Choosing LSI Dimensions by Document Linear Association Analysis

Chenggen Shi and Jie Lu
Faculty of Information Technology
University of Technology
Sydney, Australia
cshi@it.uts.edu.au and jielu@it.uts.edu.au

Abstract— Latent Semantic Indexing (LSI) has proven to be a valuable analysis tool with a wide range of applications, however the crucial question, choosing an appropriate number of dimensions for LSI, is still unsolved. In this paper, a new method which is to deal with this problem is described. It finds that a sum of total dot products between all document vectors reaches the maximum value at a specific number of dimensions for a given dataset. With this reduced dimensions LSI achieves the best performance. The performance evaluations have demonstrated that this method can choose an appropriate number of dimensions for LSI and effective detect the data structure for a dataset.

Index Terms— linear association analysis, latent semantic indexing, information retrieval, web mining

1. Introduction

As digital libraries and the World-Wide-Web (WWW) continue to proliferate the enormous volume of online textual material, developing intelligent information retrieval technology becomes one of the great challenges in the information sciences

Latent Semantic Indexing (LSI) [1] is an approach to automatic indexing and information retrieval by mapping documents as well as words to a representation in the so-called latent semantic space. The general claim is that similarities between documents or between documents and queries can be more reliably estimated in the reduced latent space representation than in the original representation. In many applications [1, 3, 4, 5, 8] this has proven to result in more robust word processing.

Although LSI has been applied with remarkable success in different domains [9, 11, 12, 13, 14, 15], its theoretical foundation remains to a large extent unsatisfactory and incomplete. The principle challenge is how to choose an appropriate number of dimensions for the LSI [1]. One can intuitively say that the appropriate number of dimensions is large enough to fit all the real structure in the data, but small enough so that we do not also fit the sampling error or unimportant details. If too many dimensions are used, the method begins to approximate standard vector methods and loses its power to represent the similarity between words. If too few dimensions are used, there is not enough discrimination among similar words and documents or it will create a serious distortion of word to word and document to document similarity.

The primary goal of this paper is to present a novel approach to choose an appropriate number of dimensions for the LSI given a dataset. Our experimental results show that the proposed approach is effective and promising. The rest of this paper is organized as follows. Section 2 overviews LSI technologies and presents document vector linear association model. Section 3 describes experimental results and observation about document vector linear association and the number of dimensions. Section 4 addresses performance evaluations of the proposed model. The conclusions are summarized in the last section 5.

2. Background and Document Vector Linear Association Model

2.1 Overview of Latent Semantic Indexing Technology

The key idea of LSI is to map word-document to a vector space of reduced dimensionality, the latent semantic space [1]. The mapping is restricted to be linear and based on a Singular Value Decomposition (SVD) of a co-occurrence table. The singular value decomposition [6][10] is commonly used in the solution of unconstrained linear least squares problems, matrix rank estimation, and canonical correlation analysis. Given an $M \times N$ matrix $A = (a_{mn})_{M \times N}$, where $M \geq N$, the SVD of matrix A is given [1]

$$A = U \sum V^T \qquad (1)$$

Where U is an $M \times N$ orthogonal matrix and $U^T U = I$; V is an $N \times N$ orthogonal matrix and $V^T V = I$; $\sum$ is an $N \times N$ diagonal matrix with

positive or zero elements (the singular values). For $h = 1, 2, \ldots, N$, let σ_h be a singular value of A, The singular values are ordered, so that

$$\sigma_1 \geq \sigma_2 \ldots \geq \sigma_N \quad (2)$$

The LSA approximation of the matrix A is computed by setting all but the largest K singular values in Σ to zero (= $\tilde{\Sigma}$).

$$\tilde{A} = U\tilde{\Sigma}V^T \quad (3)$$

Notice that the document-to-document dot products based on this approximation are given by $\tilde{A}^T\tilde{A} = V\tilde{\Sigma}^2V^T$.

While the original high-dimensional vectors are sparse, the corresponding low-dimensional latent vectors will typically not be sparse. This implies that it is possible to compute meaningful association values between pairs of documents, even if the documents do not have any words in common.

2.2 Document Vector Linear Association Model

A dot product Φ_{ij} ($i, j = 1, 2, \ldots, N$, $i \neq j$)between any two document vectors of the matrix A is defined as

$$\Phi_{ij} = \sum_{m=1}^{M} a_{mi}a_{mj} \quad (4)$$

A sum of dot products Φ_i between any one document vector and all other document vectors of the matrix A is defined as

$$\Phi_i = \sum_{j=1}^{N} \Phi_{ij} \quad (5)$$

A total sum of dot products Φ between the all document vectors of the matrix A is defined as

$$\Phi = \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Phi_{ij} \quad (6)$$

and called document vector linear association.

In general, for $A = U\Sigma V^T$ the matrices U, Σ and V must be of full rank. The beauty of an SVD, however, is that it allows a simple strategy for optimal approximate fit using smaller matrices. According to (2) the singular values in Σ are ordered by size, so the first K largest may be kept and the remaining smaller ones set to zero (= $\tilde{\Sigma}$). Since zeros were introduced into $\tilde{\Sigma}$, the representation can be simplified by deleting the zero rows and columns of $\tilde{\Sigma}$, to obtain a new diagonal matrix Σ_K, and then deleting the corresponding columns of U and V to obtain U_K and V_K respectively. (Actually, we approximate the matrix A keeping only the first K largest singular values and the corresponding columns from the U and V.) The result is described into a reduced model: A_K, an $M \times N$ matrix which is constructed from the K-largest singular triplets of the matrix A,

$$A_K = U_K \Sigma_K V_K^T \quad (7)$$

where U_K an $M \times K$ matrix, is the first K columns of the U; V_K, an $N \times K$ matrix, is the first K columns of the V; for Σ_K, an $K \times K$ matrix, we keep only the first K singular values from Σ. For any $A = (a_{mn})_{M \times N}$ and $0 < K < N < M$, there is a $\Phi = \Phi(A)$ and $A_K = \psi(A, K)$, therefore, $\Phi(A_K) = \Phi(\psi(A, K))$. When A is given, for any $0 < K < N < M$, there is a unique value Φ.

In this paper, we try to find an effective way to choose an appropriate number of dimensions for LSI by exploring the relationship between K and Φ, particularly testing the K value when Φ achieves the maximum value.

3. Experiments and Observation

3.1 Experiment 1

Table 1 shows dataset 1 collected by Deerwester, et al [1].

Table 1 Dataset 1

Documents (Titles):

D1: Human machine interface for Lab ABC computer applications
D2: A survey of user opinion of computer system response time
D3: The EPS user interface management system
D4: System and human system engineering testing of EPS
D5: Relation of user-perceived response time to error measurement
D6: The generation of random, binary, unordered trees
D7: The intersection graph of paths in trees
D8: Graph minors IV: Widths of trees and well-quasi-ordering
D9: Graph minors: A survey

Dataset 1 consists of the titles of 9 Bellcore technical memoranda. Words occurring in more than one document and not on a stop list of 439 common words used by SMART [17] are selected for indexing. They are underlined. Corresponding to the text in Table1 is the word-document matrix $A = (a_{mn})_{12\times9}$ shown in Table 2.

Table 2 The word-document matrix $A = (a_{mn})_{12\times9}$ for dataset 1

Words	Documents								
	D1	D2	D3	D4	D5	D6	D7	D8	D9
human	1	0	0	1	0	0	0	0	0
interface	1	0	1	0	0	0	0	0	0
computer	1	1	0	0	0	0	0	0	0
user	0	1	1	0	1	0	0	0	0
system	0	1	1	2	0	0	0	0	0
response	0	1	0	0	1	0	0	0	0
time	0	1	0	0	1	0	0	0	0
EPS	0	0	1	1	0	0	0	0	0
survey	0	1	0	0	0	0	0	0	1
trees	0	0	0	0	0	1	1	1	0
graph	0	0	0	0	0	0	1	1	1
minors	0	0	0	0	0	0	0	1	1

The elements of this matrix are the frequencies in which a word occurs in a document. For example, in document D2, the second column of the word-document matrix, `survey`, `user`, `computer`, `system`, `response`, and `time` all occur once. Through running the proposed $\Phi = \Phi(\psi(A, K))$ for the data in Table 2, we find that the value of Φ goes up as K increases at the beginning; when the value of K is equal to 2, Φ reaches the maximum value of 27.3. The value of Φ then goes down while K increases continuously (Figure 1).

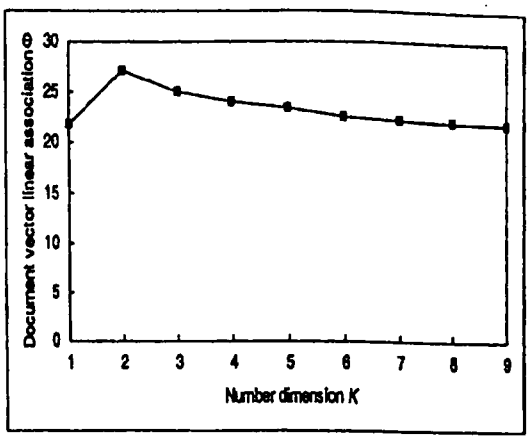

Figure 1 Document vector linear association Φ vs. number dimension K for dataset 1

3.2 Experiment 2

Table 2 shows dataset 2 collected by Berry, et al [16].

Table 3 Dataset 2

Documents (Titles)

D1: Automatic Differentiation of Algorithms: Theory, Implementation and Application
D2: Geometrical Aspects of Partial Differential Equations
D3: Ideals, Varieties, and Algorithms -- An Introduction to Computational Algebraic Geometry and Commutative Algebra
D4: Introduction to Hamiltonian Dynamical Systems and the N-Body Problem
D5: Knapsack Problems: Algorithms and Computer Implementations
D6: Methods of Solving Singular Systems of Ordinary Differential Equations
D7: Nonlinear Systems
D8: Ordinary Differential Equations
D9: Oscillation Theory for Neutral Differential Equations with Delay
D10: Oscillation Theory of Delay Differential Equations
D11: Pseudodifferential Operators and Nonlinear Partial Differential Equations
D12: Sinc Methods for Quadrature and Differential Equations
D13: Stability of Stochastic Differential Equations with Respect to Semi-Martingales
D14: The Boundary Integral Approach to Static and Dynamic Contact Problems
D15: The Double Mellin-Barnes Type Integrals and Their Applications to Convolution Theory

Dataset 2 consists of 15 book titles from book reviews published in the December 1993 issue (volume 54, number4) of SIAM Review [16]. We use the same way to process words as that of experiment 1. Corresponding to the text in Table 3 is the word-document matrix $A = (a_{mn})_{16\times15}$ shown in Table 4

Table 4 The word-document matrix $A = (a_{mn})_{16\times15}$ for dataset 2

Words	Documents														
	D1	D2	D3	D4	D5	D6	D7	D8	D9	D10	D11	D12	D13	D14	D15
algorithms	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0
application	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
delay	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
differential	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
equations	0	1	0	0	0	1	0	1	1	1	1	1	1	0	0
implementation	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0
integral	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
introduction	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0
methods	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0
nonlinear	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0
ordinary	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
oscillation	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0
partial	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0
problem	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
systems	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0
theory	1	0	0	0	0	0	0	0	1	1	0	0	0	0	1

Figure 2 shows the Φ-K curve. It is similar to the result of experiment 1. The value of Φ goes up as K increases at the beginning; when the value of K is equal to 3, Φ reaches the maximum value of 88.127. The value of Φ then goes down while K increases continuously.

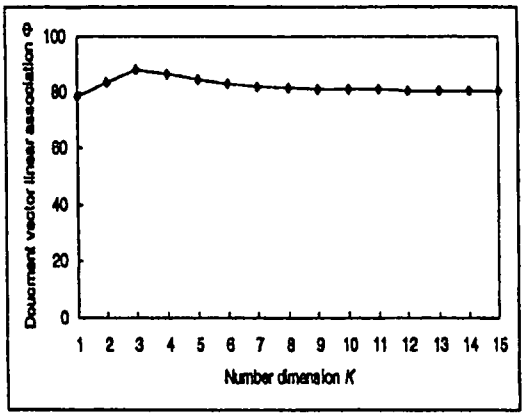

Figure 2 Document vector linear association Φ vs. number dimension K for the dataset 2

From Figure 1 and Figure 2, the observation illustrates that for a given dataset, a maximum value of Φ can be obtained when K is equal to a particular value. Thus we proposed a theory argument that for a given dataset, the document vector linear association Φ reflects LSI performance. It means that if Φ reaches a maximum value at a specific value K, the LSI can achieve the best performance with the K.

3.3 Simulation Experiments

3.3.1 Simulation Experiment 1

We generated a dataset consisting of 1000 documents by using simulation. There are 100 catalogues. Each catalogue has 10 documents and each document consists of 150 words which are randomly generated within a certain range. For a catalogue C_i ($i = 1, 2, \ldots, 100$), the range is from ((*i*-1)*5000 + 1) to *i**5000. Here we assume that one digital number is one word. Words occurring in more than one document are selected for indexing. We have the word-document matrix $A = (a_{mn})_{16974\times1000}$. Figure 3 shows the Φ-$K$ curve.

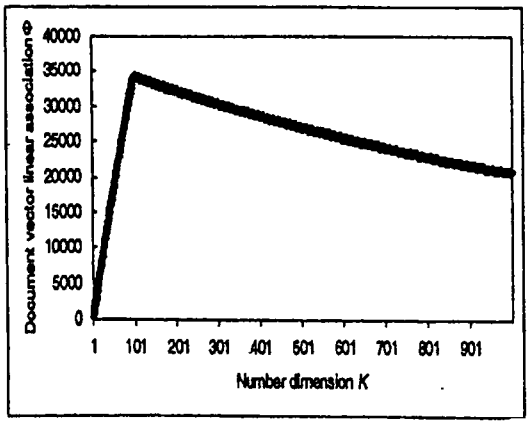

Figure 3 Document vector linear association Φ vs. number dimension K for simulation 1

It is similar to the result of experiment 1 and 2. The value of Φ goes up as K increases at the beginning; when the value of K is equal to 100 (Shown in Figure 4), Φ reaches the maximum value of 34243.5. The value of Φ then goes down while K increases continuously.

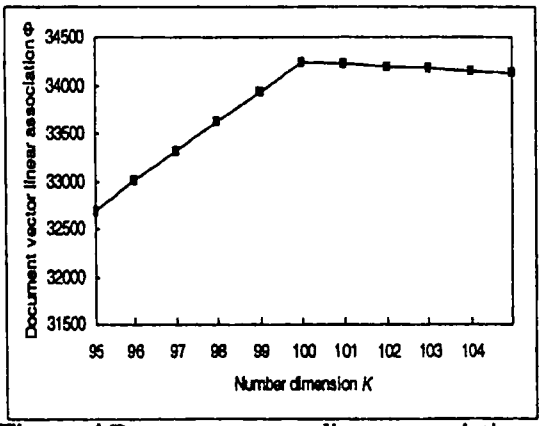

Figure 4 Document vector linear association Φ vs. number dimension K for simulation Φ dataset 1 (More details about 100)

3.3.2 Simulation Experiment 2

Similarly to the result of simulation experiment 1, we generated a dataset consisting of 1000 documents by using simulation. There are 50 catalogues. Each catalogue has 10 documents and each document consists of 150 words which are randomly generated within a certain range. For a catalogue C_i (i = 1, 2, ..., 50), the range is from ((i-1)*5000 + 1) to i*5000. For the rest 500 documents, each document also consists of 150 words which are randomly generated in the whole range (i.e. from 1 to 100*5000). We use the same way to process words as that of simulation experiment 1. We have the word-document matrix $A = (a_{mn})_{20674\times1000}$. Figure 5 shows the Φ-$K$ curve .

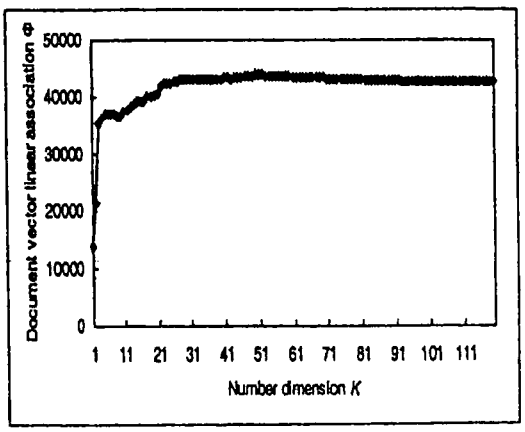

Figure 5 Document vector linear association Φ vs. number dimension K for simulation 2

It is similar to the result of simulation experiment 1. The value of Φ goes up as K increases at the beginning; when the value of K is equal to 50 (Shown in Figure 6), Φ reaches the maximum value of 43832.8. The value of Φ then goes down while K increases continuously.

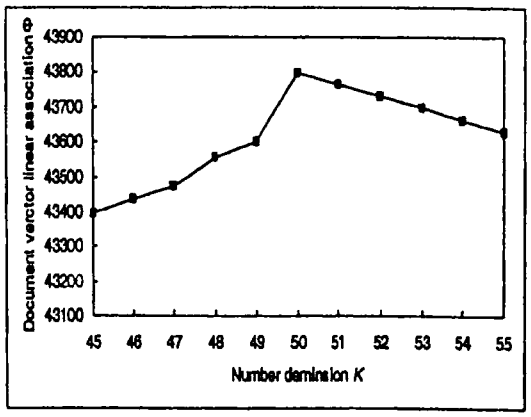

Figure 6 Document vector linear association Φ vs. number dimension K for simulation Φ dataset 2 (More details about 50)

From Figure 3 and Figure 5, the observation also illustrates that for a given dataset, a maximum value of Φ can be obtained when K is equal to a particular value, 100, 50 for simulation 1, 2 respectively, which is exactly corresponding the number catalogue of the simulation dataset 1, 2 respectively. It means that for a given dataset, the relationship between document vector linear association Φ and dimension number has a potential ability to detect the dataset structure.

4. Performance Evaluations

4.1 Set the threshold for query retrieval

Queries are treated as pseudo-documents [1]. In order to compare a query or pseudo-document to other documents of matrices, we treat this pseudo-document as the same as original documents. That is we just add one extra column vector of words to an original matrix. To set an appropriate threshold for query retrieval is very important in information retrieval applications. If the threshold is too low, lots of irrelevant documents will be achieved, however many relevant documents will be missed, if the threshold is

too high. To set the threshold for our query retrieval, we think that given a dataset and query, a threshold should consist of three components, namely the query information, the original dataset information and the document vector linear association Φ information. We compute a threshold using the following algorithm.

1) Using a word-document matrix, run LSI system to find Φ-K relationship.
2) According to Φ-K curve, identify the K value when Φ achieves the maximum value.
3) After adding a query vector to the original matrix, run LSI system with the K value and get the matrix A_K corresponding to that K.
4) Compute a sum of dot products Φ_q between a query vector and all other column vectors in the matrix A_K.
5) Finally calculate an average value as a threshold T

$$T = \Phi_q / N \quad (9)$$

4.2 Query retrieval experiment

We treat a query as input to the retrieval system, and a ranked list of returned documents as output. Those documents their dot product with a query vector is greater than or equal to the query retrieval threshold T are kept. Table 5 and 6 show the query retrieval experiment results for dataset 1 and dataset 2 respectively.

Table 5 Query experiment results for dataset 1

Conditions		K									
		1	2	3	4	5	6	7	8	9	10
Q1	NDF	5	5	4	3	3	3	3	3	3	3
T 0.409	NDFC	5	5	4	3	3	3	3	3	3	3
Q2	NDF	0	4	4	4	3	3	3	3	3	3
T 0.297	NDFC	0	4	4	4	3	3	3	3	3	3

K denotes the number of dimensions, T denotes the retrieval threshold, Q1: "human computer interaction", Q2: "trees", NDF denotes the number of documents found, and NDFC denotes the number of documents found and corrected

4.3 Performance Measures

We present performance evaluation based on a precision of query retrieval results which is similar to the corresponding measures used in traditional text retrieval [18]. A precision P is defined as

$$P = \sum_{q=1}^{Q} d_q / \sum_{q=1}^{Q} d_q' \quad (9)$$

where Q denotes the number of queries, $q = 1, 2, \ldots, Q$, d_q denotes the number of documents found and corrected using LSI method for the qth query, d_q' denotes the number of documents found manually for the qth query. Figure 7 and Figure 8 show precision measurement results for dataset 1 and dataset 2 respectively.

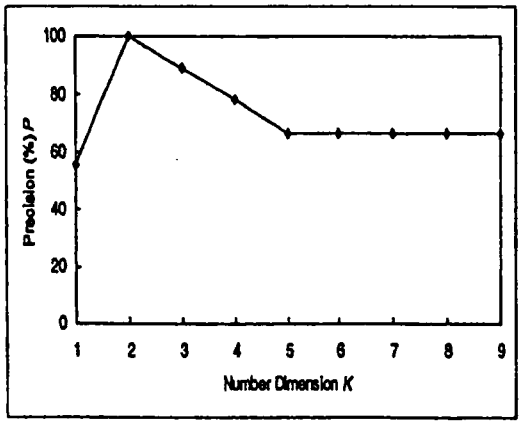

Figure 7: Precision vs. number dimension K for dataset 1

Experiment results demonstrate that LSI achieves the best performance when the K value is equal to 2 or 3 for dataset 1 or dataset 2 respectively. The performance evaluations have clearly confirmed our proposed argument. That is if Φ reaches the maximum value at a specific value K, LSI achieves the best performance with the K.

5. Conclusions

Choosing an appropriate number of dimensions for the LSI is the principle issue for the application of LSI technology. In this paper we have presented document vector linear association method to solve this problem. Its fundament is that LSI achieves the best performance when a sum of total dot products between all document vectors reaches a maximum value at a particular number of dimensions. The performance evaluations demonstrated that this method can choose an appropriate number of dimensions for LSI. Further investigations will be carried out to explore its full advantages in large datasets.

References

[1] S. Deerwester, S. T. Dumais, G. W. Furnas, Landauer. T. K., and R. Harshman. Indexing by latent semantic analysis. Journal of the American Society for

Information Science, 41(6): 391–407, 1990.

[2] G. Salton and M. J. McGill. Introduction to Modern Information Retrieval. McGraw–Hill, 1983.

[3] P.W. Foltz and S. T. Dumais. An analysis of information filtering methods. Communications of the ACM, 35(12): 51–60, 1992.

[4] T.K. Landauer and S.T. Dumais. A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review, 104(2): 211–240, 1997.

[5] J.R. Bellegarda. Exploiting both local and global constraints for multi-span statistical language modeling. http://www.telecom.tuc.gr/paperdb/icassp98/pdf/scan/ic981164.pdf

[6] Cullum, J.K. and Willoughby, R.A. Lanczos algorithms for large symmetric eigenvalue computations–Vol 1 Theory, (Chapter 5: Real rectangular matrices). Brikhauser, Boston, 1985.

[7] Dumais, S. T. Improving the retrieval of information from external sources. Behavior Research Methods, Instruments and Computers, 23(2): 229–236, 1991.

[8] Dumais, S. T. LSI meets TREC: A status report. http://trec.nist.gov/pubs/trec2/papers/txt/10.txt

[9] Nello CRISTIANINI, John Shawe–Taylor, Huma Lodhi. Latent Semantic Kernels. Journal of Intelligent Information Systems, 18:2/3, 127–12, 2002.

[10] Gene H. Golub, Franklin T. Luk and Michael L. Overton .A block lanczos method for computing the singular values and vorresponding singular vectors of a matrix. ACM Transactions on Mathematical Software, 17(2): 149–169, 1981

[11] Landauer, T. K., Laham, D., Rehder, R., and Schreiner, M.E. How well can passage meaning be derived without using word order? A comparison of Latent Semantic Analysis and humans. http://lsa.colorado.edu/papers/cogsci97.pdf

[12] Rehder, B., Schreiner, M., Laham, D., Wolfe, M., Lan-dauer, T., & Kintsch, W. Using LatentSemantic Analysis to assess knowledge: Some tech-nical considerations. Discourse Processes, 25: 337–354, 1998.

[13] Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser,A. Improving an intelligent tutor's compre-hension of students with Latent Semantic Analysis.In Lajoie, S., & Vivet, M. (Eds.), Artificial Intelligence in Education, 535–542, 1999(Amsterdam. IOSPress).

[14] Wiemer Hastings Peter.Adding syntactic information to LSA. 2000, http://reed.cs.depaul.edu/peterwh/papers/cogsci00.pdf

[15] Noriaki Kawamae. Latent semantic indexing based on factor analyssis. http://ultimavi.arc.net.my/banana/Workshop/SCI2002/papers/Kawamae.pdf.

[16] M.W. Berry, S.T. Dumais & G.W. O'Brien. Using Linear Algebra for Intelligent Information Retrieval. http://citeseer.nj.nec.com/rd/37422678,19079,1,0.25,Download/http://citeseer.nj.nec.com/cache/papers

[17] Sparck Jones, K. A statistical interpretation of term specificity and its applications in retrival. Journal of Documental, 28(1): 11–21, 1972.

[18] Yiming Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(1/2): 67–88, 1999.