Proceedings of the National Conference
On Undergraduate Research (NCUR) 2014
University of Kentucky, Lexington, KY
April 3-5, 2014

Semantic Similarity of Documents Using Latent Semantic Analysis

Chelsea Boling
Department of Mathematics
Lamar University
4400 MLK Boulevard
Beaumont, TX 77710

Faculty Advisor: Dr. Kumer Das

Abstract

Latent Semantic Analysis (LSA) is a technique that analyzes relationships between documents and its terms, and it discovers a data representation that has a lower dimension than the original semantic space. Essentially, the reduced dimensionality preserves the most crucial aspects of the data since LSA analyzes documents to find latent meaning in the corpus. The latent semantic space is determined by singular value decomposition (SVD), which enables a powerful process to simplify any rectangular matrix into a product of three unique components. The purpose of using SVD is to retrieve a sufficient amount of dimensions, which reveal a relevant structure that spans the original term-document matrix. In this study, LSA was used to find particular associations with user queries in a sample of documents from Medline Industries, Inc. Selecting an appropriate dimension for a reduced representation is suitable to represent the original latent space. The reduced model of the term-document matrix shows that SVD is capable of dealing with semantic problems. Overall, the goal is to overcome the problem of unsatisfactory indexed results by revealing hidden relationships among the terms and documents.

Keywords: Latent Semantic Analysis, Singular Value Decomposition, Text Mining

1 Introduction

The concept of LSA was patented in 1988 by a group of researchers at Bell Communications Research. The idea behind LSA was to overcome techniques that exclusively try to match search queries with the words of a document. Although this may seem adequate for the purpose of searching for relevant documents, the intuitive approach to search should be based on the conceptual content of the documents[2]. LSA attempts to overcome this problem by a statistical analysis of the latent structures of the documents. Thus, building a retrieval system, which reveals meaningful relationships, is the overall goal to overcome the problem of an incompetent search result. However, a common limitation of LSA is that there are cases in which words have multiple meanings, or polysemy. There are numerous methods to disambiguate the meaning of words by a Boolean algorithm, but it is computationally expensive to the retrieval system[2]. Although LSA is not successful with polysemy, the opposite holds true with synonymy, or words with the same meaning[5].

An effective retrieval model must be used to reveal the latent structures between the terms and documents. To achieve this goal, LSA applies singular value decomposition to the relationships of the terms and documents, which must be mathematically modeled by a matrix. LSA represents the text as an occurrence matrix, which means that each row represents a term, and each column corresponds to a document. The entries of the matrix are computed for its frequency in each document before LSA applies singular value decomposition to the text representation.

In this paper, the details of singular value decomposition are defined. The methodology of this study is discussed, such as preprocessing the data, applying SVD, and analyzing the results by using R, which is a programming language for statistical computing. Finally, the paper is concluded with an explanation of the results and the limitations of this project, as well as discussing future works on other pattern discovery techniques in text mining.

2 Singular Value Decomposition

Singular value decomposition plays an important statistical role in the fields of text mining and natural language processing. It is a method to factor a matrix by using linear algebra properties and concepts, such as matrix and vector computations, normal and orthonormal vectors, determinants, and orthogonality[2]. SVD is implemented by LSA in such a way that essentially reduces noise and preserves the most relevant attributes of a given set of data. In this study, the data is represented as a term-document matrix, in which the rows of the matrix represents the terms and the columns of the matrix represent the documents.

2.1 Fundamental Linear Algebra Concepts In SVD

Recall that the vector length of $\vec{x}$ can be computed by squaring each component of $\vec{x}$, adding the squared components, and taking the square root of the sum of the components[4]. This can be expressed as

$$|\vec{x}| = \sqrt{x_1 + x_2 + x_3 + ... + x_n} \quad (1)$$

Hence, the dot product of vectors $\vec{x}$ and $\vec{y}$ is basically multiplying vectors, where each component of $\vec{x}$ is multiplied by each component of $\vec{y}$ correspondingly[4]. This is expressed as

$$\vec{x} \cdot \vec{y} = \sum_{k=1}^{n} x_k y_k \quad (2)$$

If the dot product of two vectors is zero, then these two vectors are said to be orthogonal[4]. If a vector has a length of one, it is called a normal vector or a unit vector. Furthermore, if two vectors are normal and are orthogonal to each other, then the two vectors are orthonormal to each other. Additionally, an orthonormal basis is an orthonormal set of vectors.

The preceding concepts can be proven with further details, but what is more important is that these concepts are very useful for what is called the Gram-Schmidt method. The Gram-Schmidt method takes a given basis and converts it into a corresponding orthonormal basis[4]. Essentially, the Gram-Schmidt method is simply projecting one vector onto another vector and subtracting off the projeciont so that there is a perpendicular difference. The following is an algorithm, which will compute the corresponding orthonormal set of vectors from a given basis $B = \{b_1, b_2, ..., b_n\}$ of $\hat{A}^n$: First, normalize the first vector. This becomes the first vector v_1 in the orthonormal basis. Depending on how many vectors are in the original subspace, the next step is to find v_2 until v_n. Generally, this can be written as

$$v_i = b_i - \sum_{i=1}^{i-1} proj_{v_j} b_i \text{ where } proj_{v_j} = \frac{b_i \cdot v_j}{\|v_j\|^2} \quad (3)$$

It is a repetitive process of writing vectors and removing normalized vectors in each step until solving for the last vector of the original basis.

Recall that the identity matrix is a square matrix in which the diagonal entries are one while the other entries are zero[4]. Therefore, any matrix multiplied by the identity is itself. In addition, multiplying matrices can only be done if

the columns of the first matrix are the same as the rows of the second matrix. In other words, if matrix A is a $m \quad n$ matrix, and B is a $n \quad w$ matrix, then the matrix AB is a $m \quad w$ matrix.

Similar to the identity matrix, a diagonal matrix is a matrix in which nonzero values make a diagonal. To transpose a matrix, convert the matrix rows into its columns; this is crucial to test whether a matrix is orthogonal. Moreover, a matrix B is orthogonal if $BB^T = B^T B = I$.

The determinant of a square matrix is simply a value that provides information about what can be done to a matrix[4]. The determinant of a 2×2 matrix A is defined as follows,

$$|A| = \begin{vmatrix} a & b \\ c & d \end{vmatrix} = ad - bc \tag{4}$$

In other cases, in which matrices are larger than a 2×2 matrix, several properties of determinants back up the notion of a method called cofactor expansion along the i^{th} row of A. The goal is to repetitively "delete" rows and columns in order to create smaller matrices (e.g. 2×2 matrices) for easier computation. Cofactoring by row or column is preferential, just as long as each smaller matrix is multiplied by it.

Fundamentally, the nature of eigenvalues gives special information about a vector, such as if the vector is stretched, compressed, or reversed in direction. Therefore, an eigenvector is a nonzero vector $\vec{x}$, which satisfies $A\vec{x} = \lambda\vec{x}$ where A is a square matrix and λ is an eigenvalue. In order to solve for the eigenvalues and eigenvectors, the matrix is utilized as a system of linear equations, which can be solved for the components of the eigenvector. The eigenvalues have to satisfy the characteristic equation of matrix A for which $\det(A - \lambda I) = 0$, where I is the identity matrix.

Generally the number of eigenvalues depends on the number of elements in the matrix. By solving for the eigenvalues and coming to a linear relationship between variables (via Gaussian elimination and other techniques), there can be a case in which a variable may have an infinite number of values which satisfy the equation; however, not every component of an eigenvector can be zero. In essence, having more linearly independent eigenvectors makes understanding linear transformations a bit easier.

2.2 Singular Value Decomposition Model

The following algorithm computes the SVD of a matrix $A_{m \times n}$:

$$A_{m \times n} = U_{m \times m} \Sigma_{m \times n} V^T_{n \times n} \tag{5}$$

such that $U^T U = I$ and $V^T V = I$. This means that U is an orthogonal matrix in which the columns are the orthonormal eigenvectors (left singular vectors) of AA^T; Σ is a diagonal matrix in which the nonnegative diagonal elements are singular values of A, and V^T is the transpose of V, an orthogonal matrix in which the rows are the orthonormal eigenvectors (right singular vectors) of $A^T A$. Note that the singular values of Σ are presented as the square roots of the eigenvalues of AA^T and $A^T A$, and these singular values are arranged in descending order.

2.3 Computing The SVD Of A Matrix

Moreover, calculating the SVD of a matrix is a series of steps, which can be explained in this algorithm:

- To compute U, find the eigenvalues and eigenvectors of AA^T; this requires finding the transpose of A and computing AA^T.

- The equation $A\vec{x} = \lambda\vec{x}$ can be set up for AA^T and can be transformed into a set of linear equations, which is arranged as a coefficient matrix. Next, solve for λ by setting the determinant equal to zero. From here, the eigenvalues can be computed. This step allows us to plug in each value of λ into the original equations to solve for the eigenvectors. The eigenvectors are arranged in such a way that the eigenvector associated with the largest eigenvalue is placed in the first column vector.
- Then, the following matrix is converted into an orthogonal matrix by using the Gram-Schmidt method. Since matrix V is orthogonal like matrix U, it can be calculated the same way for A^TA; however, in the end, V is transposed.
- To construct the S matrix, take the square root of each eigenvalue and put the largest value at the beginning of diagonal of S. These are the singular values of A. The eigenvalues of U and V should be the same.

3 Methodology

The purpose of using SVD in text mining is to retrieve a sufficient amount of dimensions, which reveal a relevant structure that spans the original term-document matrix. Several experimental results have shown that selecting an appropriate dimension for a reduced representation maximizes the performance of the retrieval system[1,2]. Therefore, in a reduced model of the term-document matrix, the overall expectation is to retrieve a relevant structure, and this representation proves that SVD is capable of dealing with semantic problems.

Certainly from the literature review and other queries pertaining to text mining, other research questions are raised in this study:

- What would be the maximum dimension cutoff for a semantic space to be represented adequately?
- Following the previous question, do smaller representations perform better, or are these experimental datasets unrealistically too good to be true based on how they are constructed?
- LSA is not based on previous knowledge or dictionaries. Can LSA's performance with polysemy be enhanced using dictionaries or prior knowledge?

3.1 Creating A Corpus Of Documents

Since there is an unprecedented amount of digital information that is generated every day, it is only fair to gain insight from these records, such as medical articles, through text mining. Medical articles make an interesting dataset because one could discover unsuspected links from the vast range of literature; these unsuspected links could mean new information about a certain disease or cancer. For this study, the dataset was collected from Medline Industries, Inc., which is a private manufacturer and distributor of healthcare supplies in the United States[7]. The preliminary dataset for the experiment consisted of ten PDF documents from Medline Industries, Inc. on catheter-associated urinary tract infections, hand hygiene, ventilator-associated pneumonia, forced-air warming, and hospital-acquired infections. Moreover, for each document that was obtained, the title and abstract were extracted and converted into text files. These are the titles from each text file:

- *d1*: "Variations in risk perceptions: a qualitative study of why unnecessary urinary catheter use continues to be problematic", by Molly Harrod, Christine P. Kowalski, Sanjay Saint, Jane Forman, and Sarah L. Krein.
- *d2*: "Changing Clinical Behaviors to Lower Costs and Reduce Catheter-Associated Urinary Tract Infections (CAUTI), ARKANSAS METHODIST MEDICAL CENTER: How a foley catheter management system combined with education helped reduce catheter utilization by 21 percent", by Lisa Bridges, RN, Infection Preventionist, Arkansas Methodist Medical Center.
- *d3*: "Population kinetics of the skin flora under the glove following surgical hand disinfection with three propanol-based hand rubs a prospective, randomized, double-blind trial", by Manfred L. Rotter, Gunter Kampf, Miranda Suchomel, Michael Kundi.
- *d4* : "Effect of topical treatments on irritant hand dermatitis in health care workers", by Marty Visscher, PhD, Jennifer Davis, BS, and Randy Wickett, PhD.
- *d5*: "CHLORHEXIDINE,TOOTHBRUSHING, AND PREVENTING VENTILATOR ASSOCIATED

PNEUMONIA IN CRITICALLY ILL ADULTS", by Cindy L. Munro, RN, PhD, ANP, Mary Jo Grap, RN, PhD, ACNP, Deborah J. Jones, RN, PhD, Donna K. McClish, PhD, and Curtis N. Sessler, MD.

- *d6* : "Preventing Ventilator-Associated Pneumonia in the United States: A Multicenter Mixed- Methods Study", by Sarah L. Krein, PhD, RN; Christine P. Kowalski, MPH; Laura Damschroder, MS, MPH; Jane Forman, ScD, MHS; Samuel R. Kaufman, MA; Sanjay Saint, MD, MPH.
- *d7*: "Effectiveness of an educational program to reduce ventilator-associated pneumonia in a tertiary care center in Thailand: a 4-year study", by Apisarnthanarak A1, Pinitchai U, Thongphubeth K, Yuekyen C, Warren DK, Zack JE, Warachan B, Fraser VJ.
- *d8*: "Oral decontamination with chlorhexidine reduces the incidence of ventilator-associated pneumonia", by Koeman M1, van der Ven AJ, Hak E, Joore HC, Kaasjager K, de Smet AG, Ramsay G, Dormans TP, Aarts LP, de Bel EE, Hustinx WN, van der Tweel I, Hoepelman AM, Bonten MJ.
- *d9*: "Double gloving to reduce surgical cross-infection", by Tanner J, Parkinson H.
- *d10*: "Forced-air warming blowers: An evaluation of filtration adequacy and airborne contamination emissions in the operating room", by Albrecht M, Gauthier RL, Belani K, Litchy M, Leaper D.

This collection of text, or *corpus*, was converted into a term-document matrix using R. Initially, the corpus was represented as a 338 × 10 matrix, which consisted of noise. In order to greatly expose the semantic relationships of the corpus, this noise has to be reduced before applying the SVD to the term-document matrix. Throughout this study, LSA discovers the relationships of these documents, as well as the terms within these documents.

3.2 Pre-Processing The Corpus

In text mining, preprocessing the corpus is the most important step. Pre-processing is removing noise in the dataset by stemming, removing stop words, and anything else that needs to be removed, such as the header of a document. In other cases, preprocessing the data has to be done in consideration of any sentence overlap and other semantic issues. Pre-processing methods are defined as follows:

- *Stemming* refers to reducing words to the word's root form.
- *Removing stop words* refers to filtering the corpus of any common words. This reduces indexing, as stop words are not useful for text mining. Examples of English stop words are "as", "the", "which", and "at".
- A customized list of stop words includes words that are not relevant to the conceptual meaning of the corpus. This customized list of stop words may be developed through basic coding in R.

A new term-document frequency matrix *A* was created using R's LSA package. Headers, English stop words, and a list of words that did not serve a significant meaning to the overall term–by-document matrix were removed from the dataset. For instance, terms in the initial corpus, such as "hospitalized", "airborne", and "utilization", are not necessary for LSA to deduce the semantic similarities efficiently, as the frequency of these words creates a sparse matrix, or a matrix with many zero entries.

Table 1. sample of the term-document frequency matrix A

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
care	4	0	0	2	1	0	1	0	0	0
cauti	2	2	0	0	0	0	0	0	0	0
infection	1	0	0	1	0	0	1	1	0	1
infections	1	1	0	0	0	0	0	0	2	0
prevention	1	1	0	0	0	0	0	0	0	0
pneumonia	0	0	0	0	2	1	1	1	0	0
toothbrushing	0	0	0	0	2	0	0	0	0	0
hygiene	0	0	1	1	0	0	0	0	0	0
contamination	0	0	0	0	0	0	0	0	1	2
hypothermia	0	0	0	0	0	0	0	0	0	1

There are 53 terms in the matrix, and Table 1 shows the frequency of a sample of these terms. Each element of the matrix is a count of the number of times that term appears in the document.

3.3 Performing SVD To The Corpus

By applying LSA to the corpus in R, matrix *A* is factorized into its three unique components: U, S, and V^T.

Table 2. matrix U of the term-document matrix A

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
care	-0.5501	-0.1083	0.3104	-0.0922	0.3169	-0.1074	0.1267	-0.0272	0.1264	-0.3011
catheter	-0.1759	-0.1944	-0.0626	-0.1709	-0.2506	0.0403	-0.1236	0.0239	-0.0088	0.0125
cauti	-0.2689	-0.2563	-0.0318	-0.2345	-0.1819	0.0338	-0.0071	0.0171	0.1037	-0.0093
health	-0.2374	-0.1176	0.1681	-0.0314	0.1623	0.0286	0.0791	-0.0418	0.17	0.0161
infection	-0.2163	0.1063	0.0844	-0.0157	0.1073	0.2544	-0.0407	0.2712	0.0122	-0.1649
infections	-0.2024	-0.1874	-0.2702	0.1368	0.0209	0.0493	0.0208	-0.071	-0.0253	-0.0223
patients	-0.2806	-0.146	-0.4182	0.27	0.0536	-0.1447	-0.0145	-0.0727	-0.0117	-0.0382
practices	-0.2226	-0.0664	0.0454	-0.1948	0.1603	-0.0204	0.3819	-0.0384	-0.3175	0.6643
prevention	-0.1344	-0.1281	-0.0159	-0.1173	-0.0909	0.0169	-0.0035	0.0085	0.0519	-0.0046
tract	-0.1344	-0.1281	-0.0159	-0.1173	-0.0909	0.0169	-0.0035	0.0085	0.0519	-0.0046
urinary	-0.1344	-0.1281	-0.0159	-0.1173	-0.0909	0.0169	-0.0035	0.0085	0.0519	-0.0046
aseptic	-0.0415	-0.0662	-0.0467	-0.0536	-0.1597	0.0234	-0.1201	0.0153	-0.0607	0.0172
behavioral	-0.0415	-0.0662	-0.0467	-0.0536	-0.1597	0.0234	-0.1201	0.0153	-0.0607	0.0172
catheterization	-0.0415	-0.0662	-0.0467	-0.0536	-0.1597	0.0234	-0.1201	0.0153	-0.0607	0.0172
catheters	-0.0415	-0.0662	-0.0467	-0.0536	-0.1597	0.0234	-0.1201	0.0153	-0.0607	0.0172
cdc	-0.0546	-0.0536	-0.0037	0.0411	-0.3043	0.0306	0.0626	0.0297	-0.0536	-0.0168
clinical	-0.083	-0.1325	-0.0935	-0.1072	-0.3194	0.0468	-0.2402	0.0307	-0.1213	0.0343
disease	-0.0415	-0.0662	-0.0467	-0.0536	-0.1597	0.0234	-0.1201	0.0153	-0.0607	0.0172
alcohol	-0.0131	0.0126	0.043	0.0947	-0.1447	0.0073	0.1827	0.0144	0.0071	-0.0339
antimicrobial	-0.0131	0.0126	0.043	0.0947	-0.1447	0.0073	0.1827	0.0144	0.0071	-0.0339
antisepsis	-0.0131	0.0126	0.043	0.0947	-0.1447	0.0073	0.1827	0.0144	0.0071	-0.0339
blind	-0.0131	0.0126	0.043	0.0947	-0.1447	0.0073	0.1827	0.0144	0.0071	-0.0339
chlorhexidine	-0.1876	0.2654	0.0681	0.1738	-0.1909	-0.2404	-0.0795	0.0187	0.2566	0.0887
gluconate	-0.0646	0.0188	0.1494	0.1906	-0.1198	0.0488	0.0287	-0.0138	-0.048	0.0258
hand	-0.0908	0.044	0.2354	0.3799	-0.4091	0.0633	0.3941	0.0151	-0.0338	-0.042
hygiene	-0.0646	0.0188	0.1494	0.1906	-0.1198	0.0488	0.0287	-0.0138	-0.048	0.0258
bacterial	-0.0515	0.0062	0.1064	0.0959	0.0249	0.0416	-0.1541	-0.0282	-0.0551	0.0597
creams	-0.1029	0.0124	0.2128	0.1918	0.0497	0.0831	-0.3081	-0.0564	-0.1102	0.1195
dermatitis	-0.0515	0.0062	0.1064	0.0959	0.0249	0.0416	-0.1541	-0.0282	-0.0551	0.0597
icd	-0.1029	0.0124	0.2128	0.1918	0.0497	0.0831	-0.3081	-0.0564	-0.1102	0.1195
irritation	-0.0515	0.0062	0.1064	0.0959	0.0249	0.0416	-0.1541	-0.0282	-0.0551	0.0597
lotions	-0.1029	0.0124	0.2128	0.1918	0.0497	0.0831	-0.3081	-0.0564	-0.1102	0.1195
morbidity	-0.1099	0.2313	-0.0655	-0.0583	-0.0325	-0.0334	-0.0446	0.0128	-0.0133	-0.2564
mortality	-0.0788	0.1755	-0.0605	-0.0229	-0.0479	-0.079	-0.0607	-0.0055	0.2524	0.0699
pharmacological	-0.0442	0.0711	-0.0208	0.0062	-0.0232	-0.2101	-0.0475	0.038	0.0522	-0.0071
pneumonia	-0.1725	0.3311	-0.0945	-0.0859	-0.0443	-0.2472	-0.0177	0.0383	-0.2324	0.0905
toothbrushing	-0.0885	0.1421	-0.0416	0.0123	-0.0465	-0.4203	-0.095	0.076	0.1044	-0.0141
ventilation	-0.0442	0.0711	-0.0208	0.0062	-0.0232	-0.2101	-0.0475	0.038	0.0522	-0.0071
ventilator	-0.1725	0.3311	-0.0945	-0.0859	-0.0443	-0.2472	-0.0177	0.0383	-0.2324	0.0905
vap	-0.1841	0.4537	-0.1373	-0.1918	-0.0316	0.4809	0.067	-0.1065	-0.2022	-0.0677
death	-0.0311	0.0558	-0.005	-0.0355	0.0154	0.0456	0.0161	0.0183	-0.2657	-0.3263
icus	-0.0311	0.0558	-0.005	-0.0355	0.0154	0.0456	0.0161	0.0183	-0.2657	-0.3263
antibiotic	-0.0345	0.1045	-0.0397	-0.029	-0.0246	0.1311	-0.0132	-0.0436	0.2002	0.077
colonization	-0.0345	0.1045	-0.0397	-0.029	-0.0246	0.1311	-0.0132	-0.0436	0.2002	0.077
decontamination	-0.0691	0.2089	-0.0794	-0.058	-0.0492	0.2622	-0.0265	-0.0871	0.4004	0.154
endotracheal	-0.0345	0.1045	-0.0397	-0.029	-0.0246	0.1311	-0.0132	-0.0436	0.2002	0.077
pathogens	-0.1024	0.0452	-0.294	0.225	0.0873	0.1635	0.0111	-0.1231	0.123	0.0593
blood	-0.1019	-0.0889	-0.3815	0.3811	0.1678	0.0485	0.0366	-0.1193	-0.1158	-0.0265
contamination	-0.0466	-0.0261	-0.1434	0.1603	0.1018	0.1014	0.0001	0.6232	0.002	0.0842
airborne	-0.0063	0.0018	-0.0081	0.0166	0.0229	0.0426	-0.0061	0.3315	0.0203	0.0465
buildup	-0.0063	0.0018	-0.0081	0.0166	0.0229	0.0426	-0.0061	0.3315	0.0203	0.0465
contaminants	-0.0063	0.0018	-0.0081	0.0166	0.0229	0.0426	-0.0061	0.3315	0.0203	0.0465
hypothermia	-0.0063	0.0018	-0.0081	0.0166	0.0229	0.0426	-0.0061	0.3315	0.0203	0.0465

Table 2 shows matrix U, in which U is a 53 × 10 matrix with orthonormal columns. Table 3 shows matrix V, in which V is a 10 × 10 matrix with orthonormal columns. In its transposed form, V^T, the rows are orthonormal.

$$\begin{pmatrix} 7.2877 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 5.8524 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 5.3729 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4.7260 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4.1910 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3.8378 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3.2698 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.9289 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.1908 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2.0184 \end{pmatrix}$$

Figure 1: Singular Values of Matrix S

The diagonal matrix S consists of these singular values in descending order, which is denoted in Figure 1.

Table 3: matrix V of the latent semantic space

	d1	d2	d3	d4	d5	d6	d7	d8	d9	d10
d1	-0.6774	-0.3622	0.1658	-0.3009	0.2881	-0.0249	0.3811	-0.0199	0.2465	-0.0440
d2	-0.3024	-0.3877	-0.2511	-0.2533	-0.6692	0.0898	-0.3927	0.0449	-0.1329	0.0346
d3	-0.0955	0.0738	0.2310	0.4474	-0.6062	0.0278	0.5974	0.0422	0.0155	-0.0685
d4	-0.3751	0.0362	0.5716	0.4532	0.1042	0.1595	-0.5037	-0.0826	-0.1207	0.1206
d5	-0.3224	0.4159	-0.1118	0.0291	-0.0974	-0.8065	-0.1552	0.1113	0.1144	-0.0143
d6	-0.1337	0.1680	-0.0438	-0.1594	0.0478	-0.0142	0.2433	-0.0364	-0.5944	0.7144
d7	-0.2266	0.3264	-0.0269	-0.1677	0.0646	0.1752	0.0527	0.0536	-0.5822	-0.6587
d8	-0.2517	0.6114	-0.2134	-0.1372	-0.1032	0.5031	-0.0432	-0.1276	0.4386	0.1554
d9	-0.2475	-0.1735	-0.6832	0.6004	0.2344	0.0621	0.0398	-0.1165	-0.0846	-0.0178
d10	-0.0459	0.0105	-0.0438	0.0786	0.0960	0.1636	-0.0198	0.9708	0.0445	0.0939

Table 3 shows the values of matrix *V*, where swapping the rows with the columns and vice-versa forms V^T.

3.4 Approximation Of Matrix A With *K* Singular Values

k dimensions of matrix A were retained by computing the energy in S. In order to retain 90% of the energy in S, E_k was computed and divided by the total energy, which is when *k* = 10. E_k is defined as

$$E_k = \sum_{i=1}^{k} \sigma_{ii}^2 \tag{6}$$

where *k* denotes the number of reduced dimensions and σ_{ii} represents the singular values of S.

Table 4: energies in S

k = 1 0.2668867	*k* = 2 0.4390005	*k* = 3 0.5840666	*k* = 4 0.6963009	*k* = 5 0.7845641
k = 6 0.8585774	*k* = 7 0.9123036	*k* = 8 0.9554107	*k* = 9 0.9795286	*k* = 10 1.0000000

In Table 4, *k* values from 1 through 7 yield a retained energy of 91.2%.

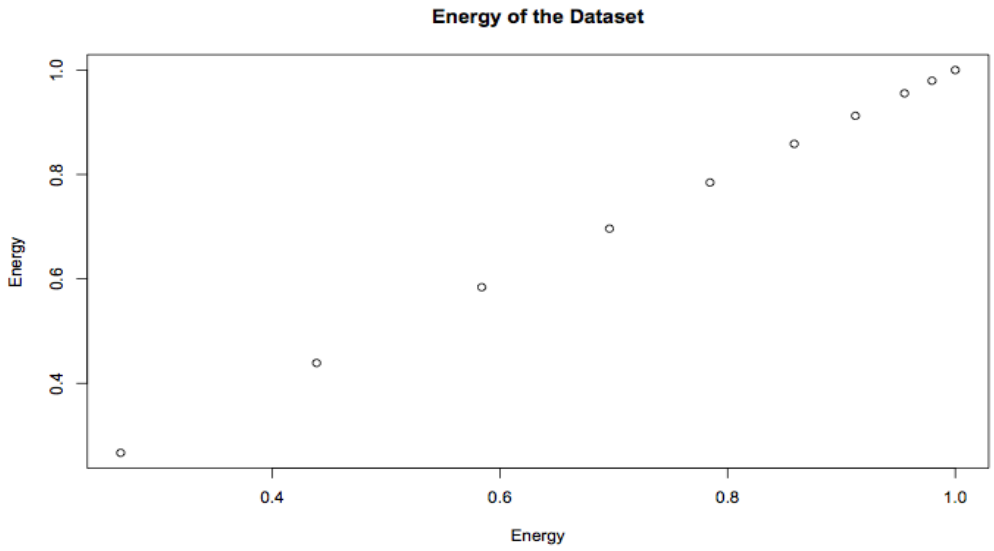

Figure 2: Energies of the Term-document Matrix

The energies are graphed in Figure 2 by using Table 4. Furthermore, reducing the dimensions of S at $k = 7$ is sufficient to compute the reduced SVD of matrix *A*.

3.5 Close Associations Within The Terms

The close associations of the terms are measured by computing the cosine similarity. The cosine similarity of two terms measures the similarity of these terms on a scale of [0, 1]. This is defined as

$$\cos(q) = \frac{a \times b}{\|a\|\|b\|} \tag{7}$$

where a and b are vectors, which represent the terms, and $\|a\|$ and $\|b\|$ are the magnitudes of a and b respectively.

Generally if the value of q is zero or close to zero, this means that the angle between the two terms is not close. In R, the *associate()* function of the LSA package sorts the closeness values in descending order. For instance, the third document is about hand hygiene, so if the cosine measure of one of the words in the third document were computed, such as "hygiene", then these are the close associations for the word "hygiene":

Table 5. closeness values of "hygiene" when $k = 10$

gluconate 1.0000000	hand 0.8944272	alcohol 0.7071068	antimicrobial 0.7071068
antisepsis 0.7071068	blind 0.7071068	bacterial 0.7071068	creams 0.7071068
dermatitis 0.7071068	icd 0.7071068	irritation 0.7071068	lotions 0.7071068

These associations were computed from the full SVD. The first 6 terms are in the third document, so it makes sense that the measures of these terms were first. Interestingly, "creams", "lotions", "dermatitis", and "ICD", which

stands for Irritant Contact Dermatitis, are close associations. These terms come from the fourth document, which is also about hand hygiene. Although the term-document frequency matrix looks sparse and comes from such a diverse range of medical topics, computing SVD puts terms similar to each other in the semantic space.

Table 6. closeness values of "hygiene" when $k = 7$

gluconate 1.0000000	hand 0.8978206	alcohol 0.7143657	antimicrobial 0.7143657
antisepsis 0.7143657	blind 0.7143657	creams 0.7075397	icd 0.7075397
lotions 0.7075397	bacterial 0.7041418	dermatitis 0.7041418	irritation 0.7041418

When the dimensions were reduced to k = 7, then the closeness values are nearly the same as in Table 5. In other words, the terms that are the closest to "hygiene" are included, but the measures are slightly different.

4 Summary of Results

LSA is essentially a statistical method for extracting relationships into word passages. It is interesting to note that LSA uniquely takes in a set of strings and separates this sample into a representation of relevant selections of text. By observing the closeness values, LSA is able to deduce the conceptual meaning of a term and associate the meaning with other terms. The term "gluconate" has a closeness value of 1.0, which indicates that "hygiene" and "gluconate" are used closely in the dataset. Trivially, these terms do appear in the same document, which discusses about gluconate soap and hand hygiene. Other terms with a closeness value of .70 and below appeared in documents not related to hand hygiene, and the relationships were not significant. The cosine similarity was measured for many terms, but "hygiene" is an interesting choice because it is among the terms that had the lowest frequency.

Furthermore, there is a clear difference between the frequency of the terms and the hidden relationships between the terms. Generally, LSA is limited in the sense that it cannot handle polysemy effectively[5]. However, the corpus in this study did not have terms that have the same meaning, so polysemy was not an issue. The documents and terms are mapped to a single semantic space, which is useful for clustering either documents or terms to retrieve how the documents or terms correspond to each other.

5 Discussion and Conclusion

The results of this research are in its preliminary stage, and testing a larger dataset, such as from the PubMed Central Open Access Subset, would show significant results about term similarities and document similarities. For future studies, the use of OCR software must be used to consider documents that are not in an editable text form. In addition, noise from the documents has to be strictly removed for a better analysis. Terms, such as "blind" and "buildup", did not have a significant effect on the other terms, and if these terms were removed, the computations would be slightly better. In this study, there was not an issue with polysemy as the dataset was small enough to observe the noise; however, polysemy in the literature for text mining has dealt with this issue from bigger datasets. More importantly, term-frequency-inverse-document-frequency (TF-IDF) weighting on the matrix may also be considered in future works so that uncommon words would have a higher weighting than common words throughout the entire dataset[9]. Although there are many applications and techniques that enhance LSA, testing other known text mining techniques, such as non-negative matrix factorization, would make a great comparison to LSA.

6 Acknowledgements

This work has been partially supported by the Office for Undergraduate Research of Lamar University.

7 References

1. Berry, M. W., S.T. Dumais, and G.W. O'Brien, "Using linear algebra for intelligent information retrieval," SIAM Review 37(4) (1995): 573–595.

2. 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) (1990): 391–407.

3. Dumais, S. T., G. W. Furnas, T. K. Landauer, S. Deerwester, and R. Harshman, "Using latent semantic analysis to improve access to textual information," CHI 88:Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, ACM Press 25 (23) (1988): 281–285.

4. Hill, Richard. "Elementary Linear Algebra with Applications" (Saunders College Pub., 1996), Third Edition.

5. Landauer, T.K., D. Laham, B. Rehder, and M.E. Schreiner, "How Well Can Passage Meaning Be Derived without Using Word Order? A Comparison of Latent Semantic Analysis and Humans," Proc. 19th Ann. Meeting of the Cognitive Science Soc. (1997): 412-417.

6. Landauer, T. K., P. W. Foltz, and D. Laham, "Introduction to Latent Semantic Analysis," Discourse Processes 25(23) (1998): 259–284.

7. Medline Research Library (2014). Medline Industries, Inc. URL http://www.medline.com/research/library/

8. R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

9. Ramos, Juan. "Using tf-idf to determine word relevance in document queries." Proceedings of the First Instructional Conference on Machine Learning. 2003.