LSA: First dimension and dimensional weighting

Hu, X.[1], Cai, Z.[1], Franceschetti, D.[2],
Penumatsa, P.[1], Graesser, A. C.[1], Louwerse, M. M.[1], McNamara, D. S.[1]
and the Tutoring Research Group[1]
[1]Department of Psychology;
[2]Department of Physics;
The University of Memphis, Memphis, TN 38152

Abstract

We report two discoveries concerning Latent Semantic Analysis (LSA). First, we observed the special properties of the first dimension of the LSA space. Second, we observed that dimensional weighting plays an important role in LSA analysis. Based on the first discovery, we examined the cosine matches without the first dimension. Based on the second discovery, we explored different dimensional weighting schemes. Based on these observations, we recommend a new algorithm for LSA cosine computation such that LSA becomes more sensitive to relevant similarities and differences.

Introduction

Latent Semantic analysis (LSA) is an application of principal component analysis (PCA) to natural language understanding. LSA reduces a large body of documents into a compact matrix representation (with only a few hundred columns), such that each row of this matrix represents a word (or a token) that appears in the document collection in the form of vector. Such a mathematical representation of words as vectors captures the semantic relationship between words in the following way. If two words appear in the same document environment, accompanied by similar words, then the vector representations of the two words are similar; that is, the normalized dot product (cosine) of the two vectors is close to 1. In the same way similarity measures can be obtained for sentences, paragraphs, or documents. For instance, for any document consisting of a list of words, the vector corresponding to a sentence, paragraph, or document is simply a weighted vector summation of the vectors for the included words. This property of a reduced semantic matrix representation has proven to be extremely useful in a range of applications in natural language processing (see Foltz, Kintsch, and Landauer, 1998 for an overview).

LSA was originally developed for information retrieval (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990). In those cases, for any query document, the corresponding vector (called the query vector) is used to obtain a cosine match with all rows of the document vectors corresponding to the documents in corpus. By using the cosine match as the similarity measure, best candidates can be obtained and they are assumed to be semantically similar to the query document. Application of this use of LSA has been documented in several of our current applications (Olde, Franceschetti, Karnavat, Graesser, & TRG, 2002; Graesser, Hu, Person, Jackson, & Toth, 2002).

Later uses of LSA went beyond document retrieval to measure semantic similarity between documents (Wiemer-Hastings, Wiemer-Hastings, Graesser, & TRG, 1999). This use of LSA is more efficient than retrieving documents from the original corpus. To determine the semantic similarity of two documents, only one cosine is needed and is computed from word vectors, whereas thousands of cosine calculations must be completed for document retrieval. In recent years, researchers have used LSA to compute semantic similarities in a variety of applications. For instance, LSA has been used as an automated essay grader, comparing student essays with ideal essays (Foltz et al., 1998). It has been used as a measurement of coherence between successive sentences (Foltz et al., 1998). LSA has in fact shown to perform as well as students on the TOEFL test (Test Of English as a Foreign Language) (Landauer & Dumais, 1997) and can even be used for understanding metaphors (Kintsch, 2000). Finally, LSA has played a crucial role in the representation of world knowledge in intelligent tutoring systems. For example, LSA is used in AutoTutor, an intelligent tutoring system that has tutorial conversations with students on a variety of topics. Currently, AutoTutor has been developed for computer literacy and conceptual physics (Graesser, VanLehn, Rose, P., & Harter, 2001). AutoTutor uses LSA to give meaning to a student answer and to match that answer to ideal good answers and bad answers (Graesser et al., 2000; Franceschetti et al., 2001; Olde et al., 2002).

In our work on AutoTutor, we observed that when similarity is computed between two documents, the cosine value is positively related to the size of these documents. We believe this is an unfortunate artifact and demonstrate here that the performance of LSA can be improved significantly by excluding the first dimension of the LSA space in calculating the cosines between document vectors. To explain how the performance of LSA can be improved, we first briefly describe the mathematical foundations of LSA and how LSA is used in AutoTutor. We then prove two theorems that show how LSA document matching can be improved. Finally we provide some general recommendations on how LSA should be used in natural language understanding applications.

Mathematical foundations of LSA

The fundamental assumption of LSA is that the semantics of any natural language are expressed by the way humans use that language. According to this assumption, the meaning of words is entirely based on co-occurrences with other words. When the language units share a common environment, they must be similar in meaning. Two words have similar meaning if they accompanied by similar words(Landauer & Dumais, 1997). For example, for a given corpus of text that contains N distinct document environments defined as sentences, paragraphs, or clusters of paragraphs, any word that occurs in the corpus can be expressed as a N-dimensional vector. Each value in the vector is either non-zero (a function of how the words appear in the paragraph) or a zero value (in case the word is not in the paragraph).

Consider Grolier's (1996) Multimedia Encyclopedia as a corpus. This corpus contains 44,227 paragraphs with a total of 76,948 distinct words. A vector with 44,227 dimension can be used to uniquely identify each of the distinct words. In this vector, the components along each dimension are computed as a function of how often each word is used in the corresponding paragraph. For example, assume the m^{th} word is table (in the list of 76,948 distinct words) and appears in paragraph n, then the vector representation for table has a non-zero value at the n^{th} element with a value of $f(m,n)$ £ $G(m)$ £ $L(m,n)$, where: $f(m,n)$ is a function of the frequency of table occurring in the m^{th} paragraph; $G(m)$ represents the weight of the m^{th} word independent of the paragraph in which it occurs; and $L(m,n)$ represents the weight of the m^{th} word (i.e., table) depending on the paragraph in which it occurs (i.e., the n^{th} paragraph). In the literature (Deerwester et al., 1990), $G(m)$ is called global weighting and $L(m,n)$ is called local weighting.

Assume matrix A is obtained for a given corpus, then singular value decomposition (SVD) is performed on A. SVD will produce three matrices: Two orthonormal matrices, U and V, and one diagonal matrix § = $diag(\sigma_1, ..., \sigma_r)$, with elements appearing in descending order, where r is the rank of A, such that

$$\mathrm{A} = \mathrm{U}§\mathrm{V}^{l}. \quad (1)$$

LSA truncates the U matrix into k columns (from original m columns) and use the truncated U matrix, called term-matrix (throughout the remainder of this paper we will use U_k to denote the truncated matrix). Each row of the U_k is a k dimensional real-valued vector. Similarly, the V matrix is also truncated into k columns from original n columns. The truncated V is called document-matrix (denoted as V_k).

In LSA, words are represented as real-valued vectors from the truncated U matrix. For any paragraph, a real-valued vector can be computed from the vectors that correspond to the terms in the paragraph. The formula for obtaining the vector is in the form

$$\mathrm{x}^{l}\,\mathrm{W}\,\mathrm{U}_k ¤_k, \quad (2)$$

where x is an m £ 1 vector with entries either 1 or 0, W is a m £ m diagonal matrix and ¤$_k$ is a diagonal k £ k matrix. After vectors are obtained for each paragraph, then the similarity of the two paragraphs are measured as the cosine value of the two vectors:

$$s(\mathrm{x}_1, \mathrm{x}_2) = \frac{(\mathrm{x}_1^{l}\,\mathrm{W}\,\mathrm{U}_k ¤_k)\,(\mathrm{x}_2^{l}\,\mathrm{W}\,\mathrm{U}_k ¤_k)^{l}}{\mathrm{k}\mathrm{x}_1^{l}\,\mathrm{W}\,\mathrm{U}_k ¤_k\mathrm{k}\mathrm{k}\mathrm{x}_2^{l}\,\mathrm{W}\,\mathrm{U}_k ¤_k\mathrm{k}}. \quad (3)$$

Accordingly, we can formulate the following definition:

Definition Assuming all words are indexed from 1 to m, $x_1 = (x_{1,1}, ..., x_{1,m})$ and $x_2 = (x_{2,1}, ..., x_{2,m})$ are two m£1 vectors corresponding two documents, such that

½

$$x_{i,j} = \begin{matrix} 1 & \text{word j is in document i} \\ 0 & \text{word } j \text{ is not in document } i \end{matrix}$$

The similarity of the two documents is defined as Equation (3).

When $k = r$ then U_k = U and ¤$_k$ = ¤.

Observations and Theorems

In this section, we first report observations and results related to the special properties of the first dimension of LSA space. Then we extend these findings to explore alternative ways of computing similarity measures using different dimensional weighting.

The importance of the first dimension in LSA

As described earlier, the intelligent tutoring system AutoTutor evaluates student answers by comparing them with ideal good answer information and bad answers as defined by experts in the system's course curriculum scripts. Physics textbooks are used to create an LSA space (see (Franceschetti et al., 2001; Olde et al., 2002)). Based on the LSA cosine match between student answers and ideal good and bad answers, AutoTutor decides whether or not a student answered the question correctly and determines its next tutorial dialog move. However, the length of the utterances in the student answer and AutoTutor's expectation show an interesting phenomenon. We randomly selected n ($n = 1, 2, 4 \ldots 512$) words from a random pool of student contributions. That is, we mimicked student contributions with variable numbers of words. Next, we randomly selected m ($m = 1, 2, \ldots 512$) words from the expectations. Mimicking the authored physics curriculum scripts with variable numbers of words, entries in Table 1 show the average cosine match values (100 cosines computed per cell) between the student contributions and the curriculum scripts. Notice that the average cosine match between documents with 256 words is 0.450, while the average cosine between documents with 16 words is 0.086. Our first observation below is:

Observation 1 The cosine match between documents is monotonically related to the number of words contained in documents.

	4	16	64	256	512
4	0.033	0.051	0.093	0.130	0.152
16	0.047	0.085	0.146	0.211	0.245
64	0.079	0.129	0.244	0.345	0.390
256	0.106	0.167	0.324	0.458	0.509
512	0.107	0.176	0.336	0.483	0.535

Table 1: Average cosine values for different document sizes.

To understand the above observation, we must further examine the LSA spaces that are used in AutoTutor. We observed that the first dimension of each term vector always had the same sign (either all non-negative or all non-positive). We found the same phenomenon within a variety of other corpora (encyclopedia, science texts, etc.).

The following theorem provides a formal proof for the above observation.

Theorem 1 Let A be an $m \times m$ real matrix with all entries non-negative. Suppose the singular value decomposition of A is (1), where $\Sigma = diag(\sigma_1, ..., \sigma_r)$. Assume $\sigma_1 > \sigma_2 \geq ..., \sigma_r$. Then all entries in the first column of U are of the same sign (all non-negative or all non-positive).

Proof The first column of U is actually an eigenvector with a unit length corresponding to the maximum eigenvalue σ_1^2 of the symmetric matrix AA^T. Since σ_1^2 is greater than all other eigenvalues of AA^T, an eigenvector of AA^T corresponding to the eigenvalue σ_1^2 can be written as (Power Approximation Method)

$$x_1 = \lim_{k \to \infty} \frac{(AA^T)^{k+1} x}{\left\|(AA^T)^{k+1} x\right\|} \qquad (4)$$

where x is any vector in R^m that is not orthogonal to x_1. We can always find an x not orthogonal to x_1 from the m orthonormal vectors $e_1, ..., e_m$, where the i^{th} entry of e_i, $(i = 1, ..., m)$ is 1 and all other entries are 0. Therefore, we can assume that all entries of x are non-negative. Because all entries of A are also non-negative, from (4) we can see that all entries of AA^T are non-negative. Since $\pm x_1$ are the only two unit eigenvectors corresponding to σ_1^2, the first column of U is one of these two vectors, their entries are either all non-negative or all non-positive.

To further examine the numerical details, we have examined the arithmetic means of the elements in each column of the U matrix. We observed that the first column has a significantly larger mean than all other dimensions. This can be measured by the ratio between the mean of the first dimension and the length of the vector of the means for all the dimensions. Denote m_i as the mean for i^{th} dimension. For example, for the physics LSA space (with 338 dimensions), $m_1 / \sqrt{\sum_{i=1}^{338} m_i^2} = 0.833$. Table 3 shows the same observation we found across corpora.

Space	# Dim	1^{st} Dim Mean	Mean length	Ratio
Physics	338	0.00528	0.00634	0.833
Science	365	0.00348	0.00479	0.726
Encyclopedia	496	0.00079	0.00119	0.669
Narrative	277	0.00139	0.00270	0.516

Table 2: Relative magnitude of first dimension for selected LSA spaces.

Observation 2 The magnitude of the first component of any document vector is significantly larger than all other components.

Observation 2 and Theorem 1 explain Observation 1. For any document with a large number of terms, the LSA vector of the document is simply the vector summation of the term vectors. While all other dimensions may cancel out, the first dimension just keeps adding up, so it is the value of the first dimension that drives the cosine value. This critical observation led us to further examine the numerical properties of the first dimension. We observed that the common words, such as is, the, and an, have larger values in their first dimension and rare words have smaller values. In LSA, a measure of the commonness is measured by weights. We examined one of the LSA spaces used in the AutoTutor project. The correlation between the first dimension and the weights was as high as -0.757 (6679 words). Table 3 shows corresponding quantities for all other corpora for which we have created LSA spaces. These spaces include physics textbooks, science textbooks, Grollier's encyclopedia, the encyclopedia augmented with WordNet, and a large sample of narrative texts.

Observation 3 Weights and the first dimension of the LSA vectors are negatively related.

Space Name	#1
Physics	−0.757
Science	−0.686
Encyclopedia	−0.573
Narrative	−0.500

Table 3: Correlations between the weight for words and the first dimension for the words.

The above observations and theorems point out that the first dimension of LSA vector needs special treatment. Table 4 shows the cosine matches between document vectors without the first dimension. We observed that removing the first dimension made a greater difference when the documents being compared were large. Notice in Table 4, that when documents have fewer than 64 words (which is more than most student contributions and expectations in AutoTutor), the LSA cosine matches are no longer a function of the document size. In contrast, when documents have more than 64 words, the cosine matches increase as a function of document size, but in a smaller magnitude than when the first dimension is retained.

	4	16	64	256	512
4	−0.02	−0.02	−0.06	0.07	−0.00
16	−0.03	0.00	0.03	0.01	0.11
64	0.01	0.09	0.06	0.02	0.08
256	0.05	0.10	0.17	0.12	0.22
512	0.04	0.13	0.14	0.12	0.20

Table 4: Average cosine values as a function of document size. For these cosines, the first dimension was removed, otherwise, the computations for the cosines were the same as those in Table 1.

The importance of dimensional weighting in LSA

In the previous section, we showed the importance of the first dimension of LSA vector. This finding motivated us to explore different ways to obtain cosine matches between documents. In particular, we explored different dimensional weighting in the computation of the cosine match between documents. First, we examined Equation (3). We obtained the following formal results that link different measures of similarity in a unified formulation.

Theorem 2 From the Definition before, assuming $k = r$, then

1. If Λ is a unit matrix, then Equation (3) is equivalent to weighted word matching.
2. If $\Lambda = \Sigma$, then Equation (3) is equivalent to similarity comparison using original term-document matrix.

Proof : 1) Assume $k = r$ and Λ is unit matrix,

$$\begin{aligned} & (x_1^T W U \Lambda)(x_2^T W U \Lambda)^T \\ = & (x_1^T W U \Lambda \Lambda U^T W x_2) = (x_1^T W U U^T W x_2) \\ = & (x_1^T W W x_2) = (x_1^T W)(x_2^T W)^T \end{aligned}$$

so,

$$\frac{(x_1^T W U \Lambda)(x_2^T W U \Lambda)^T}{\|x_1^T W U \Lambda\| \|x_2^T W U \Lambda\|} = \frac{(x_1^T W)(x_2^T W)^T}{\|x_1^T W\| \|x_2^T W\|}.$$

Notice that

$$x_1^T W W x_2 = \sum_{\text{word } i \text{ is in both documents}} w_i^2$$

2) Assume $k = r$ and $\Lambda = \Sigma$,

$$\begin{aligned} & (x_1^T W U \Sigma)(x_2^T W U \Sigma)^T \\ = & x_1^T W U \Sigma \Sigma^T U^T W x_2 = x_1^T W U \Sigma V^T V \Sigma^T U^T W x_2 \\ = & (x_1^T W U \Sigma V^T)(x_2^T W U \Sigma V^T)^T = (x_1^T W A)(x_2^T W A)^T \end{aligned}$$

therefore:

$$\frac{(x_1^T W U \Lambda)(x_2^T W U \Lambda)^T}{\|x_1^T W U \Lambda\| \|x_2^T W U \Lambda\|} = \frac{(x_1^T W A)(x_2^T W A)^T}{\|x_1^T W A\| \|x_2^T W A\|}.$$

LSA truncates the U matrix into U_k, which only contains a few hundreds columns and is used as an approximation of the original term-document matrix A. Theorem 2 and the Definition presented before can be understood as the following: (1) without dimensional weighting, the LSA cosine matching approximates weighted keyword matching, and (2) using dimensional weighting with singular values, LSA cosine matching approximates context similarity.

To explore other possible dimensional weighting, we examined cosines using several different dimensional weighting schemes: Consider the dimensional weight matrix $\Lambda = diag(\lambda_1, ..., \lambda_k)$,

$$\lambda_1 = 0, \lambda_i = \sigma_i - \sigma_{k+1}, i \geq 2 \quad (5)$$

$$\lambda_1 = 0, \lambda_i = 1, i \geq 2 \quad (6)$$

$$\lambda_1 = 0, \lambda_i = \sigma_i, i \geq 2 \quad (7)$$

$$\lambda_i = \sigma_i - \sigma_{k+1}, i \geq 1 \quad (8)$$

$$\lambda_i = 1, i \geq 1 \quad (9)$$

$$\lambda_i = \sigma_i, i \geq 1 \quad (10)$$

Figure 1 gives examples showing LSA performance for the two dimensional weighting schemes. We systematically changed the number of dimensions used in the LSA space (from 100 to 400, step size 10). Using the weighting in (5) above, for related words (for example, cat and dog) and non-related words (for example, force and blue), the cosine match is a smooth function of the number of dimensions used. This is not true for the weighting method (6) above.Researchers have previously examined the optimization of the number of dimensions used for an LSA space (Landauer & Dumais, 1997). It has been shown that there exists a range of about 300 § 100 dimensions that provide the best overall LSA performance for most corpora. The above observation of smooth curves as function of number of dimensions shows that with particular dimensional weighting scheme, there may be different ways in which the optimum range of dimensions can be selected.

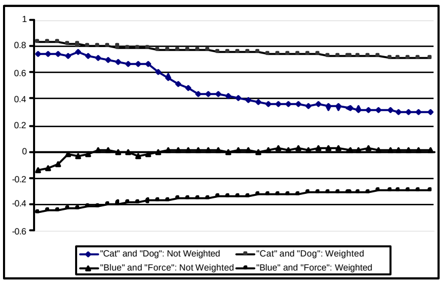

Figure 1: Cosines with two different types of dimensional weighting.

In addition, the smoothness of the LSA similarity between words when the number of dimensions change makes intuitive sense. For example, one would not expect huge differences between cosine matches using 300 dimensions and 310 dimensions. The smoothness of the curve we observed here is due to differential weighting of the dimensions. For example, if one chose 300 dimensions, then the weight for the 300^{th} dimension is the smallest (approaching zero, it plays the least amount of importance in the computation). We have also explored other dimensional weighting schemes such as (7), (8), (9), and (10) defined above. Weighting method (5) performs the best in the sense that it provides the smoothest function.

We further explored the dimensional weighting on 3 carefully selected sets of documents from AutoTutor. Each set had 42 sentences. Sentences in set I and set II were from the AutoTutor curriculum script for conceptual physics. Each sentence in set I was an expected answer of a question; the corresponding sentence in set II was an alternative answer. Therefore sentences in set I and set II were similar in meaning in a pair-wise manner. Sentences in set III were randomly selected from the NSF AutoTutor project proposal, pair-wise dissimilar to the sentences in set I. We computed the LSA cosines between the similar pairs and dissimilar pairs. The discriminability was then calculated by

$$d = \left(m_1 - m_2 \right) \Big/ \left(\sqrt{\frac{sd_1^2 + sd_2^2}{2}} \right)$$

where m_1 and sd_1 are the mean and standard deviation for the cosine matches between set I and set II (cosine for the similar pairs), and m_2 and sd_2 are the mean and standard deviation for the cosine matches between set III and set II (cosine for the dissimilar pairs) The results of these analysis are shown in Table 5. We observed better performance for the weighting method (5) in the sense that it provides the highest discriminability.

	not weighted	weighted
keep 1^{st} dim	method (9) 3.408	method (8) 2.422
remove 1^{st} dim	method (6) 3.573	method (5) 6.297

Table 5: Discriminability for the four different weighting methods.

Conclusions

In this paper, we made two important observations about LSA. The first observation concerns the non-negative (or non-positive) first component of LSA vectors. This is the primary reason that the similarity measure between any two documents is an increasing function of the document size. The second important observation (outlined in Theorem 2) is that there is a unified mathematical expression for three different types of document similarity measures, namely (1) LSA cosine match, (2) weighted word match, and (3) context similarity match. Such a unified representation led us to explore different dimensional weighting methods. We observed that the weighting methods described in (5) outperformed other weighting methods, including the method used in the lion's share of LSA literature. Specifically, it provided the smallest cosine changes as a function of the dimension of the space and better discriminations between sentence pairs. With the above two findings, we have the following recommendations for future use of LSA.

Theorem 1 states that the first non-zero components of all the LSA term vectors will have the same sign. When constructing a vector to represent a document by summing term vectors, the first component thus will grow in magnitude while the other components need not. We therefore recommend against interpreting absolute cosine values as a measure of the "match" between documents. Instead, we recommend establishing statistical distributions of the cosine match values for the target space and using these relative units to judge the similarity. For example, for documents x_1 and x_2 with document sizes n_1 and n_2, if a baseline (theoretical or empirical) distribution with mean μ_{n_1,n_2} and sd_{n_1,n_2} is obtained, then the relative similarity can be computed as the following relative score:

$$\frac{s(\mathrm{x}_1, \mathrm{x}_2) - \mu_{n_1,n_2}}{sd_{n_1,n_2}} \tag{11}$$

where $s(\mathrm{x}_1, \mathrm{x}_2)$ is defined in Equation (3). This recommendation is particularly useful when LSA is used in applications where the similarity measures are used as selection criteria, as in AutoTutor. It is not necessary in document retrieval applications, where the similarity measure is primarily used for ordering the potential candidates.

Even if one uses relative score in the form of Equation (11), we recommend using dimensional weighting, especially the dimensional weighting matrix in the form of (5). By using dimensional weighting scheme outlined in (5), the similarity measures are relatively robust with respect to number of dimensions used the LSA space. Furthermore, weighting scheme (5) removes the influence of the first dimension and thereby increases discriminability when similarity is used in selecting candidates among multiple alternatives.

Acknowledgments

The Tutoring Research Group (TRG) is an interdisciplinary research team comprised of approximately 35 researchers from psychology, computer science, physics, and education (visit http://www.autotutor.org). This research conducted by the authors and the TRG was supported by the National Science Foundation (REC 0106965), the DoD Multidisciplinary University Research Initiative (MURI) administered by ONR under grant N00014-00-1-0600, and Institute of Education Sciences (IES R3056020018-02). Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of ONR, NSF, or IES. We would like to thank Tom Landauer and Walter Kintsch by supplying a number of corpora used in this study.

References

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. Journal of the American Society For Information, 141, 391-407.

Foltz, P. W., Kintsch, W., & Landauer, T. K. (1998). The measurement of textual coherence with latent semantic analysis. Discourse Processes, 25, 285-307.

Franceschetti, D., Karnavat, A., Marineau, J., McCallie, G. L., Olde, B. A., Terry, B. L., & Graesser, A. C. (2001). Development of physics text corpora for latent semantic analysis. In Proceedings of the 23rd annual conference of the cognitive science society (p. 297-300). Mahwah, NJ: Erlbaum.

Graesser, A., Hu, X., Person, P., Jackson, T., & Toth, J. (2002). Modules and information retrieval facilities of the human use regulatory affairs advisor (HURAA). In M. Driscoll & T. C. Reeves (Eds.), Proceedings for e-learning 2002: World conference on e-learning in corporate, government, healthcare, and higher education (p. 353-360). Montreal, Canada: AACE.

Graesser, A., VanLehn, K., Rose, C., P., J., & Harter, D. (2001). Intelligent tutoring systems with coversational dialogue. AI Magazine, 22, 39-51.

Graesser, A., Wiemer-Hastings, P., Wiemer-Hastings, K., Harter, D., Person, N., & TRG. the. (2000). Using latent semantic analysis to evaluate the contributions of students in AutoTutor. Interactive Learning Environments, 8, 128-148.

Kintsch, W. (2000). Metaphor comprehension: A computational theory. Psyhonomic Bulletin and Review, 7, 257-266.

Landauer, T. K., & Dumais, S. T. (1997). A solution to plato’s problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. Psychological Review, 104, 211-240.

Olde, B., Franceschetti, D., Karnavat, A., Graesser, A., & TRG. (2002). The right stuff: Do you need to sanitize your corpus when using latent semantic analysis? In W. G. Gray & C. Schunn (Eds.), Proceedings of the 24rd annual conference of the cognitive science society (p. 708-713). Mahwah, NJ: Erlbaum.

Wiemer-Hastings, P., Wiemer-Hastings, K., Graesser, A., & TRG. (1999). Improving an intelligent tutor’s comprehension of students with latent semantic analysis. In S. Lajoie & M. Vivet (Eds.), Artificial intelligence in education (p. 535-542). Amsterdam: IOS Press.