRAN Truncated | 2,428,520 | 18,817,356 | 508 | - | - | |
| CRAN Original | 2,428,588 | 18,836,832 | 512 | - | - | |
| CISI Truncated | 2,327,918 | 29,083,372 | 17,682 | - | - | |
| CISI Original | 2,328,026 | 29,109,528 | 17,718 | - | - | |
| LISA Words Truncated | 5,380,788 | 308,556,728 | 23,504,606 | - | - | |
| LISA Words Original | 5,399,343 | 310,196,402 | 24,032,296 | | | |
| LISA Noun Phrase Truncated | 51,350 | 10,976,417 | 65,098,694 | 1,089,673 | 3 | |
| LISA Noun Phrase Original | 51,474 | 11,026,553 | 68,070,600 | 2,139,117 | 15,755 | 34 |

Table 4: Average Number of paths by term-term value for LISA Words, k=100

| Term Term Matrix Value | Degree 1 Pairs | Degree 2 Pairs | Degree 3 Pairs | Degree 2 Paths | Average No Paths for Degree 1 pairs | Average No Paths for Degree 2 pairs | Average No Paths for Degree 3 pairs |
|---|---|---|---|---|---|---|---|
| less than -0.2 | 21,946 | 186,066 | - | 66,323,200 | 3,022 | 356 | - |
| -0.2 to -0.1 | 10,012 | 422,734 | 2 | 59,418,198 | 5,935 | 141 | 29,709,099 |
| -0.1 to 0.0 | 76,968 | 127,782,170 | 18,398,756 | 1,587,147,584 | 20,621 | 12 | 86 |
| 0.0 to 0.1 | 1,670,542 | 175,021,904 | 5,105,848 | 4,026,560,130 | 2,410 | 23 | 789 |
| 0.1 to 0.2 | 662,800 | 3,075,956 | - | 721,472,948 | 1,089 | 235 | - |
| 0.2 to 0.3 | 418,530 | 974,770 | - | 389,909,456 | 932 | 400 | - |
| 0.3 to 0.4 | 320,736 | 439,280 | - | 259,334,214 | 809 | 590 | - |
| 0.4 to 0.5 | 309,766 | 232,584 | - | 195,515,980 | 631 | 841 | - |
| 0.5 to 0.6 | 241,466 | 136,742 | - | 151,687,510 | 628 | 1,109 | - |
| 0.6 to 0.7 | 158,210 | 85,472 | - | 117,150,688 | 740 | 1,371 | - |
| 0.7 to 0.8 | 128,762 | 56,042 | - | 96,294,828 | 748 | 1,718 | - |
| 0.8 to 0.9 | 113,826 | 38,156 | - | 81,799,460 | 719 | 2,144 | - |
| 0.9 to 1.0 | 119,440 | 25,958 | - | 72,273,400 | 605 | 2,784 | - |
| 1.0 to 2.0 | 547,354 | 70,616 | - | 393,001,792 | 718 | 5,565 | - |
| 2.0 to 3.0 | 208,238 | 6,678 | - | 172,335,854 | 828 | 25,807 | - |
| 3.0 to 4.0 | 105,332 | 1,112 | - | 98,575,368 | 936 | 88,647 | - |
| 4.0 to 5.0 | 62,654 | 334 | - | 64,329,366 | 1,027 | 192,603 | - |
| 5.0 to 6.0 | 40,650 | 78 | - | 45,174,210 | 1,111 | 579,157 | - |
| 6.0 to 7.0 | 28,264 | 36 | - | 33,514,804 | 1,186 | 930,967 | - |
| 7.0 to 8.0 | 21,316 | 24 | - | 26,533,666 | 1,245 | 1,105,569 | - |
| over 8.0 | 113,976 | 16 | - | 188,614,174 | 1,655 | 11,788,386 | - |

order 3 co-occurrences significantly reduce the sparseness of the original data. The lines labeled <Collection> Original indicate the number of pairs with co-occurrence order n determined from a trace of the original term-term matrix. Note that LSI finds over 99% of term co-occurrences present in the data for the first four collections, and 95% for the LISA Noun Phrase collection.

Table 4 shows the weight distribution for the LISA Words data, for k=100. The degree 2 pair weights range from -.3 to values larger than 8. These co-occurrences will result in significant differences in document matching when the LSI algorithm is applied in a search and retrieval application. However, the weights for the degree 3 transitivity pairs are between -.2 and .1, adding a relatively small degree of variation to the final results. Also note that many of the original term-term co-occurrences (degree 1 pairs) are reduced to nearly zero, while others are significantly larger.

Table 4 also shows the average number of paths by term-term value range for LISA Words. Clearly the degree 2 pairs that have a similarity close to zero have a much smaller average number of paths than the pairs with either higher negative or positive similarities. MED and CISI showed similar trends. This data provides insight into understanding the values produced by SVD in the truncated term-term matrix.

Notice that the degree 1 pairs with higher average number of paths tend to have negative similarity values, pairs with few co-occurrences tend to receive low similarity values, and pairs with a moderate number of co-occurrence paths tend to receive high similarity values. This explains how LSI emphasizes important semantic distinctions, while de-emphasizing terms that co-occur frequently with many other terms (reduces 'noise'). On the other hand, degree 2 pairs with many paths of connectivity tend to receive high similarity values, while those with a moderate number tend to receive negative values. These degree 2 pairs with high values can be considered the 'latent semantics' that are emphasized by LSI.

## 5. Conclusions and Future Work

Higher order co-occurrences play a key role in the effectiveness of systems used for text mining. We have explicitly shown use of higher orders of co-occurrence in the Singular Value Decomposition (SVD) algorithm and, by inference, on the systems that rely on SVD, such as LSI. Our empirical and mathematical studies prove that term co-occurrence plays a crucial role in LSI. The work shown here will find many practical applications.

This work is the first to study the values produced in the truncated term-term matrix, and we have discovered an explanation for why certain term pairs receive a high similarity value, while others receive low (and even negative) values. Thus we have discovered the basis for the claim that is frequently made for LSI: LSI emphasizes important semantic distinctions (latent semantics) while reducing noise in the data. The correlation between the number of connectivity paths between terms and the value produced in the truncated term-term matrix is another important component in the theoretical foundation for LSI.

Future plans include development of an approximation algorithm for LSI. For example, we can start with a very simple function such as:

$$S(A,B) = \begin{cases} -.1 & \text{if number of connectivity paths is} > 1000 \\ 0 & \text{if number of connectivity paths is} < 100 \\ (.001 * \text{number of paths}), & \text{otherwise} \end{cases}$$

This function would need to evaluated and modified based on the results and the trend data for larger collections. Our goal is to approximate the LSI term-term matrix using a faster algorithm. This matrix can then be used in place of the LSI matrix in a variety of applications, such as our term clustering algorithm [13].

## 6. Acknowledgements

The authors gratefully acknowledge the assistance of Dr. Kae Kontostathis and Dr. Wei-Min Huang in developing the proof of the transitivity in the SVD as well as in reviewing drafts of this article. The authors also would like to express their gratitude to Dr. Michael Berry for his comments on an early draft. Co-author William M. Pottenger also gratefully acknowledges his Lord and Savior, Yeshua (Jesus) the Messiah, for His guidance in his life.

## 7. REFERENCES

[1] Berry, M., T. Do, G. O'Brien, V. Krishna, and S. Varadhan. "SVDPACKC (Version 1.0) User's Guide." April 1993.

[2] Berry, M., Z. Drmac and E.R. Jessup. "Matrices, Vector Spaces, and Information Retrieval." *SIAM Review 41:2*. pp. 335-362. 1999.

[3] Berry, M. W., Dumais, S. T., and O'Brien, G. W. "Using linear algebra for intelligent information retrieval." *SIAM Review, 37(4)*, 1995, 573-595. 1995.

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

[5] Dumais, S.T. "LSI meets TREC: A status report." *The First Text REtrieval Conference (TREC1)*, *D. Harman (Ed.), National Institute of Standards and Technology Special Publication* 500-207 , pp. 137-152. 1993.

[6] Dumais, S.T. "Latent Semantic Indexing (LSI) and TREC-2." In: D. Harman (Ed.), *The Second Text REtrieval Conference (TREC2), National Institute of Standards and Technology Special Publication 500-215*, pp. 105-116. 1994.

[7] Dumais, S.T., "Using LSI for information filtering: TREC-3 experiments." In: D. Harman (Ed.), *The Third Text REtrieval Conference (TREC3) National Institute of Standards and Technology Special Publications*. 1995.

[8] Ding, C.H.Q. "A similarity-based Probability Model for Latent Semantic Indexing." *Proc of 22nd ACM SIGIR '99 Conference. pp. 59-65.* 1999.

[9] Forsythe, G.E., M.A. Malcolm, and C.B. Moler. *Computer Methods for Mathematical Computations.* pp 192-235. 1977.

[10] Kontostathis, A. and W.M. Pottenger. "Improving Retrieval Performance with Positive and Negative Equivalence Classes of Terms." Lehigh University Technical Report, LU-CSE-02-009. 2002.

[11] Kontostathis, A. and W.M. Pottenger. "A Mathematical View of Latent Semantic Indexing: Tracing Term Co-occurrences." Lehigh University Technical Report, LU-CSE-02-006. 2002.

[12] Pottenger, W.M., Y.B. Kim and D.D. Meling. "HDDI™: Hierarchical Distributed Dynamic Indexing." In *Data Mining for Scientific and Engineering Applications.* 2001.

[13] Schütze, H. "Dimensions of Meaning." *In Proceedings of Supercomputing '92.* 1992.

[14] Story, R.E. "An explanation of the Effectiveness of Latent Semantic Indexing by means of a Bayesian Regression Model." *Information Processing and Management, 32(03)* pp. 329-344. 1996.

[15] Wiemer-Hastings, P. "How Latent is Latent Semantic Analysis?" *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, Aug 1999, pp. 932-937.

[16] Zelikovitz, S. and H. Hirsh. "Using LSI for Text Classification in the Presence of Background Text." *Proceedings of CIKM-01, 10th ACM International Conference on Information and Knowledge Management.* 2001

[17] Zha, H. "A Subspace-Based Model for Information Retrieval with Applications in Latent Semantic Indexing," Technical Report No. CSE-98-002, Department of Computer Science and Engineering, Pennsylvania State University, 1998.