Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis*

Peter Wiemer-Hastings
Katja Wiemer-Hastings
Arthur C. Graesser
The University of Memphis
Department of Psychology
Memphis TN 38152-6400
(PWMRHSTN@MEMPHIS.EDU)

Abstract

AutoTutor is an intelligent tutor that interacts smoothly with the student using natural language dialogue. This type of interaction allows us to extend the domains of tutoring. We are no longer restricted to areas like mathematics and science where interaction with the student can be limited to typing in numbers or selecting possibilities with a button. Others have tried to implement tutors that interact via natural language in the past, but because of the difficulty of understanding language in a wide domain, their best results came when they limited student answers to single words. Our research directly addresses the problem of understanding what the student naturally says. One solution to this problem that has recently emerged is Latent Semantic Analysis (LSA). LSA is a statistical, corpus-based natural language understanding technique that supports similarity comparisons between texts. The success of this technique has been described elsewhere [3, 5, for example]. In this paper, we give an overview of LSA and how it is used in our tutoring system. Then we focus on an important issue for this type of corpus-based natural language analysis, namely, how large must the training corpus be to achieve efficient performance? This paper describes two studies which address this question, and systematically tests the kind of texts needed in the corpus. We discuss the implications of these results for tutoring systems in general.

*This project is supported by grant number SBR 9720314 from the National Science Foundation's Learning and Intelligent Systems Unit, and by the work of the other members of the Tutoring Research Group at the University of Memphis: Ashraf Anwar, Myles Bogner, Patrick Chipman, Scotty Craig, Rachel DiPaolo, Stan Franklin, Max Garzon, Barry Gholson, Doug Hacker, Xiangen Hu, Derek Harter, Jim Hoeffner, Jeff Janover, Bianca Klettke, Roger Kreuz, Kristen Link, Johanna Marineau, Bill Marks, Lee McCaulley, Michael Muellenmeister, Fergus Nolan, Brent Olde, Natalie Person, Victoria Pomeroy, Melissa Ring, Holly Yetman, and Zhaohua Zhang. We wish to thank Peter Foltz and Walter Kintsch for their comments on our use of LSA, and Priti Shah for suggestions on presentation of the results. We also wish to acknowledge very helpful comments on a previous draft by three anonymous reviewers.

1 Introduction

In the past many intelligent tutoring systems have been developed in scientific or mathematical tutoring domains. Topics in such domains can be relatively cleanly defined, with a set of problem-solving exercises and expected answers. This scientific bent also fits in well with the interests of many AI researchers. However, the advantages of this approach come at a cost. First, it confines tutoring domains to a narrow range of topics. Second, such tutoring systems are inflexible in accommodating different and perhaps more efficient modes of learning. Entering numerical answers into computers is just one way of interacting with a tutor. Some education researchers have argued that students learn better when they verbally process the learning material in a tutoring situation [1, for example].

We are using analyses of human-human tutoring situations and a set of new technologies to develop an intelligent tutor that interacts with students through such natural tutorial dialogue. The primary goal of the project is to produce natural interaction, not to increase student learning. Following the educational results cited above, we assume that a cooperative, constructive dialogue will increase learning. A key technological requirement for this project is a tool that robustly understands the students' natural language contributions. A corpus-based, statistical technique called Latent Semantic Analysis (LSA) has recently been used in other text analysis tasks. Its comprehension performance correlates well with human experts. We use LSA to evaluate student contributions and help the tutor decide what dialogue move to perform next. This paper gives a broad overview of LSA, and how it is used in our tutoring system, AutoTutor. We present findings which show that LSA performs comparably with human raters in evaluating the quality of student answers. Our discussions focus on a key issue for such a corpus-based natural language mechanism: the amount of corpus material that is needed to provide adequate performance. Then we address a follow-up issue of how closely that corpus should be related to the tutoring topic.

2 Overview of AutoTutor

To facilitate our description of the language understanding module, we give a general overview of the AutoTutor architecture here. For a more detailed description, see [9].

The basic "protocol" of a tutoring session with AutoTutor is modeled on human tutoring sessions [4, 6]. The tutor asks a question or poses a problem, and collaborates with the student to construct what the tutor judges to be a fairly complete answer to the question. Then the process repeats.

Most human tutors are not highly trained, but are instead peers of the students. Tutors often use simple props or drawings to help their students learn. Tutors do not get very far "into the heads" of students [6]; they typically have only a shallow understanding of what the students say, but can determine whether a response is in the general vicinity of the expected answer. Despite the lack of complete understanding, survey studies have shown a huge advantage for face-to-face tutoring sessions over classroom situations [2].

The user interface to AutoTutor consists of two windows: one for displaying animated or static graphics, and one for the student to type in her replies.[1] There is also a

[1] We will attempt to integrate a speech understanding mechanism in a later stage of the project.

talking head on the screen which speaks AutoTutor's contributions (with moving lips), and gestures to appropriate parts of the graphical display.

AutoTutor's knowledge of its tutoring domain resides in a curriculum script. This is not a script like the proverbial restaurant script or a script in a play, but a static representation of the questions or problems that the tutor is prepared to handle in a tutoring situation [7]. AutoTutor's current curriculum script contains three different topics within our tutoring domain. For each topic, there are 12 different questions, or problem-solving exercises which are graded from easy to hard, based on theoretical analyses of what it will take to completely solve them. For each question or problem there is also: (a) an optional textual or animated information delivery item, (b) a relatively lengthy complete and correct "ideal" answer, (c) that ideal answer broken down into a set of specific good answers which each address one aspect of the ideal answer, (d) a set of additional good answers, (e) a set of bad answers, (f) a set of question that the student would be likely to ask, with appropriate answers, and (g) a succinct summary. For each aspect of the ideal answer there are three additional items to help the student construct that aspect: a hint, a prompt, and an elaboration.

The current tutoring domain for AutoTutor is computer literacy. This is a required class at the University of Memphis, so we have easy access to students on whom we can test the system. Several members of the project have experience teaching this class. Although it may seem to be a contradiction from our stated desire of steering away from a more formal or scientific domain, the class is full of issues like the relative merits of the Macintosh and Windows operating systems, or different approaches to promoting computer security. AutoTutor's curriculum script focuses on such issues and deep reasoning questions.

3 Assessing student answers with LSA

As previously mentioned, LSA is a corpus-based, statistical mechanism. It was originally developed for the task of information retrieval: searching a large database of texts for a small number of texts which satisfy a query. A number of researchers have recently evaluated LSA on other tasks, from taking the TOEFL analogy test, to grading student papers [5, 3]. We give a broad overview LSA of here, and concentrate on its use in AutoTutor. (For more details about LSA, see the recent special issue of Discourse Processes, volume 25, numbers 2 & 3, 1998, on quantitative approaches to semantic knowledge representations.)

The training of LSA starts with a corpus separated into units which we will call texts here. For the AutoTutor corpus, we used the curriculum script, with each item as a separate text for training purposes. The corpus also included a large amount of additional information from textbooks and articles about computer literacy. Each paragraph of this additional information constituted a text. The paragraph is said to be, in general, a good level of granularity for LSA analysis because a paragraph tends to hold a well-developed, coherent idea (Peter Foltz, personal communication, October 1997).

LSA computes a co-occurrence matrix of terms and texts. A "term" for LSA is any word that occurs in more than one text. The cells in this matrix are the number of times a particular term occurs in a particular text. A log entropy weighting is performed on this matrix to emphasize the difference between the frequency of occurrence for a term in a particular text and its frequency of occurrence across texts. Then the matrix is reduced to an arbitrary number of dimensions, K, by a type of principle components analysis called singular value decomposition (SVD). The result is a set of weightings (the singular values, or eigenvalues) and a set of K-long vectors: one for each term, and one for each text.

The normalized sum of the vectors of the terms in any text equals the vector for the text. The distance between any two vectors is conveniently calculated by their geometric cosine. This distance is interpreted as the semantic distance, or similarity, between the terms or texts. A cosine close to 1 indicates high similarity. A cosine of 0 (for an orthogonal vector in the K-dimensional space) indicates low similarity or complete unrelatedness. It appears that the data compression of the SVD forces terms that occur in similar contexts to have similar representations; it is claimed that this contextual co-occurrence carries semantic information.

The training is done in advance of the AutoTutor tutoring sessions. AutoTutor uses the results of the training to evaluate student responses in the following way: A vector for the student contribution is calculated by summing the vectors of the terms included in the contribution. This vector is compared with the text vectors of some of the curriculum script items for the current topic. In particular, AutoTutor calculates a general goodness and badness rating by comparing the student contribution with the set of good and bad answers in the curriculum script for the current topic. More importantly, it compares the student response to the particular good answers that cover the aspects of the ideal answer. We calculate two measures with this comparison:

- **Completeness**: the percentage of the aspects of the ideal answer for the current topic which "match" the student response
- **Compatibility**: the percentage of the student response (broken down into speech acts) that "match" some aspect of the ideal answer

A "match" is defined as a cosine between the response vector and the text vector above a critical threshold. By comparing human ratings of these same measures with these LSA ratings computed with a variety of thresholds and dimensionalities (Ks) we can empirically determine which settings work best for a given task and corpus. As described in [10], such an evaluation showed that a threshold of .55 with a 200-dimensional space correlated highest with the average ratings of four human raters ($r = 0.49$). Two human raters with intermediate knowledge of computer literacy correlated with each other $r = 0.51$.[2] Because we are starting the project by attempting to model untrained human tutors (who produce excellent learning gains), we are quite happy with this level of performance.

A third variable that affects the performance of LSA in such a task is the size of the training corpus. This issue is of great practical significance for others wishing to create such a corpus-based natural language understanding mechanism. The remainder of this paper describes our exploration of this issue.

[2] The correlations reported here are for the Compatibility metric defined above. Two domain experts correlated on this metric r=0.78. The correlations between LSA and humans were lower for the Completeness score because of differences in the way the non-LSA portion of the measure was computed.

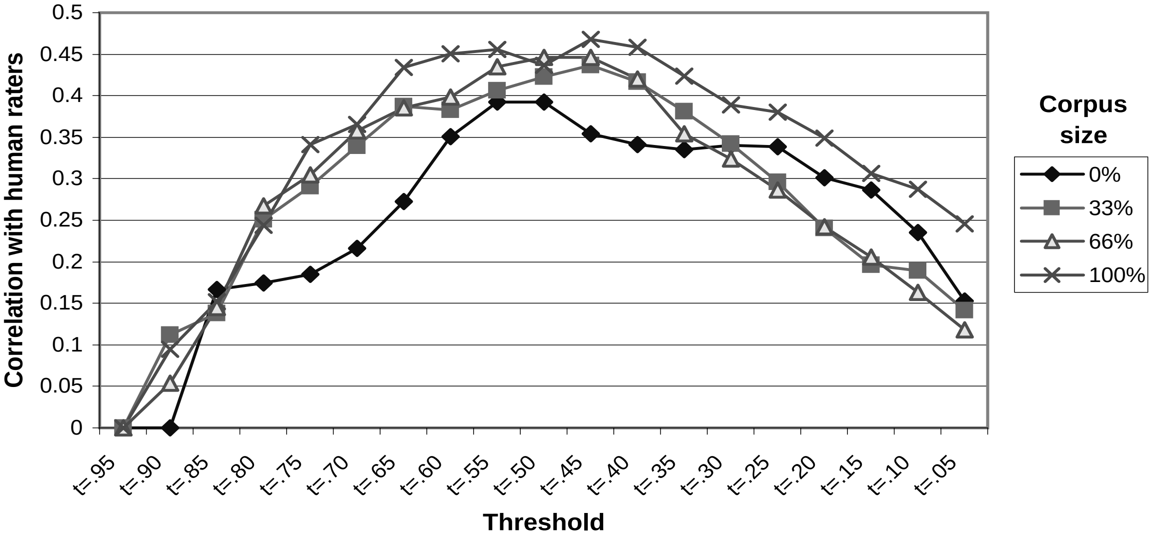

Figure 1: Evaluation performance by different sized corpora

4 How much corpus is enough?

In order to evaluate the contribution of the size of the corpus to LSA's performance, we randomly excised items from the supplemental corpus (i.e. the textbook material). It was necessary to keep the curriculum script items in the corpus in order to evaluate the metrics, but they account for only 15% of the entire 2.3 MB corpus. The supplementary corpus was split into two parts: The "specific" subcorpus deals with the tutoring topics: computer hardware, software and the internet. The "general" subcorpus covers other areas of computer literacy. The specific and general subcorpora accounted for 47% and 38% of the total corpus respectively. We tested 4 different amounts of corpus and maintained the balance between specific and general text by randomly removing none, 1/3, 2/3, or all of each of the specific and general subcorpora. The ideal balance between specific and general text is discussed below. Because the size of the corpus could affect the dimensionality and threshold, we tested the performance with a 4x3x19 design, with four levels of corpus size, 3 different dimensionalities (200, 300, and 400) that had previously performed well, and 19 critical threshold values, from 0.05 to 0.95 in 0.05 increments. For each combination of these factors, we tested LSA's correlation with the ratings of the human raters.

We performed a multivariate analysis of variance (MANOVA) on these data, with correlation between the LSA rating and average human rating as the dependent variable and corpus size, dimensionality, and threshold value as predictors. We obtained main effects for amount of text (significant at the .01 level), number of dimensions (significant at the 0.05 level), and threshold value (significant at the .01 level). There were also significant interactions between the size of the corpus and thresholds, and between dimensions and thresholds. Figure 1 plots performance for each level of corpus size by threshold, averaged across the different levels of dimensions. As expected, LSA's performance with the entire corpus was best, both in terms of the maximum correlation with the human raters and in terms of the width of the threshold value range in which it performs well. One surprising result is the negligible difference between the 1/3 and 2/3 corpora (the two lines with intermediate performance in the middle thresholds). Clearly there is not a linear relation between the amount of text and the performance of LSA. Another surprise was the relatively high performance of the corpus without

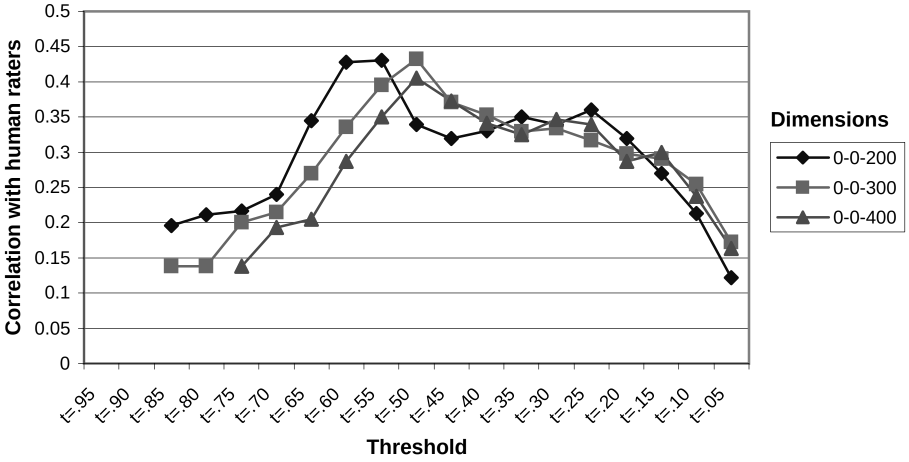

Figure 2: Performance with no additional corpus materials

any of the supplemental items, that is, with the curriculum script items alone. This demonstrates that a relatively small set of items that are closely relevant to the given task can produce acceptable performance with LSA.

Figure 2 shows the performance of the curriculum-script-only corpus for the three different dimensionalities. There is an interaction between the number of dimensions and the threshold values: at lower dimensionalities, LSA performs better with higher thresholds.[3] In addition, it shows that LSA achieves the best performance with this corpus with the 200 and 300 dimensional spaces, achieving a maximum correlation of $r = 0.43$ with the human raters. This is almost as high as the maximum correlation we obtained for the entire corpus ($r = .49$).

5 Does the LSA corpus need specific or general text?

In our first experiment, we kept the original balance between the amount of domain specific and domain general text. We originally came to this balance in an effort to give LSA a "well-rounded education". We were advised against using a very general corpus like an encyclopedia which others have used for other tasks [5] because it would dilute the knowledge base by flooding it with terms which it would never encounter in the tutoring domain (Peter Foltz, personal communication, October 1997). We did want to include a range of texts from within computer literacy so that the student could bring in other technical terms that were not strictly within the confines of the tutoring topics (hardware, software, and the internet). We collected all the text from two computer literacy textbooks, and supplemented the chapters on the tutoring topics with 10 additional articles or book chapters about each of those three topics.

We did a further manipulation of the training corpus to address the question of which ratio is best between domain specific and domain general texts. We used the same 4 partitions of each subcorpus as in the previous experiment, but this time combined the parts in each of the 16 possible ways. We again compared the performance of each resulting LSA space with three different dimensionalities and 19 different threshold

[3] At the highest threshold values, no student contribution ever exceeded the threshold, so no correlations with human ratings could be computed.

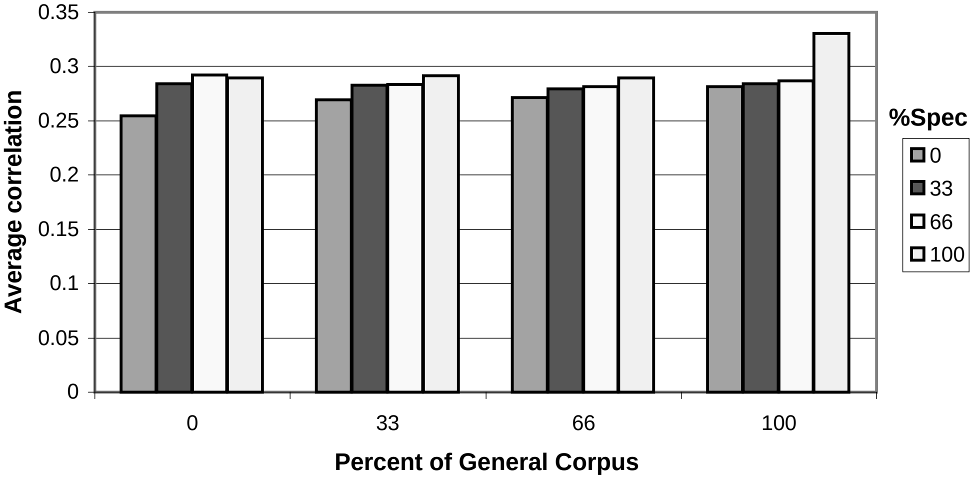

Figure 3: Effects of varying ratios of specific and general corpora

values. A MANOVA of these data showed main effects of the size of the specific corpus, the size of the general corpus, and of the threshold values (all significant at the 0.01 level). There were interaction effects between specific and general corpus size, specific corpus and thresholds, general corpus and dimensions, general corpus and thresholds, and dimensions and thresholds (the interaction between the general corpus and the dimensions was significant at the 0.05 level, all others at the 0.01 level).

Figure 3 shows the relationships obtained between the specific corpus size and the general corpus size, with the performance averaged across the number of dimensions. Each group of bars represents one level of general corpus. Within the groups, the specific corpus varied. A full line graph of all of these data is similar to that shown in figure 1. The best and worst performance were again produced by the full corpus and the curriculum-script-only corpus. All of the other corpora were crowded in between. Figure 3 again shows the non-linearity of performance that was apparent in the first experiment. The smallest and largest corpora perform significantly better and worse than the others. But the other levels of corpus size and balance are almost indistinguishable. These findings support the general notion that more of the right kind of text is better for LSA. But it also suggests that empirical testing of the corpora is still necessary. A smaller corpus takes less time to train, less storage space, and less processing time for comparisons. Thus, if there is no significant performance advantage with larger corpora, they can be avoided.

6 Discussion

An intelligent tutoring system which can interact with a student using natural language promises many advantages over traditional systems in both the range of tutoring domains and tutoring styles available, and may lead to better learning. A critical technology for such a system is a natural language processing mechanism that can robustly understand student input, but this goal has been elusive for decades. LSA provides such a mechanism that performs at the level of human raters with intermediate domain knowledge. We think it will allow us to create an intelligent tutor that simulates a human tutor.

Our analyses of different sizes of training material showed monotonic but not linear increases in LSA performance. This supports the general benefit of increasing the size of the training corpus of relevant text. Our manipulation of the balance between general and specific texts seems to support our initial hypothesis that an approximately equal balance or one slightly favoring the specific texts is advantageous to LSA.

In regard to the generality of the findings reported here, it has been noted in previous work that there is a positive correlation between the length of a text and the ability of LSA to accurately judge its quality [8]. For grading essays, LSA produced the most reliable grades when the length of the (manipulated) text was above 200 words. In the tutoring task, the length of our student contributions is significantly smaller, averaging 16 words. It is likely that this limits the level of performance of the LSA mechanism. However, it is also likely that short texts are more difficult for humans to assess, as evidenced by the correspondingly low correlations between our intermediate human raters.

Fill-in-the-blank interfaces can lead students to a guess-and-test approach to a tutoring situation. We hope that by pushing students to construct full natural language answers to questions, they will learn better. Our analyses of the performance of LSA suggest that it can provide the same level of distinction as that shown by untrained human tutors, and thereby better support the learning process.

References

[1] M. T. H. Chi, N. de Leeuw, M. Chiu, and C. LaVancher. Eliciting self-explanations improves understanding. *Cognitive Science*, 18:439–477, 1994.

[2] P. A. Cohen, J. A. Kulik, and C. C. Kulik. Educational outcomes of tutoring: A meta-analysis of findings. *American Educational Research Journal*, 19:237–248, 1982.

[3] P.W. Foltz. Latent Semantic Analysis for text-based research. *Behavior Research Methods, Instruments, and Computers*, 28:197–202, 1996.

[4] 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.

[5] 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.

[6] N. K. Person. *An analysis of the examples that tutors generate during naturalistic one-to-one tutoring sessions.* PhD thesis, University of Memphis, Memphis, TN, 1994.

[7] 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.

[8] 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.

[9] P. Wiemer-Hastings, A. Graesser, D. Harter, and the TRG. 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.

[10] P. Wiemer-Hastings, K. Wiemer-Hastings, and A. Graesser. Evaluating student answers in a tutoring session with Latent Semantic Analysis. In preparation.