The Unreasonable Effectiveness of AutoTutor

Andrew M. Olney

University of Memphis

Abstract

AutoTutor is an educational technology that tutors students by holding a conversation with them. In various studies, AutoTutor has been as effective at helping students learn as human tutors, even though AutoTutor uses relatively shallow artificial intelligence. This chapter explores the behavior of AutoTutor in light of several theories that help explain its unreasonable effectiveness for promoting deep learning.

Keywords: AutoTutor, tutoring, ICAP hypothesis, scaffolding, testing effect

The Unreasonable Effectiveness of AutoTutor

Intelligent tutoring systems (ITS) use artificial intelligence to emulate both the behavior and effectiveness of human tutors (Graesser, Conley, & Olney, 2011). As such,

ITS are perhaps the most effective educational technology of the modern era. Not only are ITS twice as effective at promoting learning that the previous generation of computer-based instruction, but they are also generally as effective as human tutors (Kulik & Fletcher, 2016). However, while it is impractical to provide students with individual human tutors, intelligent tutoring systems, by virtue of being computer programs, can be duplicated without cost and can provide round the clock tutoring.

AutoTutor is an ITS that tutors students by having a conversation with them (Nye, Graesser, & Hu, 2014). Various studies have shown that AutoTutor is particularly effective for deep learning, such as learning causal models (Graesser et al., 2004), and that AutoTutor is just as effective as human tutors (VanLehn et al., 2007). AutoTutor's success is somewhat surprising in that it uses fairly shallow artificial intelligence: AutoTutor does not use a production system, Bayes net, reinforcement learning, parsing, or comparable advanced artificial intelligence techniques that are commonly used in ITS (Woolf, 2008).

This chapter reviews AutoTutor's structure, artificial intelligence, and adaptivity in light of several theories that help explain its unreasonable effectiveness for promoting deep learning.

Human Tutors Facilitate Deep Comprehension

AutoTutor's conversational tutoring was designed to emulate the authentic behaviors and strategies of novice human tutors (Graesser, Person, & Magliano, 1995; Olney, Graesser, & Person, 2010; Person, Graesser, Magliano, & Kreuz, 1994). Perhaps unsurprisingly, the behaviors and strategies of novice tutors are relatively simple but effective. During a tutoring session, novice tutors present one problem after another to the student in a script-like fashion.

For each problem, a novice tutor approximately follows what Graesser and colleagues refer to as the 5-step tutoring frame:

1. Tutor poses the problem
2. Student attempts to answer
3. Tutor provides brief evaluation and feedback
4. Student and tutor have a **multi-turn dialogue** to improve the answer
5. Tutor assesses whether student understands the answer

The multi-turn dialogue of step 4 has variable length depending on how quickly and completely the student voices the correct answer. The turns within step 4 have been described by Graesser and colleagues as Expectation and Misconception Tailored dialogue.
The term *expectation* refers to the components of the expected (correct) answer. Typically, if the correct answer to the problem is an explanation composed of several sentences, then each sentence corresponds to an expectation.

During step 4, the novice tutor addresses the expectations that were omitted from the student's initial answer in step 2. Each expectation is addressed by an alternating pattern of tutor and student utterances. Often the tutor asks the students various kinds of leading questions and gives feedback to student answers. However, novice tutors will also provide examples, request clarifications, rearticulate solutions, and comment on the ability of the student or difficulty of the problem.

A Computerized Tutor: AutoTutor AutoTutor is modeled on the behavior and strategies of novice human tutors but also represents a fairly extreme simplification of those behaviors and strategies. Whereas the novice tutoring research that informed AutoTutor defined 34 tutor

"dialogue moves," the minimal AutoTutor implementation[1] consists of just nine of these: problem statement, pump, hint, prompt, assertion, positive/neutral/negative feedback, and summary (Olney et al., 2010). In particular, the complexity of step 4 of the 5-step tutoring frame is, in AutoTutor, simplified to a hint-prompt-assertion strategy for each expectation. AutoTutor begins step 4 by first deciding which expectation to cover first. Any expectation already articulated by the student is excluded from consideration, and if all expectations have already been articulated, AutoTutor skips step 4 entirely. AutoTutor determines expectation coverage using a statistical technique called Latent Semantic Analysis (LSA; Landauer, McNamara, Dennis, & Kintsch, 2007). Simply stated, LSA can compare student answers to expectations by first converting both answers and expectations into vectors then comparing the angle between the two vectors. An angle of zero indicates dissimilarity, and an angle of 1 indicates perfect similarity. In principle, a variety of methods could be used for the same purpose, and more sophisticated AI techniques have been developed (Rus, Olney, Foltz, & Hu, 2017). The advantage of LSA is that it is *unsupervised*, meaning that it does not require labeled data that is commonly needed for many machine learning methods. The disadvantage of LSA is that it is not very precise and cannot distinguish between antonyms or words in different orders, e.g. "gravity causes mass" and "mass causes gravity" would have a cosine of one. AutoTutor uses an author-selected threshold, e.g. .7, such that LSA similarity below the threshold means the student answer did not cover the expectation and similarity above the threshold means that the expectation has been covered. Previous research has suggested that if the threshold is carefully tuned using expert judgments, the maximum correlation between LSA in AutoTutor and expert judgments is approximately .50 (Olde, Franceschetti, Karnavat, & Graesser, 2002).

[1] Over a 20-year period, many versions of AutoTutor have been created (see Nye et al., 2014, for a review), but allowing for minor variations, this minimal AutoTutor implementation describes all one-on-one versions of AutoTutor.

At the beginning of step 4, AutoTutor uses LSA to assess what expectations have been covered and rank orders the remaining, uncovered expectations using three criteria: redundancy, similarity, and closest to threshold. The redundancy criterion uses LSA to compare uncovered expectations to each other to determine which expectation, if covered next, would cover the most other expectations. The similarity criterion uses LSA to calculate the expectation most similar to the current expectation. The closest the threshold criterion sorts the uncovered expectations according to their LSA similarity to the student's answer. The expectation covered next is selected by summing the ranks determined by these three criteria and choosing the expectation with the lowest summed rank. It should be noted that while AutoTutor's expectation selection algorithm appears highly adaptive because it considers these three criteria dynamically, only the closest to threshold criterion is truly adaptive to the student's answer. The other two criteria are not adaptive because they only consider LSA similarity between expectations, which are static. Because AutoTutor's expectation selection algorithm uses the sum of these three criteria and two of them are non-adaptive, the sequence of expectations covered by AutoTutor is biased towards a canonical non-adaptive order. It should be emphasized, however, that AutoTutor compares student answers to all expectations rather than just the current expectation and that this allows expectations to be covered at any time. Thus, while the order of expectations in AutoTutor is marginally adaptive, the expectations a given student actually sees is adaptive based on their answers.

Once an expectation has been selected, AutoTutor uses a hint-prompt-assertion sequence to elicit an answer from the student that will cover the expectation. A hint in AutoTutor is a leading question as opposed to a prompt, which queries a specific word or phrase in the expectation. For the expectation "The force of gravity pulls the balls downward," a hint is "How does the Earth's gravity affect objects?" whereas a prompt is "Gravity pulls objects in a direction that is _____?" An assertion is simply the expectation or a paraphrase of the expectation. Thus when AutoTutor uses the hint-prompt-assertion strategy, it asks the

student questions about an expectation starting with the least specific hint, then if the expectation still isn't covered in the student's answer, AutoTutor asks a more specific prompt, and finally if the student's answer is below threshold, AutoTutor asserts the correct answer. The hint-prompt-assertion cycle is adaptive in two ways. First, AutoTutor provides feedback after each student response to a hint or prompt using LSA or keyword matching to hint/prompt answers. Second, at any time the current expectation is covered, AutoTutor will terminate the hint-prompt-assertion sequence and select the next expectation.

There are a few additional kinds of dialogue used by AutoTutor beyond what has been described above, namely metacommunicative/metacognitive responses and question answering. Metacommunicative statements like "Please repeat that" will cause AutoTutor to repeat the last utterance, metacognitive statements like "I don't understand" will cause AutoTutor to encourage the student to try, e.g. "Why don't you tell me what you know, and we'll go from there," and questions (in some versions of AutoTutor) cause AutoTutor to respond with answers created with question classification and information retrieval techniques. While these tutoring behaviors, particularly question answering, require more advanced AI to handle, none these are directly tied to AutoTutor's content-based pedagogy, and previous studies indicate that only 3% of student contributions fall into these categories. Accordingly, it is doubtful that they contribute much to AutoTutor's effectiveness, and they will be excluded from further discussion.

What Causes Deep Comprehension?

Why are tutoring and intelligent tutoring systems, like AutoTutor, so effective at promoting learning? Over the past few decades, researchers have progressed from a philosophical, descriptive understanding to an understanding more deeply connected to basic cognitive processes and mechanisms. While a chronological discussion of these developments

has its merits, given the space constraints of this chapter, a more focused approach will be followed, one that begins with broad theories and successively narrows down to what are likely the crucial elements that make AutoTutor effective. The key theories and constructs I discuss are the Interactive-Constructive-Active-Passive hypothesis, scaffolding, and the testing effect.

Interactive-Constructive-Active-Passive

The Interactive-Constructive-Active-Passive hypothesis (ICAP; Chi, 2009) predicts that the type of learning activity (as defined by the overt behaviors of the student) will largely determine learning outcomes. *Passive* activities are those which only include attending, e.g. listening or watching. A prototypical passive activity is attending a lecture. *Active* activities involve both attending and doing something that requires additional attention and movement but is not cognitively demanding. What is, or is not, cognitively demanding is relative to the student in question and therefore is assumed to follow developmental norms. For example, a middle-school student underlining while reading is, by underlining, engaging in an active activity, but a college student typing verbatim notes during a lecture might also be considered to be engaging in an active activity if typing is a highly automated skill. *Constructive* activities involve the creation of a tangible product, such as the solution to a math problem or notes that summarize and interpret a lecture. *Interactive* activities are co-constructive such that two students are jointly engaged in a constructive problem, meaning that they take turns. The ICAP hypothesis is that the rank order of the effectiveness of these activities is $I \geq C \geq A \geq P$.

Support for ICAP stems from two sources. In the original proposal, Chi (2009) reanalyzed 15 previously published studies. Conditions in each study were classified to one of the four ICAP activity types, and the significant differences between the coded conditions were evaluated with respect to the ICAP hypothesis. The ICAP pattern of effectiveness was consistently found. A similar reanalysis approach for 40 previously published studies also found the ICAP pattern (Chi & Wylie, 2014). Follow-up experiments explicitly designed to test ICAP

have also found the predicted pattern of effectiveness. Menekse, Stump, Krause, and Chi (2013) performed both classroom and laboratory experiments of ICAP. In the classroom study, which did not include a passive condition or random assignment, they found that $I = C \geq A$. In the laboratory study, which did include a passive condition and random assignment, they found $I \geq C \geq A \geq P$. Wiggins, Eddy, Grunspan, and Crowe (2017) in a quasi-experimental, within-subjects design tested whether interactive activities had better learning outcomes than constructive activities in a classroom setting. Results indicated that interactive activities made students 24% more likely to get questions correct on the posttest than constructive activities, supporting the ICAP hypothesis.

While Chi and colleagues have made some connections to cognitive processes in order to explain the ICAP effect, detailed accounts have been elusive, perhaps because ICAP theoretically covers any learning task, and learning tasks themselves involve many cognitive processes. However, a model called ICAP-A has been proposed that explains the ICAP effect in terms of attention (Olney, Risko, D'Mello, & Graesser, 2015). According to the ICAP-A model, failures of attention occur from failures of proactive and reactive control of sequential action. Proactive control is strong when tasks are routinized and when tasks are novel and require top-down control. Reactive control is strong when current action is in conflict with task goals, i.e. when errors are made. Passive tasks involve no activity, therefore no opportunities for proactive control or reactive control. Active tasks are simple and routinized, meaning that they have strong proactive control but weaker reactive control because routinized tasks require minimal cognitive monitoring. Constructive tasks may contain routinized subsequences but combine these in novel ways, meaning that they have both stronger proactive control and stronger reactive control than active tasks due to the increased top-down control and monitoring required for novel tasks. Interactive tasks require each participant to allocate attentional control for their individual constructive portion of the task as well as attentional control to monitor their partner's behavior. By other-monitoring, i.e. monitoring their partner's

errors as well as their own, participants experience stronger reactive control than they would if they were engaged in the constructive tasks by themselves. Olney et al. (2015) analyzed previous studies with conditions that correspond to the four ICAP activity types and found that differences in attention in these conditions followed the ICAP order. Additionally, Mills, D'Mello, Bosch, and Olney (2015) investigated the ICAP-A hypothesis in the context of various learning activities in the Guru ITS. Results from that study indicate that mind-wandering (a momentary lapse in attention) was negatively related to learning and that mind-wandering occurred more often in active activities than constructive activities, as ICAP-A would predict (See D'Mello, this volume).

Scaffolding

If interactive activities lead to the best learning outcomes, as proposed by ICAP, then what are prototypical interactive activities? Perhaps unsurprisingly, tutoring can be interactive according to ICAP, but only if it is of a certain form (Chi, 2009; Chi, Siler, Jeong, Yamauchi, & Hausmann, 2001; Chi & Wylie, 2014). If a tutor is very dominant, offering primarily explanation and quizzing the student, then tutoring is more properly considered "guided construction" (Chi, 2009) rather than interactive. In contrast, if the tutor tries to get the student talking more by replacing explanations with scaffolding pumps (e.g., "What else can you say") and hints, then tutoring can become interactive (Chi et al., 2001).

Scaffolding is a pervasive and arguably widely misunderstood concept in the learning sciences, and as such has been defined in many different ways. As discussed by Olney (2014), the original notion of scaffolding (Wood, Bruner, & Ross, 1976) builds upon and clarifies Vygotsky's Zone of Proximal Development (ZPD), "the distance between the actual developmental level as determined by independent problem solving and the level of potential development as determined through problem solving under adult guidance or in collaboration with more capable peers," such that "what a child can do with assistance today she will be able

to do by herself tomorrow" (Vygotsky, 1978, pp. 86-87). The original theory of scaffolding can be interpreted as defining the ZPD in terms of gap between comprehension and production:

> In the terminology of linguistics, comprehension of the solution must precede production. That is to say, the learner must be able to recognize a solution to a particular class of problems before he is himself able to produce the steps leading to it without assistance. (Wood et al., 1976, p. 90)

Comprehension sets the upper bound of the ZPD; if the student were incapable of recognizing the solution when shown, then the task would be outside their ZPD and what they were currently capable of learning. In learning, comprehension and production must become coordinated, such that the standard set by others becomes internalized and used to self-monitor production, i.e. to monitor and repair production errors (Clark & Hecht, 1983). With respect to ICAP and ICAP-A, scaffolding within the ZPD involves proactive and reactive control from the self as well as reactive control arising from comprehending the tutor's behavior. When the tutor's behavior is suggestive (e.g. leading questions), comprehension processes and reactive control are more fully engaged than when the tutor's behavior is directive (e.g. tutor explanations), because the student must internalize reactive control.

The original theory of scaffolding defines six scaffolding functions consistent with the interactive mode of ICAP and which occur in AutoTutor. Although these six functions were proposed to describe scaffolding young children arranging puzzle blocks into pyramids, they are common elements in naturalistic tutoring.

Recruitment. The tutor gains the student's attention, interest, and commitment to the learning task. Recruitment is perhaps taken for granted in formal learning environments and is not explicitly represented in AutoTutor. However, concrete and motivating examples, very common in expert tutoring (Person & Graesser, 2003), can serve to recruit students to

the task. In AutoTutor, motivating examples can be authored into the problems students are asked to solve.

Reducing degrees of freedom. The tutor reduces task difficulty to the appropriate level. AutoTutor implements this function through the hint-prompt-assertion cycle, which incrementally reduces the difficulty of a student covering an expectation.

Direction maintenance. The tutor keeps the student focused on the current goal and provides motivational support when needed. AutoTutor provides direction maintenance using the curriculum script, which provides direction maintenance at the level of the 5-step tutoring frame (problem level) as well as expectation misconception tailored dialogue (expectation level; hint-prompt-assertion).

Marking critical features. The tutor draws attention to critical task features like incorrect solutions. AutoTutor implements this function by asking leading questions like hints and prompts and by providing feedback.

Frustration control. The tutor tries to limit the student's negative affect from lack of progress. The minimal implementation of AutoTutor does not attempt frustration control directly, but may indirectly provide frustration control by giving indirect feedback and less negative feedback than is warranted, a strategy that appears to be employed by human tutors (Graesser et al., 1995; Person & Graesser, 2003).

Demonstration. The tutor models the solution in part or whole. AutoTutor implements this function by asserting an expectation at the end of the hint-prompt-assertion cycle.

The alignment between AutoTutor and the six scaffolding functions suggests that AutoTutor's scaffolding, and thus interactivity under ICAP, completely derives from two aspects of its design, the hint-prompt-assertion cycle and feedback. Considered in this light, AutoTutor's

rigid, simplistic hint-prompt-assertion cycle and relatively non-discriminating feedback (a product of LSA) are not weaknesses that need to be overcome by applying more sophisticated AI, but rather the fundamental elements that make AutoTutor work.

Testing effect

Scaffolding, as discussed, falls short of a full account of AutoTutor's effectiveness based in cognitive processes and mechanisms. However, it does suggest where to look for a full account, namely the testing effect, which includes both answering questions and receiving feedback. Briefly stated, the testing effect enhances memory for material by following exposure to that material with testing (Roediger & Karpicke, 2006). The nature of the test can greatly influence the strength of the testing effect. Recognition tests, like multiple choice tests, are less effective than recall tests like short answer tests (Andrew C. Butler & Roediger, 2007). However, testing without ever giving the correct answer when students are wrong can make this effect disappear (Kang, McDermott, & Roediger, 2007). Furthermore, the testing effect is stronger when a participant answers a question correctly than when they answer incorrectly (Andrew C Butler, 2010) While the testing effect has often been shown for memory of identical material (the same test items used in study being used in a retention test), the testing effect has also been demonstrated in near and far transfer tasks (Butler, 2010), meaning that the testing effect applies to both shallow and deep
learning.

There are multiple competing explanations regarding the cognitive mechanisms and processes behind the testing effect (Roediger & Karpicke, 2006). A recent proposal, called constructive retrieval, attempts to account for the goals and expectations of the participant, the format of the testing, and the construction of a mental model for the material studied (Hinze, Wiley, & Pellegrino, 2013). The memory aspect of constructive retrieval is based on an elaborative account of memory retrieval, such that retrieval strengthens memory for not only

the item retrieved but all of the items semantically associated with it. More difficult retrieval requires reconstructive processes that initiate additional retrievals and therefore greater strengthening of memory. When retrieval queries a mental model, the retrieval should strengthen not just the item queried but cascade across the mental model.

These recent results around the testing effect have clear parallels to the scaffolding AutoTutor employs. All AutoTutor questions are recall questions coupled with feedback. If the student is incorrect on a hint, AutoTutor proceeds to a prompt before asserting the answer, which increases the likelihood that the student will be able to address the expectation correctly and therefore capitalize on the testing effect. The hint-prompt-assertion strategy also reinforces constructive retrieval by giving the student the maximal opportunity to engage in constructive processing during retrieval (pumps and hints) before giving lower constructive processing opportunities (prompts and assertions). Constructive retrieval would predict that more constructive processing leads to greater learning, and this prediction is supported by AutoTutor studies in which pumps and hints are more highly correlated with student learning than prompts and assertions (Jackson, Person, & Graesser, 2004).

It is interesting to ask just how much ITS learning gains might be explained by the testing effect. A recent meta-analysis of ITS found an average effect size (g) of .62 relative to conventional instruction (Kulik and Fletcher, 2016), and a complementary meta-analysis of the testing effect found an average effect size (g) of .51 relative to restudying. Thus, one might speculate that perhaps 80% of the effectiveness of an ITS like AutoTutor might be attributable to the testing effect. However, several previous AutoTutor studies suggest it could be as high as 100%.

VanLehn et al. (2007) report four AutoTutor studies that have a comparison condition that could be interpreted as a testing effect condition. The testing effect condition required students to write an essay answer in response to a problem statement, read a mini-lesson

covering all possible answer flaws, edit the essay, and then review the ideal answer. In other words, the testing effect condition included a constructive retrieval task, feedback, the opportunity to correctly retrieve the material, and additional feedback. All four studies found no significant difference between AutoTutor and the testing effect condition, even on a far-transfer essay test and a one-week delayed post-test.

The Challenge of Making AutoTutor More Effective

AutoTutor is highly effective at helping students learn, averaging a large effect size across various studies, Cohen's *d* = .80 (Nye et al., 2014), despite having relatively shallow AI compared to other ITS. What makes AutoTutor so unreasonably effective? AutoTutor's highly interactive scaffolding, consisting of leading questions and feedback, is likely the reason. The structure of AutoTutor's scaffolding creates a testing effect by giving the student the maximal chance for constructive retrieval of each expectation.

Perhaps paradoxically, AutoTutor's shallow AI may contribute to rather than work against its effectiveness. AutoTutor's AI is principally based on LSA, which it uses to sequence dialogue and determine feedback. Although much dialogue sequencing is based on static content rather than adapting to the student in real time, using LSA to sequence content will preferentially select the longest/most difficult expectations and corresponding questions, which creates greater opportunities for constructive retrieval than if shorter/easier expectations and questions were selected. So being less adaptive may actually help enhance the testing effect. Feedback using LSA would likewise seem disadvantageous, as LSA is not particularly precise at assessing correctness. If the LSA threshold is too high, then AutoTutor will not count student answers as correct unless they are almost identical to the ideal answer. A high threshold may create frustration on the part of the learner, but it may also have additional learning benefits. First, the student will receive more opportunities to engage in constructive retrieval, which should benefit learning. Second, a high LSA threshold may shift the strategy of the student from

thinking only about the right answer to thinking about alternative ways of expressing the right answer so that AutoTutor will accept it. Not only should this strategy shift engage additional constructive retrieval, but it also invokes the notion of other-monitoring, a key component in ICAP. Of course it is also possible for AutoTutor script authors to make the LSA threshold so low that AutoTutor accepts incorrect answers, which has no obvious benefits for learning, but that seems to rarely happen in practice, likely because this error is fairly easy for authors to detect.

What would it take to make AutoTutor more effective? Just from the standpoint of the testing effect, there are several outstanding challenges. First, it would be ideal if AutoTutor could select the question with the greatest opportunity for constructive retrieval that the student was also likely to get correct. An important first step towards this would be to consider how AutoTutor would gain the information needed to make this determination, as its student model starts empty and is constructed based on student responses to questions. Second, spacing and interleaving retrieval practice by querying an expectation, switching to another expectation, and then querying the first expectation again, should further enhance learning. This pattern has been followed in some past AutoTutor implementations, but only if the student failed to cover the expectation the first time. Likely there is a benefit to spaced practice even if the student covers an expectation on the first hint, and such practice could be computed according to existing models of optimal practice (Pavlik & Anderson, 2008). Finally, if evaluation of student answers were more diagnostic, AutoTutor could potentially respond with highly specific feedback or even follow-up questions to probe ambiguous answers or answers that conflate concepts. Such evaluation is extremely difficult but is currently being pursued by researchers (Rus et al., 2017). To put these challenges in perspective, recall that ITS are already as effective as human tutors (Kulik & Fletcher, 2016): these challenges represent what it might take to go beyond.

References

Butler, A. C. [Andrew C]. (2010). Repeated testing produces superior transfer of learning relative to repeated studying. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, *36*(5), 1118–1133.

Butler, A. C. [Andrew C.] & Roediger, H. L., III. (2007). Testing improves long-term retention in a simulated classroom setting. *European Journal of Cognitive Psychology*, *19*(4-5), 514–527. doi:10.1080/09541440701326097. eprint: http://dx.doi.org/10.1080/09541440701326097

Chi, M. T. H. (2009). Active-constructive-interactive: A conceptual framework for differentiating learning activities. *Topics in Cognitive Science*, *1*(1), 73–105. doi:10.1111/j.1756-8765.2008.01005.x

Chi, M. T. H., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). Learning from human tutoring. *Cognitive Science*, *25*(4), 471–533.

Chi, M. T. H. & Wylie, R. (2014). The ICAP framework: Linking cognitive engagement to active learning outcomes. *Educational Psychologist*, *49*(4), 219–243. doi:10.1080/00461520.2014.965823. eprint: http://dx.doi.org/10.1080/00461520.2014.965823

Clark, E. V. & Hecht, B. F. (1983). Comprehension, production, and language acquisition. *Annual Review of Psychology*, *34*(1), 325–349. doi:10.1146/annurev.ps.34.020183.001545. eprint: http://www.annualreviews.org/doi/pdf/10.1146/annurev.ps.34.020183.001545

Graesser, A. C., Conley, M. W., & Olney, A. (2011). Intelligent tutoring systems. In K. R. Harris, S. Graham, T. Urdan, A. G. Bus, S. Major, & H. L. Swanson (Eds.), *APA educational psychology handbook, vol 3: Application to teaching and learning* (pp. 451–473). Washington, DC, US: American Psychological Association.

Graesser, A. C., Lu, S., Jackson, G. T., Mitchell, H., Ventura, M., Olney, A., &

Louwerse, M. M. (2004). AutoTutor: A tutor with dialogue in natural language. *Behavioral Research Methods, Instruments, and Computers*, *36*, 180–193.

Graesser, A. C., Person, N. K., & Magliano, J. P. (1995). Collaborative dialogue patterns in naturalistic one-to-one tutoring. *Applied Cognitive Psychology*, *9*, 1–28.

Hinze, S. R., Wiley, J., & Pellegrino, J. W. (2013). The importance of constructive comprehension processes in learning from tests. *Journal of Memory and Language*, *69*(2), 151–164.

Jackson, G. T., Person, N. K., & Graesser, A. C. (2004). Adaptive tutorial dialogue in AutoTutor. In *Proceedings of the workshop on dialog-based intelligent tutoring systems at the 7th international conference on intelligent tutoring systems* (pp. 368–372). Universidade Federal de Alagoas, Brazil.

Kang, S. H. K., McDermott, K. B., & Roediger, H. L., III. (2007). Test format and corrective feedback modify the effect of testing on long-term retention. *European Journal of Cognitive Psychology*, *19*(4-5), 528–558. doi:10.1080/09541440601056620. eprint: http://dx.doi.org/10.1080/09541440601056620

Kulik, J. A. & Fletcher, J. D. (2016). Effectiveness of intelligent tutoring systems. *Review of Educational Research*, *86*(1), 42–78. doi:10.3102/0034654315581420. eprint: http://dx.doi.org/10.3102/0034654315581420

Landauer, T. K., McNamara, D. S., Dennis, S., & Kintsch, W. (Eds.). (2007). *Handbook of latent semantic analysis*. Mahwah, New Jersey: Lawrence Erlbaum.

Menekse, M., Stump, G. S., Krause, S., & Chi, M. T. H. (2013). Differentiated overt learning activities for effective instruction in engineering classrooms. *Journal of Engineering Education*, *102*(3), 346–374. doi:10.1002/jee.20021

Mills, C., D'Mello, S., Bosch, N., & Olney, A. M. (2015). Mind wandering during learning with an intelligent tutoring system. In C. Conati, N. Heffernan, A. Mitrovic, & M. F. Verdejo (Eds.), *Artificial intelligence in education* (Vol. 9112, pp. 267–276). Lecture

Notes in Computer Science. Springer International Publishing. doi:10.1007/978-3-319-19773-9_27

Nye, B. D., Graesser, A. C., & Hu, X. (2014). AutoTutor and family: A review of 17 years of natural language tutoring. *International Journal of Artificial Intelligence in Education*, *24*(4), 427–469. doi:10.1007/s40593-014-0029-5

Olde, B. A., Franceschetti, D., Karnavat, A., & Graesser, A. C. (2002). The right stuff: Do you need to sanitize your corpus when using Latent Semantic Analysis? In *Proceedings of the 24th annual meeting of the cognitive science society* (pp. 708–713). Mahwah, NJ: Erlbaum.

Olney, A. M. (2014). Scaffolding made visible. In R. Sottilare, A. Graesser, X. Hu, & B. Goldberg (Eds.), *Design recommendations for intelligent tutoring systems: Instructional management* (Vol. 2, pp. 327–340). Adaptive Tutoring. Orlando, FL: Army Research Laboratory. Retrieved from https://gifttutoring.org/documents/

Olney, A. M., Graesser, A. C., & Person, N. K. (2010). Tutorial dialog in natural language. In R. Nkambou, J. Bourdeau, & R. Mizoguchi (Eds.), *Advances in intelligent tutoring systems* (Vol. 308, pp. 181–206). Studies in Computational Intelligence. Berlin: Springer-Verlag.

Olney, A. M., Risko, E. F., D'Mello, S. K., & Graesser, A. C. (2015). Attention in educational contexts: The role of the learning task in guiding attention. In J. Fawcett, E. F. Risko, & A. Kingstone (Eds.), *The handbook of attention* (pp. 623–642). MIT Press.

Pavlik, P. I. & Anderson, J. R. (2008). Using a model to compute the optimal schedule of practice. *Journal of Experimental Psychology: Applied*, *14*(2), 101–117.

Person, N. K. & Graesser, A. C. (2003). Fourteen facts about human tutoring: Food for thought for ITS developers. In *Ai-ed 2003 workshop proceedings on tutorial dialogue systems: With a view toward the classroom* (pp. 335–344).

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*(2), 205–229.

Roediger, H. L., III & Karpicke, J. D. (2006). The power of testing memory: Basic research and implications for educational practice. *Perspectives on Psychological Science*, *1*(3), 181–210. PMID: 26151629. doi:10.1111/j.1745-6916.2006.00012.x. eprint: https://doi.org/10.1111/j.1745-6916.2006.00012.x

Rus, V., Olney, A. M., Foltz, P. W. F., & Hu, X. (2017). Automated assessment of learner-generated natural language responses. In R. Sottilare, A. Graesser, X. Hu, & G. Goodwin (Eds.), *Design recommendations for intelligent tutoring systems: Assessment methods* (Chap. 13, Vol. 5, pp. 155–170). Adaptive Tutoring. Available at: https://gifttutoring.org/documents/. Orlando, FL: U.S. Army Research Laboratory.

VanLehn, K., Graesser, A. C., Jackson, G. T., Jordan, P., Olney, A., & Rose, C. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, *31*, 3–62.

Vygotsky, L. S. (1978). *Mind in society*. Cambridge, MA: Harvard University Press.

Wiggins, B. L., Eddy, S. L., Grunspan, D. Z., & Crowe, A. J. (2017). The ICAP active learning framework predicts the learning gains observed in intensely active classroom experiences. *AERA Open*, *3*(2), 2332858417708567. doi:10.1177/2332858417708567. eprint: https://doi.org/10.1177/2332858417708567

Wood, D., Bruner, J., & Ross, G. (1976). The role of tutoring in problem solving. *Journal of child psychology and psychiatry*, *17*(2), 89–100.

Woolf, B. P. (2008). *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. Morgan Kaufmann Publishers/Elsevier. Retrieved from http://books.google.com/books?id=MnrUj3J%5C_VuEC