A Revised Algorithm for Latent Semantic Analysis

Xiangen Hu, ZhiqiangCai, Max Louwerse, AndrewOlney, Phanni Penumatsa, Art Graesser, and TRC
Department of Psychology, The University of Memphis, Memphis, TN 38152

Abstract

The intelligent tutoring system AutoTutor uses latent semantic analysis to evaluate student answers to the tutor's questions. By comparing a student's answer to a set of expected answers, the system determines how much information is covered and how to continue the tutorial. Despite the success of LSA in tutoring conversations, the system sometimes has difficulties determining at an early stage whether or not an expectation is covered. A new LSA algorithm significantly improves the precision of AutoTutor's natural language understanding and can be applied to other natural language understanding applications.

1 Introduction

The use of intelligent technology in education is on the rise. Intelligent tutoring systems, once restricted to artificial intelligence labs at major universities, arc migrating to mainstream schools [Koedinger *et al,* 1997]. Intelligent tutoring systems (ITS) in this environment face a difficult challenge: to understand the student and manage the tutoring session in the face of vague orungrammatical input. For most ITSs, language understanding and dialog management are core components. Particularly in a classroom, these systems live or die by their ability to understand what the student is trying to say. One technique, Latent Semantic Analysis has been successfully developed for such purposes [Landauer *et ai,* 1998b; Landauer and Dumais, 1997; Foltz *et ai,* 1998; Graesser *et ai,* 2002]. LSA, a statistical technique utilizing unsupervised learning, is both highly portable to other domains and adept at recognizing vague or incomplete input.

This paper outlines some problems inherent in the traditional LSA algorithm and solutions to this problem by using a different LSA algorithm. Not only does this new algorithm increase the precision of ITS language understanding, but it also offers a new perspective on a commonly used technique in cognitive science, computational linguistics and artificial intelligence.

1.1 Latent Semantic Analysis

Latent Semantic Analysis is a statistical, corpus-based language understanding technique that estimates similarities on a scale of-1 to 1 between the latent semantic structure of terms and texts [Dccrwester et al., 1990]. The input to LS A is a set of corpora segmented into documents. These documents are typically paragraphs or sentences. Mathematical transformations create a large term-document matrix from the input. Forcxample, if there are *m* terms in n documents (usually m and *n* are very large, for now, assume r≥m), then a matrix of.$\mathbf{A} = f_{ij} \times G(j) \times L(i,j)_{m,n}$ is obtained. The value of f_{ij} is a function of the integer that represents the number of times term 7 appears in document j $L(i,j)$ is a local weighting of term 7. in document *j* and $G(i)$ is the global weighting for term v. Such a weighting function is used to differentially treat terms and documents to reflect knowledge that is beyond the collection of the documents. This matrix of *A* has, however, lots of redundant information. Singular value decomposition reduces this noise by linearly decomposing the matrix A into three matrices A = $\mathbf{U}\sum\mathbf{V}^{\mathrm{T}}$; U is m x m and V is $n \times n$ square matrices, such that $\mathbf{U}\mathbf{U}^{\mathrm{T}} = \mathbf{I}_{m \times m}$, $\mathbf{V}\mathbf{V}^{\mathrm{T}} = \mathbf{I}_{n \times n}$, and $\sum$ is m x n diagonal matrix with singular values on the diagonal. By removing dimensions corresponding to small singular values and keeping the dimensions corresponding to larger singular values, the representation of each term is further reduced to a smaller vector witli only *k* dimensions. The new representation for the terms (the reduced U matrix) arc no longer orthogonal, but the advantage of this is that only the most important dimensions that correspond to larger singular values are kept. This method of statistically representing knowledge has proven to be useful in a range of studies. For instance, studies have shown that LSA performs as well as students on TOEFL (test of English as a foreign language) tests lLandauer and Dumais, 1997], that it grades essays as reliably as humans [Landauer *ct al,* 1998a] and that it reliably measures the coherence between a sentence and successive sentences [Foltz *ct ah,* 1998]. Finally, LSA has successfully been used in intelligent tutoring systems like AutoTutor [Gracsser *ct ai,* 2002; 1999].

1.2 AutolWor

AutoTutor is a conversational agent that assists students in actively constructing knowledge by holding a conversation in natural language with them [Graesser *et ai,* 2001], In addition to latent semantic analysis, at least four other com-

ponents can be distinguished: 1) a dialog management system guides the student through the tutor-student conversation. Fuzzy production rules and a Dialog Advancer Network form the basis of these conversational strategics; 2) curriculum scripts organize the pedagogical macrostructure of the tutorial. These scripts keep track of the topic coverage and follow up on any problems the student might have; 3) a talking head with facial expressions and synthesized speech is used for the interface. Parameters of the facial expressions and rudimentary gestures are generated by fuzzy production rules; 4) mixed-initiative dialog, including the appropriate use of discourse markers to make the conversation smoother; a speech act classifier that accounts for the pragmatics of incoming expressions; and a question answering tool that dynamically answers student questions on a variety of topics.

Auto Tutor was originally developed for computer literacy. Over the last two years a web version of AutoTutor was developed that tutors in conceptual physics. In both domains world knowledge is provided by LSA spaces of domain specific text books.

2 An improved LSA algorithm

At the beginning of a session, AutoTutor presents a question to the student, and the student responds to this question. AutoTutor analyzes the accuracy of this answer by comparing the student's answer with a series of expected ideal answers. Over the course of the tutoring session, the student covers each of the expectations. The tutor allows the student to move to the next problem only once all expectations arc covered.

Although possible in theory, a single contribution from a student usually does not cover all expectations at once. Instead, the student simply types one sentence at a time in the conversation with the tutor. Based on the student's responses, the tutor will then provide appropriate feedback based on the quality of responses. To provide feedback to a student's contributions, the tutoring system needs to know the following:

1. information related to the expected answer elements that is 1) new (what was not in the previous contributions); 2) old (what was in the previous contributions)
2. information not related to the expected answer elements that is 1) new (what was not in the previous contributions); 2) old (what was in the previous contributions)

Depending on these four components, AutoTutor chooses the most appropriate feedback. This mechanism for student contributions is illustrated in Table 1. For example, AutoTutor needs to provide highly positive feedback when students provide new relevant information (cell labeled ++). For relevant but repeated information (cell labeled +), AutoTutor needs to provide only some non-negative feedback. For irrelevant contributions, AutoTutor needs to point out the repeated misconceptions, eventually with negative feedback (cell with —). The system returns non-positive feedback in cases of irrelevant information occurring the first time (cell with -).

The challenge for AutoTutor is to obtain information that belongs to each of the cells in the table. That is, the system needs to take into account both the relevance of the information and whether or not the information is new.

Type of feedback	relevant	irrelevant
New	+ +	-
Old	+	

Table 1: Four types of feedback the system provides on the basis of relevance and newness of student contribution.

- Question: Suppose a runner is running in a straight line at constant speed and throws a pumpkin straight up. Where will the pumpkin land? Explain why.
- Expectation: The pumpkin will land in the runner's hands.
- Student contributions:
  - (1) 1 think, correct me if I am wrong, it will not land before or behind the man.
  - (2) The reason is clear, they have the same horizontal speed.
  - (3) The pumpkin will land in the runner's hand.
  - (4) Did I say anything wrong?
  - (5) Come on, I thought I have said that.

	Old LSA	New Infor.	New contribution	New LSA
(1)	0.431	100%	0.431	0.431
(2)	0.430	99%	0.175	0.466
(3)	0.751	88%	0.885	1.0
(4)	0.713	98%	0.000	1.0
(5)	0.667	97%	0.000	1.0

Table 2: Example of student contribution and evaluation based on two LSA methods

In earlier versions of the system, AutoTutor put all the student contributions (from the first response to the most recent response) together as one document and would then compare with the expectation. One of the reasons for this was that it has often been claimed that the best performance in LSA comes from paragraphs rather than sentences (see [Foltz *et al.,* 1998]). However, simple vector algebra shows that vector summation of term-vectors for the combined contributions may in fact reduce the similarity between the expectation and contributions when contributions are added. This reduction is evident in the example given in second column of Table 2. The tutor asks the student a question at the start of a conceptual physics problem. The student's answer is matched with an ideal expected answer. The question now is what happens to the LSA coverage scores if the student submits (new/old) (relevant/irrelevant) multiple contributions.

From the expected answer we know that a student's answer like (1) is almost correct. Now imagine that the student answers (2). The cosine match drops, resulting in AutoTutor asking for more information. Now assume that the student also answers (3), which is the exact ideal answer. Using a traditional vector addition algorithm, the similarity is not 1.0. This loss of precision results from the noise introduced by the irrelevant information in the student's answers. By adding the contributions' vectors, the system cannot distinguish be-

tween the different parts (new/old) (relevant/irrelevant) of the student's contributions. So although LSA effectively compares semantic similarities between two large documents, LSA lacks precision for comparing smaller documents in a progressive sequence. Under a vector-addition model, a student whose answers improve dramatically over the course of the tutoring session is "penalized" for an initial bad answer.
To solve this limitation, we propose an alternative solution. Instead of simply combining contributions into a larger document, each contribution is treated as "independent" in a vector subspace. The combination of the contributions is then not represented as a simple vector summation, but instead as a span in the subspace. This way, the vector for any new contribution can be algebraically decomposed into two components. One component is the projection of the vector to the spanned subspace of the previous contributions, and the other component is perpendicular to the subspace. These two components of the most recent contribution correspond to relevant information (projection to the subspace) and new information (the perpendicular component). Finally, the cosine match of the new information with the expectation is the measure of new (additional) coverage of the expectations. We applied this method to the example such as Table 2 and observed desirable increase in LSA's precision, as illustrated in column 3,4, and 5 of Table 2. The rows of Table 2 present the five student contributions. The column 'New Info' gives the percentage of new information for each of the contributions, compared to the previous contributions. The third column is the relevance of the new contribution to the span. The final two columns give the old and new LSA scores.

Although contribution (3) is identical to the expectation, it still is only 88% new. The reason is that contributions (1) and (2) already contain some of the information from (3). For example, although (2) contains 99% new information, it has only marginally (0.175) contributed as coverage. Notice that the quality in the subsequent student contributions does not deteriorate, but the old LSA values do. The new LSA values, on the other hand, account for additional relevant information, even bringing the coverage score to the maximum value of 1.

The method provided here can be used to compute all four cells in Table 1, because it differentiates whether the information is new or old, and whether it is relevant or not. Furthermore, since it provides information at every step, numerical information of the values can be used to provide secondary information for feedback. For example, the rate of increase in the new LSA algorithm provides us with information on the development student performance on a step-by-stcp basis. By being able to localize LSA scores, AutoTutor can now determine the effectiveness of its dialog moves.

The proposed new algorithm can potentially be used in applications like essay grading, where the student's composition covers the key elements for a given essay. The algorithm can measure development of student performance and can take into account whether information is old or new, relevant or irrelevant.

3 Conclusion

This paper addressed the use of latent semantic analysis in intelligent tutoring systems like AutoTutor. Despite the success of LSA in AutoTutor, previous versions were not able to differentiate between relevant/irrelevant or new/old information in student contributions. Replacing the vector-addition based algorithm with a span-based algorithm does not only improve AutoTutor's evaluation of student contributions, but is most likely to improve LSA performance in a wide range of other natural language understanding applications.

References

iDeerwester *et al.,* 1990] S. Dccrwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society For Information,* pages 391-407, 1990.

[Foltz et a/., 1998] P. W. Foltz, W. Kintsch, and T. K. Landauer The measurement of textual coherence with latent semantic analysis. *Discourse Processes,* pages 285-307, 1998.

[Graesscr *et al,* 1999] A. C. Gracsscr, K. Wiemer-Hastings, P. Wiemer-Hastings, R. Kreuz, and the Tutoring Research Group. AutoTutor: A simulation of a human tutor. *Journal of Cognitive Systems Research,* 1:35—51, 1999.

[Gracsser et al., 2001] A.C. Gracsscr, N. Person, D. Harter, and TRG. Teaching tactics and dialog in AutoTutor. *International Journal of Artificial Intelligence in Education,* 12:257-279,2001.

[Gracsscr *et al,* 2002] A. C. Graesser, X. Hu, B. A. Olde, M. Ventura, A. Olney, M. Louwerse, D. R. Franc esc hetti, and N. Person. Implementing latent semantic analysis in learning environments with conversational agents and tutorial dialog. In *Proceedings of the 24th Annual Meeting of the Cognitive Science Society,* page 37. Mahwah, NJ: Erlbaum, 2002

[Koedinger et al., 19971 K.R. Koedinger, J.R. Anderson, W.H.. Hadley, and M.A. Mark. Intelligent tutoring goes to school in the big city. *International Journal of Artificial Intelligence in Education,* 8:30-43, 1997.

[Landauer and Dumais, 1997] T. K. Landauer and S. T Dumais. A solution to plato's problem: The latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review,* pages 211 240, 1997.

[Landauer *et ai,* 1998a] T. K. Landauer, P. W. Foltz, and D. Laham. Introduction to latent semantic analysis. *Discourse Processes*, pages 259-284, 1998.

lLandauer et al., 1998b] T. K. Landauer, D. Laham, and P. W. Foltz. Learning human-like knowledge by singular value decomposition: A progress report. In M. J. M. 1. Jordan, Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems,* pages 45-51. Cambridge: MIT Press, 1998.