Using Latent Semantic Analysis to Evaluate the Contributions of Students in AutoTutor

Arthur C. Graesser, Peter Wiemer-Hastings, Katja Wiemer-Hastings, Derek Harter,

Natalie Person*, and the Tutoring Research Group

The University of Memphis

* Rhodes College

Send correspondence to:

Arthur C. Graesser
Department of Psychology
The University of Memphis
CAMPUS BOX 526400
Memphis, TN 38152-6400
(901) 678-2742
fax: 901-678-2579
a-graesser@memphis.edu

Submittion for Special Issue of Interacting Learning Environments,
guest edited by Joseph Psotka

running head: LSA in AutoTutor

keywords: computer literacy, latent semantic analysis, question answering,
tutorial dialog, tutoring

Abstract

AutoTutor is a fully automated computer tutor that assists students in learning about hardware, operating systems, and the Internet in an introductory computer literacy course. AutoTutor presents questions and problems from a curriculum script, attempts to comprehend learner contributions that are entered by keyboard, formulates dialog moves that are sensitive to the learner's contributions (such as prompts, elaborations, corrections, and hints), and delivers the dialog moves with a talking head. Latent semantic analysis (LSA) is a major component of the mechanism that evaluates the quality of student contributions in the tutorial dialog. LSA's evaluations of college students' answers to deep reasoning questions are equivalent to the evaluations provided by intermediate experts of computer literacy, but not as high as more accomplished experts in computer science. LSA is capable of discriminating different classes of student ability (good, vague, erroneous, versus mute students) and in tracking the quality of contributions in tutorial dialog.

Using Latent Semantic Analysis to Evaluate the Contributions of Students in AutoTutor

The Tutoring Research Group at the University of Memphis has been developing a fully automated computer tutor, called AutoTutor (Graesser, Franklin, Wiemer-Hastings, and the TRG, 1998; Wiemer-Hastings, Graesser, Harter, & the TRG, 1998). AutoTutor attempts to comprehend student contributions and to simulate dialog moves of human tutors. AutoTutor is currently simulating tutorial dialog moves of normal (unskilled) tutors, but eventually we hope to incorporate sophisticated ideal tutoring strategies. AutoTutor is currently being developed for college students who take an introductory course in computer literacy. These students learn the fundamentals of computer hardware, the operating system, and the Internet.

A snapshot of AutoTutor in action should be sufficient to convey the nature of AutoTutor. There is a talking head that serves as a conversation partner with the learner. The talking head delivers AutoTutor's dialog moves with synthesized speech, appropriate intonation, facial expressions, and gestures. At the top of the screen, AutoTutor prints the questions and problems that are produced from a curriculum script. These questions and problems invite lengthy responses and deep reasoning (e.g., answers to why, how, what-if), as opposed to being fill-in-the blank questions or shallow questions. There is a multi-turn tutorial dialog between AutoTutor and the learner during the course of answering a question (or solving a problem). The learner types in his/her contributions during the exchange by keyboard. For some topics, there are graphical displays and animation, with components that AutoTutor points to. We want AutoTutor to be a good conversation partner that comprehends, speaks, points, and displays emotions, all in a coordinated fashion. However, other questions and problems do not have a graphical display and animation, as is the case in normal tutoring. From the standpoint of the present article, the focus will be on AutoTutor's ability to extract information and comprehend the text that students type in the keyboard. It is the text comprehension component where the use of latent semantic analysis (LSA) is particularly relevant.

Why would we want to simulate a normal, unskilled human tutor? Available research has already documented that human tutoring is a very effective method of learning and instruction. Human-to-human tutoring enhances learning by .4 to 2.3 standard deviation units compared to classroom controls and other suitable controls (Cohen, Kulik, & Kulik, 1982; Bloom, 1984). Interestingly, human tutors are extremely effective even though over 90% percent of the tutors in actual school systems are untrained; that is, they have moderate domain knowledge and no training in tutoring skills. They are peer tutors, cross-age tutors, or paraprofessionals, but rarely accomplished professionals. The notion that untrained tutors are extremely effective in promoting learning is quite counterintuitive, if not provocative. What is the magic behind normal unskilled tutoring?

This paradox prompted us, several years ago, to investigate what real tutors do that is so effective. We analyzed videotapes of approximately 100 hours of untrained tutors in naturalistic tutoring sessions (Graesser & Person, 1994; Graesser, Person, & Magliano, 1995; Person & Graesser, 1999; Person, Graesser, Magliano, & Kreuz, 1994; Person, Kreuz, Zwaan, & Graesser, 1995). Our analyses lead us to the general conclusion that there is something about interactive discourse that is responsible for learning gains. Our research revealed that it not the case that learners are active question-askers who take command of the tutorial agenda; it is the tutor who sets the agenda and introduces most of the topics, questions, and problems. In the sample of tutoring sessions that we examined, the tutor set 100% of the agenda in a tutoring session, introduced 93% of the topics, presented 82% of the examples, and asked 80% of the questions. We found that the human tutors and learners have a remarkably incomplete understanding of each other's knowledge base and that many of each other's contributions are not deeply understood. It is not fine-tuned "student modeling" that is important, but rather a tutor that serves as a conversation partner when common ground is minimal. This finding is compatible with tests of the ANIMATE tutor developed by Nathan, Kintsch, and Young (1992); ANIMATE produced impressive learning gains on algebra word problems, but did not construct a detailed map of what the student knows. Most human tutors have only an approximate assessment of the quality of student contributions and of the major chunks of knowledge that are covered in the session.

Our anatomy of normal tutoring sessions revealed that normal unskilled tutors do not use most of the ideal tutoring strategies that have been identified in education and the intelligent tutoring system enterprise. These strategies include the Socratic method (Collins, 1985), modeling-scaffolding-fading (Collins, Brown, & Newman, 1989), reciprocal training (Palincsar & Brown, 1984), anchored learning (Bransford, Goldman, & Vye, 1991), error diagnosis and correction (van Lehn, 1990; Lesgold, Lajoie, Bunzo, & Eggan, 1992), frontier learning, building on prerequisites (Gagne, 1977), and sophisticated motivational techniques (Lepper, Aspinwall, Mumme, & Chabay, 1990). Detailed discourse analyses have been performed on samples of these sophisticated tutoring strategies (Fox, 1993; Hume, Michael, Rovick, & Evens, 1996; McArthur, Stasz, & Zmuidzinas, 1990; Merrill, Reiser, Ranney, & Trafton, 1992; Moore, 1995; Putnam, 1987), but these sophisticated tutoring strategies were practicially nonexistent in the unskilled tutoring sessions that we videotaped and analyzed (Graesser et al., 1995). Our plan is to have later versions of AutoTutor incorporate some of these ideal tutoring strategies, so there will be a hybrid of unskilled strategies and ideal strategies. But our initial goal was to simulate the dialog moves of the normal unskilled tutor, which are known to be very effective (Cohen et al., 1982).

We discovered that a key feature of effective tutoring lies in generating dialog moves that assist learners in the active construction of explanations, elaborations, and mental models of the material. Other researchers have also proposed that active constructions of explanations are critical for learning, and have a greater impact than merely presenting information to learners (Anderson, Corbett, Koedinger, & Pelletier, 1995; Bransford, Goldman, & Vye, 1991; Chi, Bassok, Lewis, Reimann, & Glaser, 1989; Chi, de Leeuw, Chiu, & La Vancher, 1994; Moore, 1995). Human tutors assist this construction of knowledge by delivering collaborative discourse moves that encourage learners to built explanations and that fill in the holes that learners fail to provide.

AutoTutor simulates the normal tutor's attempt to collaboratively construct answers to questions, explanations, and solutions to problems. It does this by formulating dialog moves that assist the learner in an active construction of knowledge, as the learner attempts to answer the questions and solve the problems posed by the tutor. Thus, AutoTutor serves as a discourse prosthesis, drawing out what the learner knows and scaffolding the learner to an enhanced level of mastery. The categories of dialog moves in AutoTutor are presented below.

(1) Positive immediate feedback. "That's right" "Yeah"

(2) Neutral immediate feedback. "Okay" "Uh-huh"

(3) Negative immediate feedback. "Not quite" "No"

(4) Pumping for more information. "Uh-huh" "What else"

(5) Prompting for specific information. "The primary memories of the CPU are ROM and _____"

(6) Hinting. "The hard disk can be used for storage" or “What about the hard disk?”

(7) Elaborating. “CD ROM is another storage medium.”

(8) Splicing in correct content after a student error. This is a correction.

(9) Requestioning. "So once again, what is the function of a CPU?"

(10) Summarizing. "So to recap," <succinct recap of answer to question>

Ideally, AutoTutor will produce dialog moves that fit the conversational context, that are sensitive to the learner's abilities, that have pedagogical value, and that follow the norms of polite conversation.

It is beyond the scope of this article to present all of the details of the mechanisms of AutoTutor (Graesser et al., 1998; Wiemer-Hastings et al., 1998). Instead, we plan on accomplish two objectives. The first section provides an overview of AutoTutor, with a particular emphasis on the role of LSA. It should be noted that AutoTutor is a working system, not mere vaperware. The second section presents data on how well LSA performs when it is used to evaluate the contributions of students during question answering and problem solving.

How does AutoTutor Work? The Seven Modules of AutoTutor

AutoTutor has seven modules which are briefly described in this section. The modules include a curriculum script, language extraction, speech act classification, latent semantic analysis, topic selection, dialog move generation, and a talking head.

(1) **Curriculum Script**. A curriculum script is a loosely ordered, but well-defined set of skills, concepts, example problems, and question-answer units (McArthur et al., 1990; Putnam, 1987). Most tutors follow a script-like macrostructure, but deviate from the structure when the student manifests difficulties, misconceptions, and errors. AutoTutor similarly has a curriculum script that organizes the tutorial dialog. It contains didactic descriptions, tutor-posed questions, cases, problems, figures, and diagrams (along with anticipated good responses to each topic). AutoTutor's curriculum script for computer literacy includes three macrotopics: hardware, the operating system, and the Internet. The three macrotopics follow an order that parallels the computer literacy course and the textbook (Beekman, 1997). There are 12 topics within each macrotopic. Within each set of 12 topics, 3 levels of difficulty are crossed with 4 topic formats.

The three levels of difficulty (easy, medium, difficult) map onto taxonomies of cognitive difficulty and question difficulty (Bloom, 1956; Graesser & Person, 1994; Wakefield, 1996). Medium questions tap causal networks, plan-driven procedures, and logical justifications (e.g., Why or how does X occur?, What if X occurs?, What are the consequences of X?), whereas easy questions tap lists of components and properties of components (e.g., What are the components or properties of X?). Difficult questions required the comparison, synthesis, or integration of disparate ideas, as in the case of analogical reasoning or the application of knowledge to a real world problem.

The four topic formats are: (1) Question+Answer, (2) Didactic-information +Question+Answer, (3) Graphic-display+Question+Answer, and (4) Problem+Solution. Each topic format includes a main, focal question that is presented to the learner. The Question+Answer format simply asks a question, with no didactic content leading up to it (e.g., "Why do computers need operating systems?"). The Didactic-information+Question +Answer format starts out with didactic content and then asks a question related to that content. The Graphic-display+Question+Answer format presents pictorial information and then asks a question that refers to the pictorial information. The Problem+Solution format presents a short problem scenario that learner is asked to solve; a question directly captures what the focal problem is.

The Answer or Solution content that is associated with each topic includes a number of data slots that specify anticipated responses. The content that students type into the keyboard will ultimately be matched to the content within these slots, and LSA is used in these pattern matching operations. For example, presented below is an example question and a number of good answers (i.e., aspects of a complete answer) that would be expected by AutoTutor.

QUESTION: Why do computers need operating systems?

GOOD-ANSWER-1: The operating system helps load application programs.

GOOD-ANSWER-2: The operating system coordinates communications between the software and the peripherals.

GOOD-ANSWER-3: The operating system allows communication between the user and the hardware.

GOOD-ANSWER-4: The operating system helps the computer hardware run efficiently.

An ideal complete answer consists of a set of N good answers or aspects, {A_1, A_2,…A_N}. All of these aspects need to be covered in the tutorial dialog after the question is asked. The learner's keyboard input is matched to each aspect and all possible combinations of aspects. When considering all 36 questions, each topic had 3 to 9 aspects that comprised the ideal complete answer. It should be noted that these anticipated good answers are stored in English, as opposed to LISP code or other structured code. Therefore, the curriculum script can be easily authored by a teacher or other individual who is not an expert programmer. One of the salient benefits of using LSA for the pattern match operations is that the content of curriculum script can be written in English descriptions.

There are additional data slots associated with each topic, over an above the ideal complete answer. There is a list of anticipated bad answers, corresponding to misconceptions and bugs that need correction. When the learner's input has a high enough LSA match to one of the bad answers (B_i), AutoTutor produces a dialog move that corrects the misconception or bug (i.e., a correction is spliced in). Thus, there is a corrective splice associated with each anticipated bad answer. Associated with each good answer aspect (A_i), there are different articulation formats that correspond to three different dialog move categories: elaborations, hints, and prompts. For example, the hint format for A_1 would be “What about application programs?”, whereas the prompt format would be “The operating system helps load application ______”. The hints and prompts are designed to get the learner to contribute information, whereas the information is merely delivered by AutoTutor in the case of dialog moves that are elaborations, e.g., “The operating system helps load application programs.” There are other data slots within each topic, but the slots discussed are sufficient for the present article that focuses on LSA.

(2) **Language Extraction**. Language modules analyze the words in the messages that the learner types into the keyboard during a particular conversational turn. There is a large lexicon with approximately 10,000 words. Each lexical entry specifies its alternative syntactic classes and frequency of usage in the English language. For example, “program” is either a noun, verb, or adjective. Each word that the learner enters is matched to the appropriate entry in the lexicon in order to fetch the alternative syntactic classes and word frequency values. There is also an LSA vector for each word. AutoTutor is capable of segmenting the input into a sequence of words and punctuation marks with 99%+ accuracy, of assigning alternative syntactic classes to words with 97% accuracy, and of assigning the correct syntactic class to a word (based on context) with 93% accuracy (Olde, Hoeffner, Chipman, Graesser, & the TRG, 1999). A neural network assigns the correct syntactic class to word W, taking into consideration the syntactic classes of the preceding word (W-1) and subsequent word (W+1).

(3) **Speech Act Classification**. A neural network is used to segment and classify the learner's content within a turn into speech acts. There are five speech act categories: Assertion, WH-question, YES/NO question, Directive, and Short Response. AutoTutor can currently classify 89% of the speech acts correctly, using a neural network architecture. It is the content of the learner's Assertions that is used to assess the quality of learner contributions. The Assertions are therefore most relevant to the present analysis of LSA. The short feedback that AutoTutor delivers after a conversational turn (i.e., positive, negative, neutral) is sensitive AutoTutor's evaluation of the quality of the set of Assertions within a conversational turn. Suppose there is a high LSA match between the set of Assertions within turn T and one or more of the good answer aspects, $\{A_1, A_2, \ldots A_N\}$, associated with the topic. AutoTutor would present positive short feedback at the beginning of its next turn. Suppose that there is a high LSA match between the set of Assertions within turn T and one of the anticipated bad answers, B_i. In this case, AutoTutor would present some sort of negative short feedback at the beginning of its next turn, and splice in the correction. When the Assertions of turn T have modest or low matches with the good answer aspects or the bad answers, then various forms and degrees of neutral short feedback are delivered by AutoTutor.

AutoTutor has a different strategy for dealing with the other speech act categories: WH-question, YES/NO question, Directive, and Short Response. These strategies, which are needed for a smooth mixed-initiative dialog, will not be addressed in the present article.

(4) **Latent Semantic Analysis** (LSA). The knowledge about computer literacy is represented by LSA (Landauer & Dumais, 1997; Landauer, Foltz, & Laham, 1998). LSA is a statistical technique that compresses a large corpus texts into a space of 100 to 500 dimensions. The K-dimensional space is used when evaluating the relevance or similarity between any two bags of words, X and Y. The relevance or similarity value varies from 0 to 1; in most applications of LSA, a geometric cosine is used to evaluate the match between the K-dimensional vector for one bag of words and the vector for the other bag of words. From the present standpoint, one bag of words is the set of Assertions within turn T. The other bag of words is the content of the curriculum script associated with a particular topic, i.e., good answer aspects and the bad answers. LSA has had remarkable success in capturing the world knowledge that is needed grading essays of students (Foltz, 1996) and in matching texts to students of varying abilities to optimize learning (Wolfe, Schreiner, Rehder, Laham, Foltz, Kintsch, & Landauer, 1998). Kintsch's construction-integration model of text comprehension has incorporated LSA as a major component in its knowledge construction phases (Kintsch, 1998). LSA has been quite successful in evaluating the quality of learner Assertions in AutoTutor's domain of computer literacy (Wiemer-Hastings, Wiemer-Hastings, Graesser, and the TRG, 1999), as will be discussed later.

The LSA space for the domain of computer literacy was based on two books on computer literacy, approximately 30 articles that focus on hardware, operating systems, and the Internet, and on the curriculum script. An LSA analyses requires the preparation of a document by word (D x W) co-occurrence matrix. Each cell in the matrix specifies the number of occurrences of word W_i in Document D_j. The computation of the K dimensions is then derived from a statistical procedure called singular value decomposition. In order to prepare the DxW matrix, the researcher needs to define what constitutes a document unit. A single document was defined as a paragraph in the case of the textbooks and 30 articles. In the case of the curriculum script, a single document was defined as a good answer aspect. When we performed the LSA on the 2.3 MB corpus of documents, the solution that we adopted had 200 dimensions.

(5) **Topic Selection**. Topics are selected by fuzzy production rules that are sensitive to the structure of the curriculum and to the learner's ability (i.e., knowledge about the topic, as manifested in the exchanges during previous topics). The topics within the hardware macrotopic are selected before the operating system topics, which in turn are covered before the Internet. Within each macrotopic, the difficulty level of the topic is matched to the student ability according to the zone of proximal development (see also Wolfe et al., 1998). That is, the good student will receive difficult topics whereas the underachieving student will receive easy topics.

Student ability is based on the learner's Assertions in all of the previous learner turns in the tutorial session. An Assertion quality score is computed for the set of Assertions in any given learner turn, T_i, which we denote as $Q(T_i)$. Part of the computation of the quality score is based on its resonance with the ideal good answer of the current topic. This consists of the highest LSA cosine match between T_i and any particular good answer aspect (A_j) associated with the current topic, or any combination of aspects in the ideal good answer, $\{A_1, A_2, \ldots A_N\}$. The highest number of aspects per topic is 9, so the most matches that would ever be made in this computation is $[2^9 - 1]$ = 511 matches. The maximum cosine match score among the 511 comparisons would serve as an index of how well T_i resonates with the ideal complete answer. However, the Assertion quality score also needs to be penalized for matches to bad answers. Therefore, formula 1 better captures the Assertion quality score.

$$Q(T_i) = [\mathit{max}\{\text{cosine}(T_i, \text{aspect combination})\} - \mathit{max}\{\text{cosine}(T_i, \text{bad answer } B_j)\}] \quad (1)$$

The computation of student ability is simply the mean of the Assertion quality scores for all of the previous learner turns in the tutorial session, as denoted in formula 2.

$$\text{Student ability} = \Sigma\, Q(T_i) \qquad (2)$$

It should be noted that we have adopted other metrics of student ability and Assertion quality scores, but they are quite similar to those expressed in formulae 1 and 2.

(6) **Dialog Move Generator**. After the learner types in the content of his turn, AutoTutor needs to generate dialog moves. Sometimes AutoTutor answers a question asked by the learner. But student questions are not particularly frequent speech acts in tutoring (Graesser & Person, 1994). More often, AutoTutor responds to and builds on the Assertions of the learner as the two parties collaboratively answer questions or solve problems. In this sense AutoTutor serves as a discourse prosthesis, drawing out what the learner knows and scaffolding the learner to an enhanced level of mastery. AutoTutor normally delivers immediate short feedback about the quality of the learner's Assertions in the preceding turn (i.e., positive, neutral, versus negative feedback), followed by a

lengthier substantive contribution that prompts the learner for more information, that adds information, or that corrects a student error. The more substantive contributions include pumps, prompts, hints, elaborations, splices, requestions, and summaries, as we listed earlier. Thus, AutoTutor normally returns two dialog moves within a turn, such as neutral immediate feedback followed by a hint.

The categories of the substantive dialog moves during AutoTutor's turns are determined by a set of fuzzy production rules. These rules are tuned to (a) the quality of the student's Assertions in the preceding turn, as computed by LSA, (b) global parameters that refer to the ability, verbosity, and initiative of the student, and (c) the extent to which the content of the topic has been covered. For example, two production rules for producing a hint are:

IF [(student ability = MEDIUM or HIGH) & ($Q(T_i) = LOW$)]

THEN [select HINT]

IF [(student ability = LOW) & (student verbosity = HIGH) & ($Q(T_i) = LOW$)]

THEN [select HINT]

The content of the hint is determined by an algorithm that captures the zone of proximal development. That is, the hint tries to drag out information (specified in the curriculum script) that slightly extends the boundaries of what the student knows or what has been covered in the topic. Stated differently, AutoTutor selects the next good answer aspect to focus on. So how, more specifically, does AutoTutor select the next aspect? AutoTutor keeps track of the extent to which each aspect (A_i) has been covered as the dialog evolves for a topic. The coverage metric varies from 0 to 1 and gets updated as each Assertion is produced by the tutor or learner. LSA is used to compute the extent to which the various Assertions cover the particular aspects associated with a topic. If some threshold (t) is met or exceeded, then the aspect A_i is considered covered. Our analyses revealed that a threshold value between .5 and .7 was a reasonable value for considering an aspect covered (Wiemer-Hastings et al., 1999). AutoTutor selects, as the next aspect to cover, that aspect that has the highest subthreshold coverage score. Therefore, AutoTutor builds on the fringes of what is known. A topic is finished when all of the aspects have coverage values that meet or exceed the threshold t. This zone of proximal development is only one mechanism for determining the next good answer to cover. Other considerations involve temporal order (X precedes Y in time) and logical prerequisite (X is a prerequisite for Y). However, the exact algorithm for selecting the next dialog move is not the primary focus of the present article.

(7) **Talking Head with Gestures**. Most of the tutor's dialog moves are delivered by a talking head that synchronizes synthesized speech, facial expressions, and gestures. Microsoft Agent was used as the talking head with syntesized speech. The facial expressions and intonation in the immediate short feedback are sensitive to the quality of the Assertions in the student's most recent turn. The parameters of the facial expressions and intonation are generated by fuzzy production rules (McCauley, Gholson, Hu, Graesser, and the TRG, 1998). The intonation and facial expressions are constrained by both politeness and pedagogy. For example, when a learner's Assertions are highly error ridden, it is impolite to say "No, your wrong" because that is a face-threatening act that has the potential of lowering learner output. A better way is to say "Okay" verbally, but

have a pause or an intonation that signifies skepticism. The verbal code delivers politeness whereas the intonation delivers pedagogy. The mechanisms of the talking head are beyond the scope of the present article, however.

Evaluation of LSA in AutoTutor

We performed some evaluations that assessed how well our LSA representation in AutoTutor can accurately assess the quality of learner Assertions. Three sets of analyses are reported in this section. First, we will summarize some results of a study that was reported by Wiemer-Hastings et al. (1999). That study evaluated 192 answers to the questions in the curriculum script; these answers were generated by approximately 100 students in the computer literacy course, but not in the context of the tutorial dialog of AutoTutor. Second, we report some analyses that assess the extent to which AutoTutor can accurately discriminate students with different abilities, based on their Assertions in the tutorial dialog. Third, we will report analyses that assess whether AutoTutor can accurately track the coverage of the ideal complete answer during the tutorial interaction within a topic.

Anaylsis of 192 Answers to Questions about Computer Literacy

Wiemer-Hastings et al. (1999) analyzed how well the LSA space on computer literacy could accurately evaluate the answers to questions in the curriculum script. A sample of the 36 questions in the curriculum script was presented to students enrolled in the computer literacy course at the University of Memphis. The college students answered these questions by typing in their answers into a web cite facility. That is, they simply answered these open-class questions without the benefit of AutoTutor. The data were collected after the college students had read the relevant chapters in the book and had received a lecture on each macrotopic (i.e., harware, operating system, Internet). The underlying research issue is whether the quality of these answers to the questions could be accurately evaluated by the LSA space of AutoTutor.

We needed a gold standard in order to evaluate the validity of AutoTutor's LSA component. Four trained experts rated the 192 answers to the questions. Two of these raters were intermediate experts. They were graduate students in cognitive science who had read the Beekman text on computer literacy and had three or more years of active experience with computer technologies. The other two raters were accomplished experts. There had a graduate degree in computer science and also read the Beekman text. Each of the four experts rated the 192 answers on a scale of answer quality. The quality rating was in the form of a compatibility percentage. Specifically, they judged the percentage of Assertions in the answer that matched some aspect of the ideal complete answer. Thus, these experts had the list of the good answer aspects for each topic in addition to the student's answer to the topic. If there were 4 Assertions in the student's answer, and 3 of them matched one of the aspects of the ideal complete answer, then a score of .75 would be assigned as the answer quality score. The LSA quality score was computed in an analogous fashion. We computed the proportion of Assertions in the student answer that matched one particular aspect, or some combination of aspects, by a value that met or exceeded some threshold. As it turns out, a threshold value of .55 yielded the best results in an analysis that tested all possible thresholds and that kept track of the threshold which had the highest correlation with the experts' quality ratings. In summary, the Wiemer-Hastings et al. (1999) study scaled the 192 student answers on different measures of answer quality. There was an LSA quality score and a quality score provided by each of 4 human experts.

The results of correlational analyses were quite encouraging for advocates of LSA. The correlation between LSA's answer quality scores and the mean quality scores of the four experts was .49. This .49 correlation is indistinguishable from the .51 correlation between the ratings of the two intermediate experts, but significantly lower than the .78 correlation between two accomplished experts. It appears, therefore, that the LSA space of AutoTutor exhibits the performance of an intermediate expert, but not an accomplished expert. It should be noted that the vast majority of human tutors have intermediate expertise, rather than accomplished expertise, so the LSA space does an excellent job simulating the normal unskilled human tutor.

Wiemer-Hastings et l. (1999) performed a number of auxiliary analyses which are potentially informative to LSA researchers. The K parameter was varied between sizes of 100 and 500 dimensions, in 100 dimension increments. There was an increase in performance (i.e., correlation with human quality scores) between 100 and 200 dimensions, but performance did not significantly differ between 200 and 500 dimensions. The threshold parameter (t) was varied between .05 and .95; as mentioned above, the peak performance was at t = .55, although there was very little fluctuation between threshold values of .50 and .70. An ablation analysis was performed to assess the impact of the training corpus of texts. The curriculum script was always included in these ablation analyses. The documents in the curriculum script account for 15% of the 2.3

MB corpus. The other 85% included the 2 textbooks and the 30 articles on computer literacy. The ablation analyses were conducted by randomly removing 0%, 33%, 67%, versus 100% of the text documents and rerunning the LSA solutions. We were surprised to learn that the LSA produced rather impressive performance even when 100% of the texts were removed. The curriculum script alone produced a correlation of .39, which compares favorably with the .49 correlation with 100% of the text corpus. There is an emerging adage in the LSA community that says "the more text, the better." This adage is undoubtedly quite correct, but we would like to add another adage: "a modest amount of text about a particular topic is pretty damn good."

Evaluating Student Ability

If the LSA component of AutoTutor is worth its weight in gold, it should be able to discriminate the ability of students (i.e., knowledge about computer literacy) from the Assertions that the students contribute in the tutorial dialog. Person et al. (1994) has reported that the Assertions of students in one-on-one tutoring is much more diagnostic of student ability than is the quality of student questions and the students' metacognitive perceptions on how well they understand.

In order to test AutoTutor's capacity to discriminate student ability, we created different classes of virtual students. That is, we had a virtual tutor (namely AutoTutor) participate in an interactive dialog with virtual students. If AutoTutor is functioning properly, then its student ability scores should reflect the quality of the Assertions of the virtual students. The following virtual students were created for each of the 36 topics in the curriculum script:

(1) Good verbose student. The first 5 turns of the virtual student had 2 or 3 Assertions that human experts had rated as good Assertions. These Assertions were sampled from the 100 college students' answers to the 36 questions. The student is regarded as verbose because the student has 2 or 3 Assertions within one turn, which is more than the average number of Assertions per turn in human tutoring.

(2) Good succinct student. The first 5 turns of the virtual student had 1 Assertion that human experts had rated as a good Assertion.

(3) Vague student. The first 5 turns of the virtual student had an Assertion that had been rated as vague (neither good nor bad) by the human experts.

(4) Erroneous student. The first 5 turns of the virtual student had an Assertion that contained a misconception or bug according to human experts.

(5) Mute student. The first 5 turns of the virtual student had semantically depleted

content, such as "Well", "Okay", "I see", and "Uh".

We expected that these classes of virtual students would produce very different values of student ability in AutoTutor.

Table 1 presents mean LSA values that AutoTutor computed when evaluating the quality of student Assertions. Table 1 includes mean LSA matches to good aspects versus bad answers as a function the five classes of virtual students. A difference of .07 between these means should be regarded as statistically significant when determining whether 2 means are different. A number of conclusions can be made on the bases of the data reported in Table 1. The most general conclusion is that AutoTutor is quite discriminating in identifying different classes of students. The LSA matches to good answer aspects follow the following ordering among the virtual students: Good verbose > good succinct > vague = erroneous > mute. The LSA matches to bad answers follow the following ordering: Erroneous > good verbose > good succinct = vague > mute. When we apply the Assertion quality score computed in formula 1, the scores are .19, .23, .08, -.22, and 0 for the good verbose, good succinct, vague, erroneous, and mute students, respectively. These profile of scores are capable of discriminating the students of different ability.

Table 1

LSA Assertion Quality Scores as a Function of Different Classes of Virtual Students

	Assertions Match:	
Class of Virtual Student	Good answer aspects	Bad answers
Good Verbose Student	.79	.60
Good Succinct Student	.72	.47
Vague Student	.58	.50
Erroneous Student	.53	.75
Mute Student	.03	.03

Another interesting outcome is the impact of verbosity on the LSA scores. The good verbose students had higher matches to the good answer aspects than did the good succinct students. But at the same time, the verbose students had higher matches to the bad answers. Thus, the more a student says, the higher the likelihood that some stretch of text will match a bad answer. It should be noted that the difference score reflected in the Assertion quality metric in formula 1 is not influenced by text length; this may be an advantage of using such a score.

Evaluating the Coverage of the Ideal Complete Answer

If AutoTutor's LSA component is operating in a sensible fashion, then the coverage of the ideal complete answer should increase as the tutorial dialog within the topic proceeds. As each turn is taken by the tutor and student, the LSA coverage scores are updated for each good answer aspect, A_i. The coverage score for the ideal complete answer is the proportion of aspects that meet or exceed the threshold ($t = .55$). This coverage score should increase as a function of turns within a topic.

We created a Monte Carlo virtual student in order to test this prediction. The Monte Carlo student has a combination of good answers, erroneous answers, and vague answers that matches the distribution of contributions of actual students in one-on-one tutoring. This distribution is based on the tutoring transcripts analyzed by Person et al. (1994). Thus, the Assertions of the Monte Carlo virtual student were randomly selected from a pool of good answer aspects, erroneous answers, and vague answers for a topic, with percentages that match the distribution in human tutoring. These Assertions of the virtual student were fed into AutoTutor in order to simulate a virtual conversation. We then recorded the coverage scores of the ideal complete answers as a function of the turns in the tutorial dialog.

As expected, the coverage scores showed a monotonic increase as a function of the first 6 learner turns in the tutorial dialog. The means were .27, .39, .45, .62, .66, and .76 for turn 1, 2, 3, 4, 5, and 6, respectively. The increase was of course statistically significant when an analysis of variance was performed. Therefore, AutoTutor's LSA component does an excellent job tracking the coverage of an ideal complete answer during tutorial dialog.

Final Comments

Our evaluation of AutoTutor is encouraging news for those who advocate the use of LSA in representing world knowledge. AutoTutor's LSA component was capable of evaluating the quality of student Assertions as well as intermediate experts, of discriminating different classes of virtual students, and of tracking the coverage of an ideal answer during tutorial dialog. Such evaluations are important because they determine the topics that AutoTutor selects, the immediate short feedback of AutoTutor, the categories of AutoTutor's dialog moves, and the aspects of an ideal answer that AutoTutor selects to focus on next. The LSA component is clearly a critical module of AutoTutor.

There are additional ways that LSA is used in AutoTutor, but these have not yet been evaluated or implemented. LSA will be used when the learner asks questions. Consider the case of YES/NO questions, such as "Isn't RAM primary memory?". The proposition being queried ("RAM is primary memory") is matched, via LSA, to the good answers in the entire curriculum script, and also to the bad answers. If there is a high

LSA match to a good aspect, then AutoTutor will answer YES. The answer will be NO if there is a high match to a bad answer. Otherwise, the answer of AutoTutor is indecisive (“maybe”, “sometimes”, “perhaps”). In the case of WH-questions, LSA is used to match the queried proposition to anticipated questions. For example, in the case of definition questions (“What does X mean?”), X is matched to the entries in a glossary and AutoTutor produces the definition if there is a high match.

LSA is also used to provide fine-grained reactions of the talking head. There are different degrees of positive immediate feedback, depending on how high the LSA match is to the learner’s Assertions in a turn. An extremely high match or a high increment in topic coverage produces a vigorous, enthusiastic nod. A small incremental gain in LSA coverage will produce more subtle positive feedback or “positive neutral” feedback. And then there are mixed evaluations, where the information in a turn matches both good aspects and bad answers. One can imagine an immediate feedback with mixed facial and intonational cues. Similarly, AutoTutor can be augmented to produce backchannel feedback that acknowledges learner contributions, as in the case of head nods and “uh-huh”. The intensity and intonation of the back channel feedback should be sensitive to the quality of student contributions in a turn. Indeed, we suspect that LSA will have an impact on virtually every module of AutoTutor, just as world knowledge is inextricably bound to virtually all modules of discourse comprehension and production.

References

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. The Journal of the Learning Sciences, 4, 167-207.

Beekman, G. (1997). Computer confluence. New York: Benjamin/Cummings.

Bloom, B. S. (1984). The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. Educational Researcher, 13, 4-16.

Bloom, B. S. (1956). Taxonomy of educational objectives: The classification of educational goals. Handbook I: Cognitive domain. New York: McKay.

Bransford, J. D., Goldman, S. R., & Vye, N. J. (1991). Making a difference in people's ability to think: Reflections on a decade of work and some hopes for the future. In R. J. Sternberg & L. Okagaki (Eds.), Influences on children (pp. 147-180). Hillsdale, NJ: Erlbaum.

Chi, M. T., Bassok, M., Lewis, M. W., Reimann, P., & Glaser, R. (1989). Self-explanations: How students study and use examples in learning to solve problems. Cognitive Science, 13, 145-182.

Chi, M. T. H., de Leeuw, N., Chiu, M., & LaVancher, C. (1994). Eliciting self-explanations improves understanding. Cognitive Science, 18, 439-477.

Cohen, P. A., Kulik, J. A., & Kulik, C. C. (1982). Educational outcomes of tutoring: A meta-analysis of findings. American Educational Research Journal, 19, 237-248.

Collins, A. (1985). Teaching reasoning skills. In S.F. Chipman, J.W. Segal, & R. Glaser (Eds), Thinking and learning skills (vol. 2, pp 579-586). Hillsdale, NJ: Erlbaum.

Collins, A., Brown, J. S., & Newman, S. E. (1989). Cognitive apprenticeship: Teaching the craft of reading, writing, and mathematics. In L. B. Resnick (Ed.), Knowing, learning, and instruction: Essays in honor of Robert Glaser (pp. 453-494). Hillsdale, NJ: Erlbaum.

Foltz, P.W. (1996). Latent semantic analysis for text-based research. Behavior Research Methods, Instruments, and Computers, 28, 197-202.

Fox, B. (1993). The human tutorial dialog project. Hillsdale, NJ: Erlbaum.

Gagné, R. M. (1977). The conditions of learning (3rd ed.). New York: Holdt, Rinehart, & Winston.

Graesser, A.C., Franklin, S., & Wiemer-Hastings, P. and the TRG (1998). Simulating smooth tutorial dialog with pedagogical value. Proceedings of the American Association for Artificial Intelligence (pp. 163-167). Menlo Park, CA: AAAI Press.

Graesser, A.C., & Person, N.K. (1994). Question asking during tutoring. American Educational Research Journal, 31, 104-137.

Graesser, A.C., Person, N.K., & Magliano, J.P. (1995). Collaborative dialog patterns in naturalistic one-on-one tutoring. Applied Cognitive Psychology, 9, 359-387.

Hume, G. D., Michael, J.A., Rovick, A., & Evens, M. W. (1996). Hinting as a tactic in one-on-one tutoring. The Journal of the Learning Sciences, 5, 23-47.

Kintsch, W. (1998). Comprehension: A paradigm for cognition. Cambridge, MA: Cambridge University Press.

Landauer, T.K., & Dumais, S.T. (1997). A solution to Plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. Psychological Review.

Landauer, T.K., Foltz, P.W., Laham, D. (1998). An introduction to latent semantic analysis. Discourse Processes, 25, 259-284.

Lepper, M. R., Aspinwall, L. G., Mumme, D. L., & Chabay, R. W. (1990). Self-perception and social-perception processes in tutoring: Subtle social control strategies of expert tutors. In J. M. Olson & M. P. Zanna (Eds.), Self-inference processes: The Ontario symposium (pp. 217-237). Hillsdale, NJ: Erlbaum.

Lesgold, A., Lajoie, S., Bunzo, M., & Eggan, G. (1992). SHERLOCK: A coached practice environment for an electronics troubleshooting job. In J. H. Larkin & R. W. Chabay (Eds.), Computer-assisted instruction and intelligent tutoring systems (pp. 201-238). Hillsdale, NJ: Erlbaum.

McArthur, D., Stasz, C., & Zmuidzinas, M. (1990). Tutoring techniques in algebra. Cognition and Instruction, 7, 197-244.

McCauley, L., Gholson, B., Hu, X., Graesser, A.C., and the Tutoring Research Group (1998). Delivering smooth tutorial dialog using a talking head. Proceedings of the Workshop on Embodied Conversation Characters (pp. 31-38). Tahoe City, CA: AAAI and ACM.

Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems. The Journal of the Learning Sciences, 2, 277-305.

Moore, J.D. (1995). Participating in explanatory dialogues. Cambridge, MA: MIT Press.

Nathan, M.J., Kintsch, W., & Young, E. (1992). A theory of word algebra problem comprehension and its implications for the design of learning environments. Cognition & Instruction, 9, 329-389.

Olde, B.A., Hoeffner, J., Chipman, P., Graesser, A.C., and the Tutoring Research Group (1999). A connectionist model for part of speech tagging. Proceedings of the American Association for Artificial Intelligence (pp. 172-176). Menlo Park, CA: AAAI Press.

Palinscar, A. S., & Brown, A. (1984). Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. Cognition & Instruction, 1, 117-175.

Person, N.K, & Graesser, A.C. (1999). Evolution of discourse in cross-age tutoring. In A.M. O'Donnell and A. King (Eds.), Cognitive perspectives on peer learning (pp. 69-86). Mahwah, NJ: Erlbaum.

Person, N. K., Graesser, A. C., Magliano, J. P., & Kreuz, R. J. (1994). Inferring what the student knows in one-to-one tutoring: The role of student questions and answers. Learning and Individual Differences, 6, 205-219.

Person, N. K., Kreuz, R. J., Zwaan, R., & Graesser, A. C. (1995). Pragmatics and pedagogy: Conversational rules and politeness strategies may inhibit effective tutoring. Cognition and Instruction, 13, 161-188.

Putnam, R. T. (1987). Structuring and adjusting content for students: A study of live and simulated tutoring of addition. American Educational Research Journal, 24, 13-48.

VanLehn, K. (1990). Mind bugs: The origins of procedural misconceptions. Cambridge, MA: MIT Press.

Wakefield, J.F. (1996). Educational psychology: Learning to be a problem solver. Boston: Houghton Mifflin.

Wiemer-Hastings, P., Graesser, A.C., Harter, D., and the Tutoring Research Group (1998). The foundations and architecture of AutoTutor. Proceedings of the 4th International Conference on Intelligent Tutoring Systems (pp. 334-343). Berlin, Germany: Springer-Verlag.

Wiemer-Hastings, P., Wiemer-Hastings, K., and Graesser, A. (1999). Improving an intelligent tutor's comprehension of students with Latent Semantic Analysis. Artificial Intelligence in Education (pp. 535-542). Amsterdam: IOS Press.

Wolfe, M.B.W., Schreiner, M.E., Rehder, B., Laham, D., Foltz, P.W., Kintsch, W., Landauer, T.K. (1998). Learning from text: Matching readers and texts by latent semantic analysis. Discourse Processes, 25, 309-336.

Author Notes

This research was funded by the National Science Foundation (SBR 9720314) in a grant awarded to the first author of this manuscript. The following members of the Tutoring Research Group at the University of Memphis conducted research on this project: Ashraf Anwar, Myles Bogner, Tim Brogdon, Patrick Chipman, Scotty Craig, Rachel DiPaolo, Stan Franklin, Max Garzon, Barry Gholson, Art Graesser, Doug Hacker, Peggy Halde, Derek Harter, Jim Hoeffner, Xiangen Hu, Jeff Janover, Bianca Klettke, Roger Kreuz, Kristen Link, William Marks, Johanna Marineau, Lee McCaulley, Brent Olde, Natalie Person, Victoria Pomeroy, Mat Weeks, Shannon Whitten, Katja Wiemer-Hastings, Peter Wiemer-Hastings, Shonijie Yang, Holly Yetman, and Zhaohua Zhang. Requests for reprints should be sent to Art Graesser, Department of Psychology, Campus Box 526400, The University of Memphis, Memphis, TN 38152-6400, a-graesser@memphis.edu.