How Latent is Latent Semantic Analysis?

Peter Wiemer-Hastings
University of Memphis
Department of Psychology
Campus Box 526400
Memphis TN 38152-6400
pwmrhstn@memphis.edu*

Abstract

Latent Semantic Analysis (LSA) is a statistical, corpus-based text comparison mechanism that was originally developed for the task of information retrieval, but in recent years has produced remarkably human-like abilities in a variety of language tasks. LSA has taken the Test of English as a Foreign Language and performed as well as non-native English speakers who were successful college applicants. It has shown an ability to learn words at a rate similar to humans. It has even graded papers as reliably as human graders. We have used LSA as a mechanism for evaluating the quality of student responses in an intelligent tutoring system, and its performance equals that of human raters with intermediate domain knowledge. It has been claimed that LSA's text-comparison abilities stem primarily from its use of a statistical technique called singular value decomposition (SVD) which compresses a large amount of term and document co-occurrence information into a smaller space. This compression is said to capture the semantic information that is latent in the corpus itself. We test this claim by comparing LSA to a version of LSA without SVD, as well as a simple keyword matching model.

1 Introduction

Although classical Natural Language Processing techniques have begun to produce acceptable performance on real world texts as shown in the Message Understanding Conferences [DARPA, 1995], they still require huge amounts of painstaking knowledge engineering and are fairly brittle in the face of unexpected input. Recently, corpus-based statistical techniques have been developed in the areas of word-tagging and syntactic grammar inference. But these techniques are not aimed at providing a semantic analysis of texts.

In the late 1980's, a group at Bellcore doing research on information retrieval techniques developed a statistical, corpus-based method for retrieving texts. Unlike the simple techniques which rely on weighted matches of keywords in the texts and queries, their method, called Latent Semantic Analysis (LSA), created a high-dimensional, spatial representation of a corpus and allowed texts to be compared geometrically. In the last few years, several researchers have applied this technique to a variety of tasks including the synonym section of the Test of English as a Foreign Language [Landauer *et al.*, 1997], general lexical acquisition from text [Landauer and Dumais, 1997], selecting texts for students to read [Wolfe *et al.*, 1998], judging the coherence of student essays [Foltz *et al.*, 1998], and the evaluation of student contributions in an intelligent tutoring environment [Wiemer-Hastings *et al.*, 1998; 1999]. In all of these tasks, the reliability of LSA's judgments is remarkably similar to that of humans.

The specific source of LSA's discriminative power is not exactly clear. A significant part of its processing is a type of principle components analysis called singular value decomposition (SVD) which compresses a large amount of co-occurrence information into a much smaller space. This compression step is somewhat similar to the common feature of neural network systems where a large number of inputs is connected to a fairly small number of hidden layer nodes. If there are too many nodes, a network will "memorize" the training set, miss the generalities in the data, and consequently perform poorly on a test set. The input for LSA is a large amount of text (on the order of magnitude of a book). The corpus is turned into a co-occurrence matrix of terms by "documents", where for our purposes, a document is a paragraph. SVD computes an approximation of this data structure of an arbitrary rank K. Common values of K are between 200 and 500, and are thus considerably smaller than the usual number of terms or documents in a corpus, which are on the order of 10000. It has been claimed that this compression step captures regularities in the patterns of co-occurrence across terms and across documents, and furthermore, that these regularities are related to the semantic structure of the terms and documents.

*This work was supported by grant number SBR 9720314 from the National Science Foundation's Learning and Intelligent Systems Unit.

In this paper, we examine this claim by comparing several approaches which assess the quality of student contributions in an intelligent tutoring situation. We use human judgments of quality as a baseline, and compare them to three different models: the full LSA model, a version of LSA without SVD, and a simple keyword-matching mechanism. The paper starts with a description of the quality judgment task, and describes how LSA was used to rate the contributions. In section 3, we describe the implementation of LSA without SVD, and compare it to the SVD results. In section 4, we compare these to a basic keyword matching algorithm which used both a weighted and an unweighted matching technique. We close with a discussion of these results.

2 Evaluating student contribution quality with LSA

To provide a baseline description against which the alternative methods can be judged, this section describes the rating task for both the humans and LSA, gives some technical details of the LSA implementation, and describes how it performed in relation to the human raters.

2.1 The quality evaluation task

As the litmus test for the various evaluation techniques, we have chosen the domain of an intelligent tutoring system called AutoTutor that was developed with the goal of simulating natural human-human dialogue [Wiemer-Hastings *et al.*, 1998]. The tutoring domain for this project was computer literacy. The main knowledge structure for AutoTutor was a curriculum script [Putnam, 1987] that contained 12 questions in each of three different topics: computer hardware, operating systems, and the internet. For each question in the curriculum script, there was a variety of information about expected student answers and possible follow-up dialogue moves. The questions were designed to be deep reasoning questions which for which a complete answer would cover several aspects. AutoTutor's curriculum script contained an expected good answer for each of the aspects of a question, as well as a prompt, hint, and elaboration that could potentially elicit that answer. The use of these dialogue moves was based on studies of human tutors [Graesser *et al.*, 1995]. Dialogue move rules decided which move to use based on the student's ability and on which expected good answers were already covered. LSA was the primary mechanism for determining that coverage based on comparisons between the student responses and the expected good answers. When a particular contribution achieved a cosine match above an empirically determined threshold, that aspect of the question was considered as covered for the purposes of the tutoring task. This approach led to the definition of the basic evaluation measure:

> **Compatibility = Matches / Propositions**, where Propositions is the number of speech acts in the student contribution, and Matches is the number of Propositions that achieved an above-threshold LSA cosine with one of the expected good answers for this question.

Loosely speaking, this is the percentage of the student's contribution that sufficiently matched the expected answer.

The test set for this task was based on eight questions from each of the three tutoring topics. Students in several sections of a university-level computer literacy course were given extra credit for typing in answers to the questions in a word processing document. They were encouraged to write complete, thorough answers to the questions. Eight substantive (i.e. not "I don't know") answers were randomly selected for each of the 24 questions, constituting a test set of 192 items.

2.2 Human Ratings

To assess the depth of knowledge that LSA uses, human raters of different levels of experience with the subject matter were used. Two raters, a graduate student and a research fellow, were computer scientists with high levels of knowledge of the computer literacy domain. Two additional raters, a graduate student and professor in Psychology, had intermediate-level knowledge. They were familiar with all of the text materials from the computer literacy domain that were used in the project.

The human raters were asked to break the student responses into propositions, i.e. parts that could stand alone in a sentence. Then they were asked to judge on a six-point scale the percentage of each student's propositions that "matched" part of the ideal answer. They were not instructed as to what should constitute a match. The correlation between the two expert raters was r=0.78. Between the intermediate knowledge raters, the correlation was r=0.52. The correlation between the average expert rating and the average intermediate rating was r=0.76. All of the correlations were significant at the 0.01 level.

2.3 LSA implementation

We briefly describe the LSA mechanism here in order to demonstrate the difference between it and the other approaches. Further technical details about LSA can be found in [Deerwester *et al.*, 1990; Landauer and Dumais, 1997] and several of the articles in the 1998 special issue of Discourse Processes on quantitative approaches to semantic knowledge representations.

As mentioned above, the basic input to LSA is a large corpus of text. The computer literacy corpus consisted of two complete computer literacy textbooks, ten articles on each of the tutoring topics, and the entire curriculum script (including the expected good answers). Each curriculum script item counted as a separate document, and the rest of the corpus was separated by paragraphs because they tend to describe a single complex concept. The entire corpus was approximately 2.3 MB of text. LSA defines a term as a word (separated by whitespace or punctuation) that occurs in at least two documents.

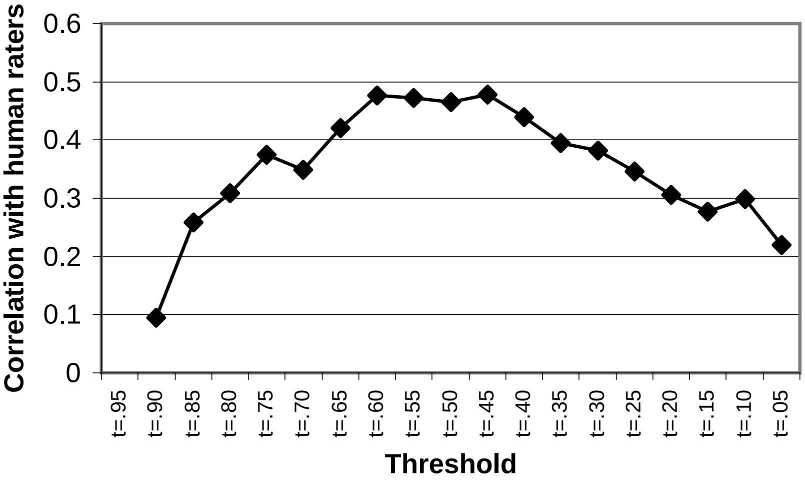

Figure 1: The correlation between LSA quality judgments and those of human raters.

There is also a list of about 400 very frequent words ("the", "and", and "for", for example) that are not used as terms. As previously mentioned, LSA creates from this corpus a large co-occurrence matrix of documents by terms, in which each cell is the number of times that that term occurred in that document. Each cell is then multiplied by a log entropy weight which essentially reduces the effect of words which occur across a wide variety of contexts (more about this later). SVD then creates a K-dimensional approximation of this matrix consisting of three matrices: a D by K documents matrix, a K by T terms matrix, and a K by K singular values (or eigenvalues) matrix (D is the number of documents, and T is the number of terms). Multiplying these matrices together results in an approximation to the original matrix. Each column of the terms matrix can be viewed as a K-long vector representing the "meaning" of that term. Each row of the documents matrix can be seen as a K-long vector representing the meaning of that document. Furthermore, each document vector equals the sum of the vectors of the terms in that document.

The LSA mechanism for AutoTutor works by calculating the vectors for the student contributions and comparing them to the document vectors for the expected good answers using the cosine metric. Empirical analyses of the corpus size, the number of dimensions, and the thresholds showed that the LSA mechanism performed best with the entire corpus described above, and with 200 dimensions in the LSA space. Figure 1 shows the correlations between the LSA ratings and the average of the human ratings over a variety of cosine match thresholds. The correlation between LSA and the humans approaches that between the human raters. Although a slightly higher correlation was achieved with a 400-dimension LSA space, this increased performance was limited to a single threshold level. This was interpreted as a potential outlier, and the 200 dimension space, with its relatively flat performance across thresholds, was preferred.

3 LSA without SVD

As previously mentioned, LSA has several attributes that may be responsible for its ability to make effective similarity judgments on texts. In addition to the compression/generalization provided by the SVD calculation, LSA might get its benefits from its initial representation of word "meaning" as a vector of the documents that it occurs in. Before the SVD processing, this representation is modified by an information theoretic weighting of the elements, which gives higher weights to terms that appear distinctively in a smaller number of texts, and lower weights to terms that occur frequently across texts. The comparison of texts using the cosine measure on such vectors might also be responsible for such good performance. To test how much discriminative power LSA gains from SVD, we implemented a version of LSA without SVD. This section describes the implementation and evaluation of this mechanism, and relates it to the evaluation of the standard LSA approach.

3.1 Creating the term vectors

To create this model, we started with the documents by terms co-occurrence matrix after the information theoretic weighting and before the SVD processing. We took the columns of this matrix as a representation of the meaning of each term. Because there were over 8000 documents in the corpus and most terms occur in a small number of documents, this is a very sparse representation. Still, it is possible to compare these vectors using the cosine metric. Two terms which occur in exactly the same set of documents would have a cosine of 1.0. Terms which occur in disjoint sets of documents have a cosine of 0. It is also possible with this representation to compute a document vector by adding the vectors of the terms in the document. However, it is not possible to construe the rows in the co-occurrence matrix as the vectors representing document meaning because they have a different rank (the number of terms in the corpus) and

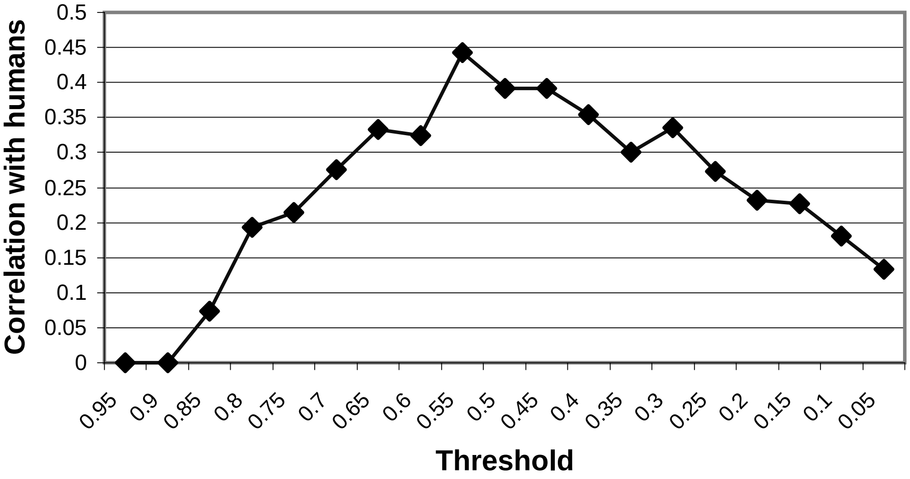

Figure 2: The correlation between LSA without SVD and human raters.

because there is no reason to equate a pattern of term occurrence (the terms are alphabetized in the representation) with a pattern of document occurrence. Thus, we had to calculate vectors not just for the student contributions but for the expected good answer documents as well.

3.2 Evaluation

After these vectors were computed, the evaluation was done in exactly the same way as the evaluation of the full LSA model. Figure 2 shows the correlations between the average of the humans' ratings and the non-SVD model. It is clear that the combination of the distributed, weighted vectors and the geometrical comparisons were sufficient to produce judgments approaching those of the full LSA model. The maximum performance here is $r = 0.43$. As a reminder, the maximum performance of the full LSA model was $r = 0.48$. The maximum performance in this case, however, occurs at just one threshold. For the 200-dimension LSA model, there was fairly stable performance across several thresholds.

4 Keyword matching

Because the performance of the non-SVD algorithm was so close to that of the full LSA implementation, we decided to evaluate a simple keyword-based approach for this task. This section describes the implementation and testing of that approach.

4.1 The matching algorithm

To compare texts with a keyword-matching approach, we used the same segmentation of the student contribution, the same set of expected good answers for each question, and the same set of terms (as keywords) as in the other approaches. We used the same Compatibility measure (Matches / Propositions) that we used for LSA. To determine the extent to which a student contribution speech act S matched an expected good answer E, we defined the keyword match, KM, as follows:

$$KM = \frac{\sum_{t \in S \cap E} w_t}{max(count(S), count(E))}$$

The variable w_t is the weight for a particular term. We tested this keyword approach using both a 1.0 weight for all terms, and also using the information theoretic weight calculated by LSA. The keyword match is essentially the sum of the weights for each keyword that occurs in both the student contribution and the expected good answer, divided by the maximum number of keywords in these two texts. As in the other evaluations, we correlated the performance of the metric at a range of different threshold levels as described in the next section.

4.2 Evaluation

In our first evaluation of the keyword model, we used the same set of thresholds as in the non-SVD evaluation, namely from 0.95 down to 0.05 in 0.05 increments. This resulted in somewhat of a floor effect in the testing however. The LSA weights for terms varied from about 0.3 to 1, but the highest values were only for very rare terms. Thus, most KM values for the weighted approach were relatively low, reaching a maximum of around .35, so we also ran the analysis on a set of thresholds from 0.38 down to 0.02 in 0.02 increments.

Figure 3 shows the correlations with the human ratings for the unweighted keyword model, and both threshold sets for the the weighted model. Note that the threshold labels do not correspond to the actual thresholds for the 0.38 to 0.02 threshold set. The actual thresholds, however, are not important. The general shape of the curve is a fairly clear indicator of the behavior of these models.

The most striking feature of this experiment is the peak correlation of r=0.47 shown by the weighted model at the 0.08 threshold level. This is almost equivalent to the maximum performance of the full LSA model. Similar to the 400-dimension LSA model and the no-SVD model described earlier, however, this point appears to

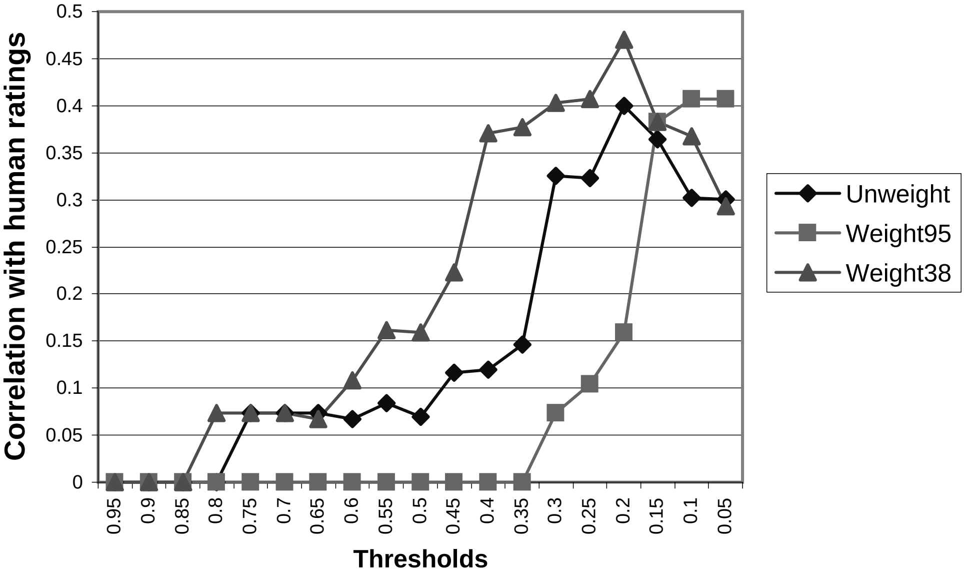

Figure 3: Performance of the keyword matching technique.

be an outlier that would be unlikely to apply across another test set, because it is significantly higher than the neighboring thresholds, which display a fairly flat curve.

We are comfortable in claiming that the simple keyword model can achieve a reliable correlation of $r = 0.40$ with the human raters, with the weighted model showing a relatively flat contour across a range of thresholds. This level of performance is quite close to that shown by the LSA without SVD model, and within about 20% of the performance of the full LSA model. Given the large difference in computational resources required to calculate the keyword approach (the terms and their weights are simply accessed in a hash table), such an approach to text comparison could be beneficial when computational resources are more important than getting the most reliable judgments.

Although the computation of the keyword match was fairly simple, it must be noted that the information theoretic approach used in the weighted keyword model came from the two-textbook corpus that was used for LSA. Collecting this amount of text was a daunting task, but alternative term weights could be calculated from a smaller corpus or from an online lexicographic tool like WordNet [Fellbaum, 1998].

5 Discussion

In this paper we addressed the question of the contribution of the compression step of SVD to LSA, and we compared LSA to a simple keyword-based mechanism in evaluating the quality of student responses in a tutoring task. We showed that although the performance of the full LSA model was superior to the reduced models, these alternatives approached the discriminative power of LSA.[1]

LSA gets its power from a variety of sources: the corpus-based representation of words, the information theoretic weighting, the use of the cosine to calculate distances between texts, and also SVD. SVD should make LSA more robustly able to derive text meaning when synonyms or other similar words are used. This may be reflected by the wider range of thresholds over which LSA performance remains relatively high.

Even though LSA without SVD seems to perform fairly well, it must be noted that the use of SVD results in a very large space and processing time advantage by drastically reducing the size of the representation space. If we took LSA without SVD as the original basis for comparison, and then discovered the advantages of SVD with its ability to “do more with less”, it would clearly be judged superior to the non-SVD LSA model.

It should also be noted that this task is rather difficult for LSA. It has been previously shown that LSA does better when it has more text to work with [Rehder *et al.*, 1998], with relatively low discriminative abilities in the $2 - 60$ word range, and steadily climbing performance for more than 60 words. In fact, other researchers have reported that in short-answer type situations, LSA acts rather like a keyword matching mechanism. It is only with longer texts that LSA really distinguishes itself (Walter Kintsch, personal communication, January, 1999). Because the student texts in this study are relatively short (average length = 18 words), LSA had less information on which to base its judgments, and therefore, its abilities to discriminate were reduced. It is possible that with longer texts there would be more of a

[1] Similar results of a relatively small effect of SVD on a different corpus were reported by Guy Denhière, personal communication, July 1998.

difference between the performance of LSA and the alternative methods presented here. On the other hand, we must also point out that this lack of text seems to have hurt the human raters' abilities to discriminate as well, resulting in fairly low inter-rater reliability scores.

The results presented here do not mitigate the promise of such corpus-based, statistical mechanisms as LSA. They suggest, however, that more research is needed to further tease apart the strengths of the various aspects of such an approach. In future research, we will remove the information theoretic weighting from the non-SVD model to determine how well the system can perform by treating all words as equals.

In conclusion, if you want a text evaluation mechanism based on comparisons, and if you have a good set of texts as a basis of comparison, you have several options. A simple keyword match performs surprisingly well, and is relatively inexpensive computationally. A mechanism like the no-SVD model presented here does not produce better maximum performance than the keyword model on these relatively short texts, but it does produce good performance across a range of thresholds, indicating a robustness to be able to handle a variety of inputs. The full LSA model exceeds both the performance and the robustness of both of these models, achieving results comparable to those of humans with intermediate domain knowledge. Because the initial goal of the AutoTutor project is to simulate a normal human tutor that has no special training but nevertheless produces significant learning gains, we are happy with this level of performance. In future research, we will address the possibility of combining structural analysis of the student texts with LSA's semantic capabilities. This may hold the key to approaching the performance of expert human raters in this task.

Acknowledgments

This work was completed with the help of Katja Wiemer-Hastings, Art Graesser, Roger Kreuz, Lee McCaulley, Bianca Klettke, Tim Brogdon, Melissa Ring, Ashraf Anwar, Myles Bogner, Fergus Nolan, and the other members of the Tutoring Research Group at the University of Memphis: Patrick Chipman, Scotty Craig, Rachel DiPaolo, Stan Franklin, Max Garzon, Barry Gholson, Doug Hacker, Xiangen Hu, Derek Harter, Jim Hoeffner, Jeff Janover, Kristen Link, Johanna Marineau, Bill Marks, Michael Muellenmeister, Brent Olde, Natalie Person, Victoria Pomeroy, Holly Yetman, and Zhaohua Zhang. We also wish to acknowledge very helpful comments on a previous draft by three anonymous reviewers.

References

[DARPA, 1995] DARPA. *Proceedings of the Sixth Message Understanding Conference (MUC-6)*. Morgan Kaufman Publishers, San Francisco, 1995.

[Deerwester *et al.*, 1990] S. Deerwester, 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*, 41:391–407, 1990.

[Fellbaum, 1998] C. Fellbaum. *WordNet: An electronic lexical database*. MIT Press, Cambridge, MA, 1998.

[Foltz *et al.*, 1998] P. W. Foltz, W. Kintsch, and T. K. Landauer. The measurement of textual coherence with latent semantic analysis. *Discourse Processes*, 25:285–308, 1998.

[Graesser *et al.*, 1995] A. C. Graesser, N. K. Person, and J. P. Magliano. Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, 9:359–387, 1995.

[Landauer and Dumais, 1997] 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*, 104:211–240, 1997.

[Landauer *et al.*, 1997] T. K. Landauer, D. Laham, R. Rehder, and M. E. Schreiner. How well can passage meaning be derived without using word order? a comparison of Latent Semantic Analysis and humans. In *Proceedings of the 19th Annual Conference of the Cognitive Science Society*, pages 412–417, Mahwah, NJ, 1997. Erlbaum.

[Putnam, 1987] R. T. Putnam. Structuring and adjusting content for students: A study of live and simulated tutoring of addition. *American Educational Research Journal*, 24:13–48, 1987.

[Rehder *et al.*, 1998] B. Rehder, M. Schreiner, D. Laham, M. Wolfe, T. Landauer, and W. Kintsch. Using Latent Semantic Analysis to assess knowledge: Some technical considerations. *Discourse Processes*, 25:337–354, 1998.

[Wiemer-Hastings *et al.*, 1998] P. Wiemer-Hastings, A. Graesser, D. Harter, and the Tutoring Research Group. The foundations and architecture of AutoTutor. In B. Goettl, H. Halff, C. Redfield, and V. Shute, editors, *Intelligent Tutoring Systems, Proceedings of the 4th International Conference*, pages 334–343, Berlin, 1998. Springer.

[Wiemer-Hastings *et al.*, 1999] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. In *Proceedings of Artificial Intelligence in Education, 1999*, Amsterdam, 1999. IOS Press.

[Wolfe *et al.*, 1998] M. Wolfe, M. E. Schreiner, B. Rehder, D. Laham, P. W. Foltz, W. Kintsch, and T. K. Landauer. Learning from text: Matching readers and texts by Latent Semantic Analysis. *Discourse Processes*, 25:309–336, 1998.