Contrasting Fundamental Assumptions in Adaptive Learning and Modeling Human Tutors with TutorIT

JOSEPH M. SCANDURA*

University of Pennsylvania and TutorITweb.com

Adaptive learning has become a dominant theme in settings ranging from academic laboratories to commercial education. Despite tens of millions of dollars invested by governments, universities, the private sector and companies, however, progress has been both costly and limited. No established initiative has attempted to model the processes human teachers and tutors use in interacting with students. This article is an attempt to explain why. It also outlines a potential solution to the problem: Highly efficient authoring and delivering platforms that make it possible to both efficiently create and deliver human like tutorials.

INTRODUCTION

Several decades back, Benjamin S. Bloom a well-respected educational researcher reported that students who received one on one tutoring did better on average than 98% of the students who covered the same material in a classroom setting. This two-standard deviation difference is often referred to as Bloom’s 2-sigma effect (Bloom, 1984). http://web.mit.edu/5.95/readings/bloom-two-sigma.pdf.

Several generations of researchers have tried since but to date no one has been able to duplicate those results.

INTELLIGENT TUTORING SYSTEMS (ITS)

In the same time frame, a team of psychologists at Carnegie Mellon University introduced Intelligent Tutoring Systems (ITS) as one way to solve the problem.

*Corresponding author: scandura@scandura.com
ITSs are based on sets of condition-action pairs, called production systems, together with assumed learning mechanisms. Carnegie Learning (CL) math builds directly on these ideas. ALEKS is another major commercial product, this one based on knowledge space theory. Knowledge in ALEKS is represented in terms of relational networks.

After years of research and tens of millions spent perfecting and evaluating such systems, both kinds of ITS remain expensive to build and maintain. Each such system must be developed de novo for each body of content to be taught (cf. Scandura, 2014a). The process has been both demanding and time consuming – one tutorial at a time. The Carnegie ITS, for example, uses large workbooks to support declarative (e.g., factual) knowledge. Moreover, the results in the recent $6M Rand Corporation study of Carnegie Algebra show no gain after one year and a move from the 50th to 58th percentile in year 2 – well below the 2 sigma ideal of 98th.

BIG DATA

More recently, attention has turned to decision making based on large data bases (BIG DATA). Knewton, for example, has created an adaptive learning engine that makes all decisions automatically for any and/or all content. Google has had tremendous success in this area. Why not do the same thing in education? Using a variety of statistical techniques (e.g., Item Response Theory, active versus non-productive time), the Knewton engine makes all pedagogical decisions automatically.

What’s largely missing in this decision making is Human judgment! The importance of human judgement along with AI is the essential point of a very recent Time Magazine article on AI pioneer David Gelernter (March 7, 2016). Google search, for example, is a great tool for narrowing choices. Ultimate choices, however, require human input – not just automation.

The same is true in education, specifically in tutoring. Despite decades of research, human tutoring remains the gold standard. BIG DATA solutions have another, more basic limitation. Decision making is based on student behavior on problems, taken as wholes. Automated decision making based on BIG DATA does not get at subordinate cognitive (or other) operations and decisions students must master to be successful. This requires human judgement.

HOW BIG DATA WORKS

Ignoring layouts, for example, Knewton might test on column subtraction problems like: 9–4=, 95–53=, 667–348=, 4802–3489=. If a student gets one right,
Knewton will move to a “harder” problem – a problem that typical students might find more difficult. If wrong, the student would be moved to an “easier” one.

**All decisions are based on what students on average find easier or harder.** No matter how much data is collected, or the way data is analyzed, easier or harder is a general measure. It may have little if anything to do with specific difficulties any given student might be having on any given problem. Suppose a student gets 667–348 correct (i.e., answers 319). In this case, Knewton, for example, might search and find a related problem that some subgroup of students on average find more difficult – for example, 4802–3489=. Conversely, if incorrect, Knewton will find an easier problem.

This type of decision making fails to directly address the question of why a student gets a problem wrong (or right). In the first case: What does the student know that enabled him or her to get the right answer, 319? In the second case: What was missing?

Instructional decisions in Knewton and other Big Data solutions are made automatically based on large collections of data, from millions of students studying all manner of content.

**HOW HUMAN TUTORS DO THEIR JOB**

This is not what human tutors do. Good human tutors know the subject matter. They also know what students should know and be able to do to be successful. The basic process of human tutoring is relatively straightforward: **Find out what the student knows and what he or she doesn’t know. Provide help when and as needed. Continue the process until the student demonstrates that they can solve any of the targeted tasks.** HOW the human tutor does this and HOW students learn is of strictly secondary importance.

A tutor may present information verbally, with pictures or by example, etc. Irrespective, what really matters is student mastery. If a student demonstrates mastery, tutoring may progress to more complicated tasks. If not, tutoring should not proceed until the student demonstrates mastery.

**MODELING HUMAN TUTORS**

TutorIT is designed to work like a human tutor. The first step in creating a TutorIT tutorial is to identify what students need to learn to be successful. Good human tutors know how and do this intuitively. TutorIT comes with companion authoring platforms, in all four (EZauthor, Customizer, AuthorIT and Scope & Sequence) that
make this possible. Subject Matter Experts (SME) use these authoring platforms to systematically create arbitrarily precise representations of the knowledge to be acquired.

TutorIT takes these knowledge representations as input and makes ALL tutoring decisions automatically. TutorIT automatically provides specific feedback based on what each student needs and when.

This process parallels the human tutoring process. Good human tutors know precisely what students need to learn. To be successful on a given problem, the student must know what to do at each step of the process – what decisions to make, what actions and when. Given any student response, TutorIT lets the student know whether an answer is right or wrong, and directly or indirectly what should be done and when.

TutorIT uses patented processes to quickly determine what part(s) of what needs to be learned any given student knows at each point in time. Student knowledge changes incrementally as result of learning. Moreover, TutorIT does this with optimal efficiency. TutorIT tests and/or provides remediation on every step and every decision students must make to be successful. When a problem comes up, TutorIT efficiently pinpoints precisely what the student knows and what is missing. TutorIT does this in a highly efficient manner. Success on specific steps in a solution enables TutorIT to automatically infer how a student will perform on more complex tasks – conversely for failure. Like a good human tutor, TutorIT also (automatically) remediates on the spot.

Given 4802–3489, for example, TutorIT is not satisfied with just the answer. TutorIT evaluates each step in the process and like a human tutor takes appropriate action. It determines, for example, whether or not the student knows a basic fact, can regroup from the next column and what to do if the top digit in the next column is 0. The student must demonstrate not only what to do but when to do it. Unlike other adaptive learning systems, the process stops only after a student demonstrates the level of mastery predetermined by the author. Where desired, even mastery levels can easily be adjusted by the teacher.

TutorIT can do still more. Students may be required to do more than simply get things right. Authors may require students to know especially important things very well – to automate, to perform various combinations of steps in one’s head. TutorIT can enforce such requirements automatically. In each case, think of a tireless tutor working with the student monitoring and reacting to each step the student takes – continuing until the student either demonstrates a predetermined level of mastery or gives up. Like a good human tutor, TutorIT interacts with students dynamically while they are solving problems.
(NOTE: The caveat “gives up” reveals a common limitation in ALL tutoring or instruction, whether human directed or automated. The student must be motivated to continue. Having said this, research demonstrates that success is perhaps the strongest motivator of all.)

ALL ITS ARE CONTENT SPECIFIC

Traditional ITS place the focus on HOW students learn. Tutoring in each ITS is tied very closely to the specific content being taught. All pedagogical decisions in each ITS must be programmed and tested individually for each body of content. This is a time and expertise dependent process. Even worse, ITS researchers also have found that the cognitive task analysis procedures used miss 70% of what needs to be learned for success (Clark, 2013).

AUTHORIT AND TUTORIT

AuthorIT and TutorIT rest on a theoretical foundation that is very different than either content specific Intelligent Tutoring systems (ITS) or Knewton-like systems based on Big Data (e.g., Scandura, 2007, 2014a). Collectively, AuthorIT and TutorIT address both problems. Given any content domain, the AuthorIT authoring system guides authors as they identify WHAT students need to learn for success (Scandura, 2014a). The result is an arbitrarily detailed knowledge representation that accommodates the needs of all students in any given target population.

In short, AuthorIT is used to represent the knowledge to be acquired. AuthorIT is based on Structural Analysis (SA), an explicit process in the SLT used to represent to be learned knowledge. Unlike other forms of task analysis, all knowledge has both declarative and procedural aspects. While further discussion is beyond the scope of this short note, this is accomplished in SA by representing ALL knowledge hierarchically (e.g., Scandura, 2007). SA can even accommodate different ways of accomplishing any given task.

Because of the proprietary way knowledge is represented, TutorIT takes any such knowledge representation as input and makes all pedagogical decisions automatically – without any programming. The result is a highly efficient authoring system AND a highly efficient tutoring system that applies universally to ALL content.

AuthorIT and TutorIT are distinguished by two unique features. First is AuthorIT’s unique ability to systematically represent what needs to be learned for
success—simultaneously at arbitrarily many levels of expertise. Second is TutorIT’s unique and general capability to make use of such representations to make pedagogical decisions (what to test, what to teach and when) all automatically. All such decisions are based entirely on the hierarchical structure of given knowledge representations. These unique features have multiple side benefits – such as being able to develop once and run the same tutorial in different modes: a) as a dynamically adaptive TutorIT tutorial, b) as a highly efficient (adaptive) diagnostic test, c) simple performance aid or d) practice or test. Moreover, teachers can easily increase or decrease required levels of mastery for their students as desired.

TutorIT automatically does many things that require custom programming in traditional ITS. For example, TutorIT infers what operations and decisions any given student has mastered at each stage of the process. Success on some steps enables TutorIT to automatically and very efficiently infer mastery (or lack thereof) on so-called easier or harder (sub)tasks. Furthermore, TutorIT ensures by design that a student who completes a given tutorial will have mastered everything that the author specified as necessary for success. Authors decide such things as the best way to perform any given class of tasks, the level of mastery required for success, what should be automated and to what degree.

All inferencing is automatic and based directly on observable behavior. TutorIT does not require tracking free form student processes as in traditional ITS. The foundational SLT, as well as AuthorIT and TutorIT, are based directly and solely on observable student behavior.

The above conclusions are supported by numerous basic research studies based on SLT. Key theoretical ideas have been summarized in (e.g., Scandura, 2007, 2013) – with essential methods detailed in five patents. A recent study conducted by Novak (2015) with TutorIT supports a core hypothesis. A student may not finish a TutorIT tutorial. But, if a student does complete a given TutorIT tutorial, that student will by definition have demonstrated mastery of the content—to a level pre-specified by the author. One might disagree with an author’s opinion as to what students should be learning, or whether any given tutorial covers all that is important. Concerns as to educational value, however, are independent of mastery.

REFERENCES

Clark, R. vs. Scandura, J. M. TICL Presidential Round Table organized by P. Fadde on Cognitive Task Analysis. Top-down or Bottom-up, TICL, 2013.