

Transforming Content into Dialogue-based Intelligent Tutors

Robby Robson, Fritz Ray

Eduworks Corporation

Corvallis, OR

robby.robson@eduworks.com, fritz.ray@eduworks.com

Zhiqiang Cai

University of Memphis

Memphis, TN

zca@memphis.edu

ABSTRACT

Multiple studies have demonstrated significant learning gains from intelligent tutoring systems. At the same time, training organizations are heavily invested in eLearning and schoolhouse materials. Moving from these more traditional and less effective instructional methods to personalized, interactive and more effective methods is desirable, but it is also impractical unless it can be done efficiently and cost effectively.

The research presented in this paper describes a process that can accomplish this goal by transforming existing digital learning content into cloud instances of interactive, dialogue-based intelligent tutoring systems. The process in question starts with a collection of instructional material and applies data mining techniques to identify key concepts. It then applies a different set of techniques to break the content into nuggets associated with those concepts and find an optimal learning path. In the resulting system, the nuggets are presented to learners as text and speech. Learners then answer questions about the content, and the system uses techniques from computational linguistics to analyze and adapt to learner responses. This paper outlines the process, describes the underlying technology and methods, and discusses lessons learned from prototyping the intelligent tutoring process with combat medic training.

Although the chief value of this paper to the community is in the transformational process it represents, the paper also reports several results concerning automated processing of text-based learning content that are of interest in their own right and that fit in with the modern trend of applying methods from “big data” to problems in other areas, including education and training.

ABOUT THE AUTHORS

Dr. Robby Robson is president and co-founder of Eduworks Corporation. His research interests include online learning environments, reusable design, and applications of computational linguistics to learning, education, and training. He chaired the IEEE Learning Technology Standards Committee from 2000 – 2008 and has led multiple National Science Foundation and Department of Defense projects that explored the use of emerging technologies. He holds a doctorate in mathematics from Stanford University and has held posts in both academia and industry.

Edward “Fritz” Ray is the lead software engineer at Eduworks Corporation. He has architected and developed applications in areas ranging from educational digital libraries to semantic analysis, semantic search, competency management, and patent analysis. In addition, Fritz has researched and made improvements to semantic similarity algorithms and designed and developed middleware for rapid web service development and deployment in domains that use unstructured data. Mr. Ray received his B.S. in Computer Science from the Oregon Institute of Technology.

Zhiqiang Cai is a research assistant professor with the Institute for Intelligent Systems at the University of Memphis. He has a Master of Science degree in computational mathematics received in 1985 from Huazhong University of Science and Technology, P. R. China. His current research interests are in algorithm design and software development for tutoring systems and natural language processing. He is the chief software designer and developer of QUAID (question understanding aid), Coh-Matrix (a discourse analysis engine) and AutoTutor (conversation-based intelligent tutors).

Transforming Content into Dialogue-based Intelligent Tutors

Robby Robson, Fritz Ray
Eduworks Corporation
Corvallis, OR

robby.robson@eduworks.com, fritz.ray@eduworks.com

Zhiqiang Cai
University of Memphis
Memphis, TN
zca@memphis.edu

INTRODUCTION

Intelligent tutoring systems (or simply “intelligent tutors”) are adaptive learning systems modeled on one-on-one human tutoring. They are known to be effective: Studies such as (Dodds and Fletcher 2004, VanLehn 2011) consistently report learning gains on the order of one sigma, and reports such as (Fletcher 2011) show reduction in time to mastery as well. The long-standing goal is to attain a two sigma effect size (Corbett 2001), but even a one sigma effect size corresponds to a full letter grade in a traditional educational setting. In addition to being effective, intelligent tutors tend to be more interactive and more engaging than traditional online courses. Interactivity itself can improve learning: In a meta-analysis of 20 studies, (Bernard, Abrami et al. 2009) found a weighted average effect size of a .46 improvement in student achievement due to student-content interaction in online learning.

Given the learning gains and potential to increase engagement, it is desirable to deploy intelligent tutors across the entire spectrum of Department of Defense and other training programs. To do this, fast and economical methods are needed for developing tutoring systems for a broad range of subjects. In addition, the resulting intelligent tutors should be compatible with existing technologies used to deliver online learning. For these reasons, the authors have begun developing a process that transforms existing didactic instructional materials (e.g. documents, PowerPoint presentations, manuals, and basic Interactive Multimedia Instruction (IMI)) into interactive, dialogue-based intelligent tutoring systems. This paper describes this process and reports on research results obtained to date.

GENERAL APPROACH

Intelligent tutoring systems have been described by (VanLehn 2011) in terms of an *inner loop* that provides adaptive feedback and responses based on recent activity and an *outer loop* that determines what activity comes next. Generally speaking, the outer loop is responsible for adaptation based on parameters such as learning goals and past performance, while the inner loop adapts content based on a recent series of interactions and possibly on measurements of the learner’s current physical or affective state (Durlach and Spain 2013). The transformation process we describe treats the inner and outer loop differently. In the transformed content, the outer loop provides compatibility with standard mastery-based instructional design and with Learning Management Systems (LMS), while the inner loop provides an interactive and micro-adaptive experience.

The Outer Loop

Training materials are typically organized around an overall goal (Gagné and Merrill 1990) and use a mastery learning approach that involves progressing through a set of enabling learning objectives (ELOs) to achieve a terminal learning objective (TLO) as shown in Figure 1. Although other

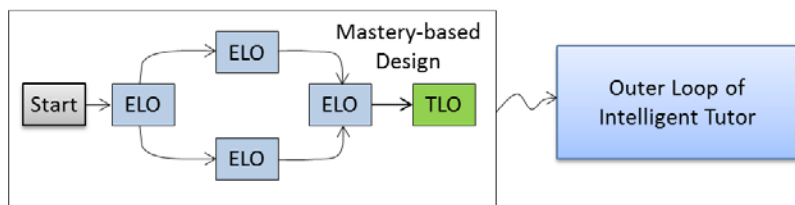


Figure 1: ELOs and TLOs map to the Outer Loop

terminology and more general structures may be used (Verbert and Duval 2008), the ELO-TLO structure is widely applicable. In our transformation process, we assume that this is the basic structure of the source content and that progression through ELO’s to the TLO will form the outer loop of the corresponding intelligent tutor.

In addition to being structured around mastery, a large body of training focuses on cognitive objectives (as opposed to affective or psychomotor objectives). For this reason we further assume that the ELOs and TLOs in the source

material correspond to mastering concepts in a given domain of instruction. To transform the source material into an intelligent tutor, we must therefore identify these concepts and the order in which they should be mastered. The steps in an automated process for doing this can be described as follows:

1. Starting with didactic instructional materials, automatically extract a set of concepts that must be learned.
2. Having extracted the concepts, automatically detect how they relate to each other as prerequisites.
3. Given a concept whose mastery is to be treated as a terminal learning objective, automatically determine an optimal path through a set of prerequisite concepts that leads to mastery of the terminal objective.

To accomplish these steps, we applied and extended techniques from data mining, computational linguistics and artificial intelligence (AI) that have previously been used to automate related processes such as grading, metadata generation, and question generation (Robson and Ray 2012). The technical details of these techniques are beyond the scope of this paper, but their outputs are described in more detail later on. The use of data mining and computational linguistics to extract concepts and detect their relationships is one key to our process.

The Inner Loop

Once we have determined what concepts will be taught and in what order they should be learned, the next step in our process is to use the source content to create a system that teaches each concept in an interactive, adaptive fashion. In other words, we develop the inner loop for each concept. For this purpose we use *AutoTutor Lite*TM (Hu, Cai et al. 2009, Wolfe, Widmer et al. 2013), one of a family of conversation-based intelligent tutors currently being used to teach computer literacy, physics, critical thinking and other subjects (Graesser, Chipman et al. 2005, Graesser, Penumatsa et al. 2007, Hu, Cai et al. 2009, Person, Olney et al. 2012). *AutoTutor Lite* presents content and engages learners in conversations with an avatar. Learner responses are evaluated using latent semantic analysis (Hu, Cai et al. 2007). The avatar asks questions that give hints or contain prompts based on student responses.

The second key to our process is using the source content to create instances of *AutoTutor Lite* for each concept addressed in the outer loop (Figure 2). This step also involves computational linguistics and natural language processing. To date, this step has been partially automated with ongoing research efforts targeting high degrees of automation.

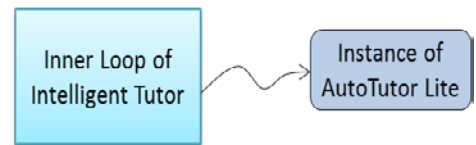


Figure 2: Inner Loop

SCORM for the Outer Loop

The third key to our process is using SCORM sequencing and navigation (ADL 2013) to implement the outer loop. This provides an easy path to LMS-compatibility and SCORM delivery engines such as SCORM Cloud (Rustici Software 2012) that automatically display tables of contents and navigation elements. The SCORM sequencing paradigm is closely aligned with traditional instructional designs (Robson and Regan 2008), making it easy to implement a progression of learning objectives in a SCORM package. Figure 3 shows the overall process.

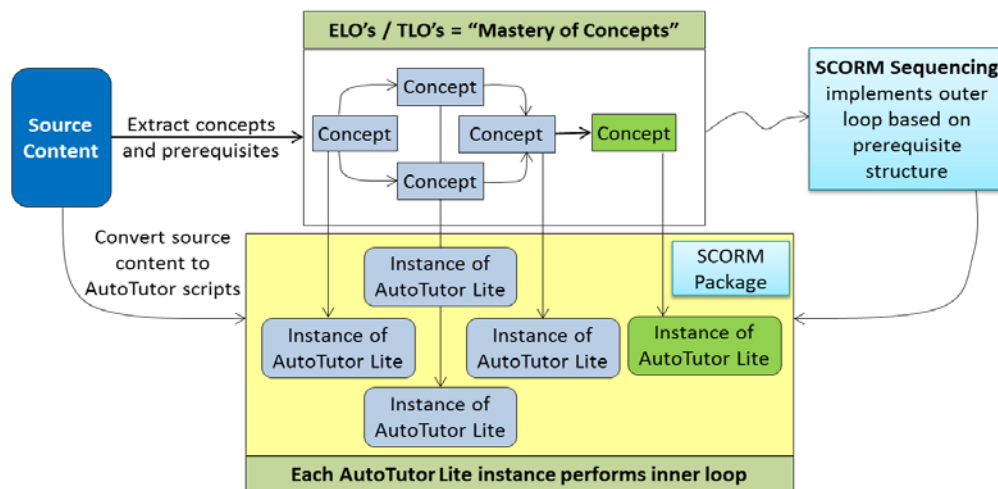


Figure 3: Overview of Transformation Process

Setting Expectations

The performance of computational linguistics algorithms is often measured using *precision* and *recall* as portrayed in Figure 4. In classical information retrieval, there is a universe U of objects and a set D of desired objects. For example, if one is searching for text passages relevant to a particular learning objective, then U is the set of all text passages and D is the set of matching passages. A retrieval (or computational linguistics) algorithm works perfectly if it retrieves exactly the objects in D and none that are not in D . In practice, algorithms make two kinds of mistakes (shown as area R in Figure 4): they retrieve objects *not* in D and they miss objects that *are* in D . *Recall* measures what percentage of the objects in D are identified by an algorithm and *precision* measures what percentage of retrieved objects are actually in D .

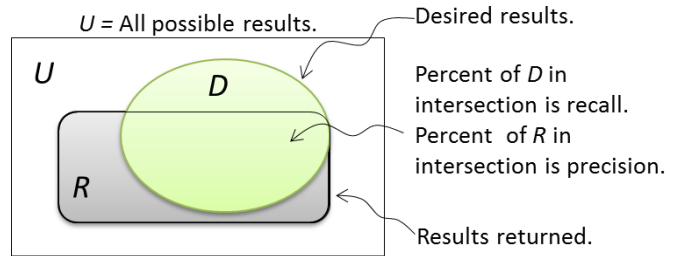


Figure 4: Recall and Precision

In practice, algorithms make two kinds of mistakes (shown as area R in Figure 4): they retrieve objects *not* in D and they miss objects that *are* in D . *Recall* measures what percentage of the objects in D are identified by an algorithm and *precision* measures what percentage of retrieved objects are actually in D .

Like search algorithms, many of the algorithms we use are actually *ranking* algorithms. Instead of returning a fixed set of objects believed to be in D they rank objects according to how likely they are to be in D . For these reasons, we often look at the percentage of the k highest ranked results that are in D for various values of k . This is called *precision at rank k* . Thus 85% precision at rank 5 means that 85% of the top 5 ranked results are in D .

In computational linguistics, precisions above 80% are considered excellent and precisions of 50% or more at high ranks are often hard to achieve (Manning, Raghavan et al. 2008, Carterette and Voorhees 2011). Automated transformation techniques based on computational linguistics may therefore require intervention to reach the level of quality acceptable in real-world settings. It is also a reminder that the goal of automation is to make transformation efficient and cost effective, not to make it instantaneous or to eliminate the need for a human in the loop.

DISCUSSION OF EACH STEP

The four main steps involved in our transformation approach are *concept extraction*, *prerequisite detection*, *inner loop instantiation* and *outer loop creation*. This section describes our results for each step.

Concept Extraction

Our approach to transformation starts with a body of didactic content, which we assume represents the way in which a subject should be taught. To test our process we used a collection of combat medic training content (Army Military Occupation Specialty 68W training) provided to us by Engineering and Computer Simulations, Inc. in Orlando. The source documents in this training content collection consisted of manuals and lessons in Microsoft Word™, PowerPoint™ and PDF format. A small portion of the table of contents from a typical lesson in a source document is shown in Figure 5:

LESSON 5	
OPENING AND MANAGING A CASUALTY'S AIRWAY	
5-1. MOVE TO SAFETY	
5-2. CHECK THE CASUALTY FOR RESPONSIVENESS	
5-3. POSITION THE CASUALTY	
5-4. OPEN THE CASUALTY'S AIRWAY (HEAD-TILT/CHIN-LIFT)	
5-5. CHECK THE CASUALTY FOR BREATHING	
5-6. DETERMINE APPROPRIATE ACTION	
5-7. PERFORM RESCUE BREATHING	
5-8. INSERT A NASOPHARYNGEAL AIRWAY	
5-9. POSITION THE CASUALTY (RECOVERY POSITION)	
EXERCISES	
SOLUTIONS	

LESSON 5	
OPENING AND MANAGING A CASUALTY'S AIRWAY	
TASK	Open and manage the airway of a simulated casualty.
CONDITIONS	Given a simulated casualty and a combat lifesaver medical equipment set.
STANDARDS	Score a GO on the performance checklists. Additional injuries to the casualty are prevented.

Figure 5: Section Of Combat Life Saver Course (AIPD 2012)

We then applied data mining techniques designed to extract a set of previously unknown “topics” from a set of documents. These methods are used in intelligence gathering, for marketing analysis and in other similar applications. Their output consists of lists of words and weightings that indicate how strongly each word relates to the topic represented by the list. Examples of such topics generated by our methods are:

- *airway > mouth > blood = fluid > bleeding ...*
- *injury >> trauma > care > identify ...*
- *position > soldier > figure > personnel > operations ...*

These do not look like the topics in the combat medic training manual (see Figure 5), but they are recognizable as instructional topics and can be used as the basis of automated computations.

To put more substance around each topic, and to prepare for generating corresponding instances of *AutoTutor Lite*, we used semantic similarity algorithms (Hu, Cai et al. 2007, Robson and Ray 2012) to identify the paragraphs from the source content that were most closely related to each topic. The paragraph is often used as the unit of semantically coherent text in computational linguistics, and bundling source content with computer-generated topics made it easier to compare them. For practical purposes, the definition of a concept in our transformation process is:

Concept = Computer-generated “topic” + Associated paragraphs from source content

To see whether our underlying topic detection and semantic similarity algorithms were working, we examined the weighted lists of vocabularies (representing the topics) and the ten most related paragraphs from the combat medic source content. We asked whether the lists of words represented discernible concepts and then looked at each associated paragraph to see whether it was relevant to the implied concepts. In a selection of 50 randomly chosen concepts we observed that:

- Only 8% of the computer-generated topics did *not* seem to capture discernible concepts.
- When we manually inspected the relevance of the top ten ranked paragraphs for each computer-generated topic, we found that 83% of paragraphs were relevant, 7% were partially relevant, and only 10% irrelevant. All of the top ten paragraphs were relevant for 34% of the computer-generated topics examined.

Within the context of computational linguistics algorithms, these results are excellent and confirmed that our algorithms were functioning as expected.

We applied the same techniques to a different domain, middle school mathematics, and obtained reasonable results but also observed a phenomenon called *topic pollution*. Topic pollution arises when instructional content contains topics unrelated to its main subject. For example, a mathematics word problem might talk about computing the volume of a barrel, but barrels are not a math topic. Similarly, instructional content has a propensity to include grading policies, instructor contact information, study hints, and other administrative data. In our initial analysis of the combat medic content, our algorithms identified a number of topics related to “course” and “lesson” that were picked up from administrative sections of documents, and when we examined a set of lesson plans, “teaching” emerged as a pivotal concept.

Topic pollution does not create a problem when the “pollutants” are isolated. A topic about barrels is usually too weak to survive the pruning process that will be described later, and if clearly unwanted topics emerge, they can be safely excised by hand. A more subtle problem arises when a pollutant topic either contains words such as “course” that appear in other topics with different meanings (e.g. course of action).

We continue to research ways to eliminate these more difficult forms of topic pollution, but even taking simple measures to eliminate topic pollution improved results in the middle school math domain. In one experiment, we applied our algorithms to learning resources that addressed math topics in a digital library called SMILE (HowToSmile 2013). SMILE contains a range of science and math activities categorized by subject. We used the official categorization as the “ground truth,” applied our algorithms without considering the official classification, and compared results. The digital library contained 2,096 harvestable resources of which 972 were in math. After eliminating basic topic pollution we achieve the results shown in Figure 6 (next page):

	Ranks				
	1 - 5	1 - 25	26 - 100	101 - 200	201 - 300
Precision	100%	84%	80%	63%	67%

Figure 6: Precision in retrieving math resources from SMILE

Prerequisite Detection

We tried several different approaches to prerequisite detection. The approach we adopted was based on the assumptions that if Concept A is an immediate prerequisite of Concept B, then

- A is likely to be strongly related to B, and
- A is likely to be discussed prior to B in source materials that relate to both concepts.

Both of these were measured using statistical techniques, the first by comparing the distribution and weights of words in computer-generated topics and the second by examining the order in which words associated with given topics appeared in source content. We then introduced cut-offs for considering two concepts to be related and for considering one related concept to be a prerequisite for learning the other.

The above computations resulted in a “concept map” that could be represented as a directed graph and pruned to generate a minimal spanning tree. The purpose behind pruning is to make the instructional process efficient: It ensures that the intelligent tutor guides the learner through a minimal set of ELO’s to achieve the TLO and that the ELO’s presented are those most strongly related to the TLO.

Figure 7 shows a small piece of the concept map generated for combat medic training. Each node represents a computer-generated topic and each edge is directional. The quantities indicated on each edge are computer-generated measures of relatedness (R) and directionality (the ratios shown). In the last step of our process, maps such as these are translated into SCORM sequencing rules.

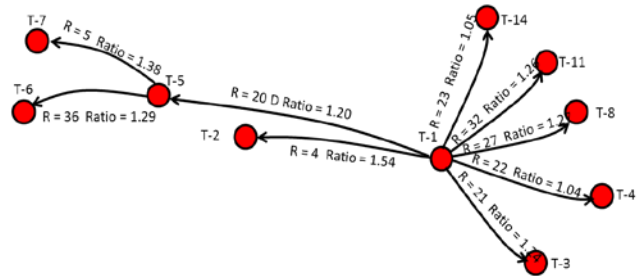


Figure 7: Computer-generated concept map

Inner Loop Creation

In our process the inner loop for each concept, or more precisely for the ELO defined by mastery of each concept, is realized as an instance of *AutoTutor Lite* created from the original source content. *AutoTutor Lite* is a purely web-based application that requires several pieces of data:

- Content that is displayed to the learner
- Scripts that are spoken by an avatar
- Questions and hints given by an avatar
- Model answers used to evaluate learner responses

Each of these can be placed into XML wrappers that are then interpreted by *AutoTutor Lite*. Internally, *AutoTutor Lite* compares student answers to content using a variant of Latent Semantic Analysis (Landauer and Dumais 2008) and provides prompts and hints based on these comparisons. It also displays several scores that indicate whether an answer is relevant or irrelevant and old or new. These scores are rolled up into a summary *coverage score* for the current attempt and an aggregate score for all answers.

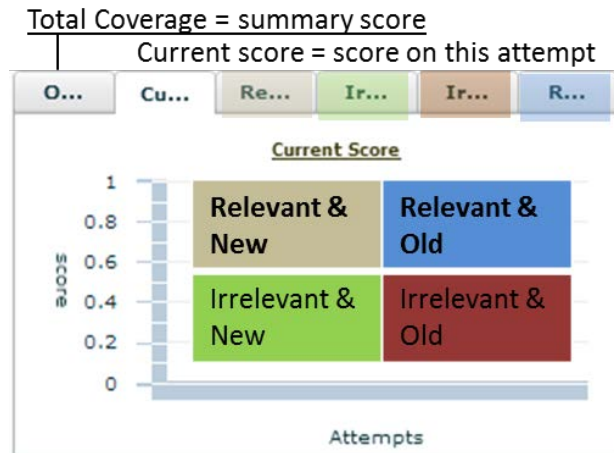


Figure 8: Scores Computed by AutoTutor Lite

Our initial attempts at populating *AutoTutor Lite* with combat medic source content were technologically successful but did not produce the best tutors. The reasons were that (a) even paragraph sized chunks of text tended to be too long when read out loud by an avatar, and (b) we did not have a means to produce hints and prompts. We have since trained an AI algorithm to recognize sentences that can be transformed into good hints and prompts using parsing and sentence rewriting techniques similar to those used for automated question generation (Mitkov, Ha et al. 2006). We have also identified computable criteria that give preference to content such as bulleted lists for use in dialogue-based tutors such as *AutoTutor Lite*. This step in our process is the subject of ongoing research.

Outer Loop Creation

The outer loop of our target intelligent tutor is implemented in SCORM, shown previously in Figure 3. We assign a SCORM objective to each concept and use internally developed tools to publish a corresponding SCORM wrapper. That process is straightforward, but for SCORM sequencing to work, the content must report on the status of objectives. *AutoTutor Lite* does not natively do this, so we added a self-reporting mechanism that enabled users to enter their total coverage score and send it back to an LMS via the SCORM API. In the future, self-reporting will be replaced with an automated reporting system that includes support for the Experience API (ADL 2013).

The complete package is shown in Figure 9. The web browser shell is generated by SCORM Cloud from a SCORM package. The bulk of the screen displays an instance of *AutoTutor Lite* created from didactic combat medic content. The menu on the left is an outer loop navigation menu that includes logic that

- Hides prerequisite lessons once a lesson is mastered, and
- Automatically displays prerequisite lessons if a lesson is attempted but not passed.

The screenshot displays the 'Tourniquet Procedure Tutor' interface. At the top, there are navigation buttons: 'PREVIOUS', 'NEXT', 'VIEW DEBUG', 'CLOSE ITEM', 'EXIT WITHOUT SAVING DEBUG', and 'RETURN TO LMS'. The left sidebar contains a 'HIDE MENU' button and a list of topics: 'Tourniquet Overview', 'Tourniquet Procedure', and 'Resuscitation'. The main content area is titled 'Tourniquet Procedure Tutor' and includes the following elements:

- INSTRUCTIONS:** Run through the entire tutor for this topic. On the last screen, type in your answer ending with a period ".", and hit "Enter". You may adjust your answer and enter as many Attempts as you like. Enter the score for your best Attempt (shown in the "Total Coverage" chart) in the field below to submit it to the LMS. Click the "Next" button above to continue based on your score.
- Score Input:** A text box containing '0.0' and a 'Submit Score' button.
- Question:** 'How do you know when the tourniquet has been applied correctly?' with a prompt 'Type Your Answer Below:'.
- Answer:** 'The bleeding has stopped and there is no pulse.'
- Total Coverage Chart:** A bar chart showing scores for three attempts. The y-axis is labeled 'SCORE' and ranges from 0 to 1. The x-axis is labeled 'Attempts'. The scores are approximately 0.3, 0.6, and 0.8.
- Avatar:** A 3D avatar of a man in a military uniform is visible in the top right corner of the simulation area.
- Background:** A 3D simulation of a combat medic scene with a patient lying on the ground and a medic attending to them.

At the bottom of the interface, it says 'Generated using University of Memphis AutoTutor Lite™ software from combat medic training content provided by Engineering & Computer Simulations, Inc.'

Figure 9: Transformed combat medic content

CONCLUSIONS AND FUTURE WORK

The process we have described holds significant promise for transforming large quantities of existing Interactive Media Instruction (IMI) Level-1 and Level-2 content (Campbell, Cantrell et al. 2006) into more effective intelligent tutors at low cost, and our team continues to engage in research designed to improve and perfect this process, both algorithmically and operationally. The research results cited demonstrate the effectiveness of the underlying algorithms, and the results obtained for combat medic content provide evidence that the end-to-end process is viable. In addition, the process of generating a concept map from a collection of existing content can be applied in other contexts where a concept map is useful, and the process of converting a concept map into SCORM sequencing simplifies the development of SCORM content regardless of whether the output is an intelligent tutor or some other form of IMI.

As a concluding remark, we note that the transformation process described in this paper produces intelligent tutors whose outer loops are based purely on mastery of objectives. Following the SCORM paradigm, these objectives are defined and reported by content. In the language of intelligent tutoring systems, there is no persistent student model (Woolf 2010), also called a learner model.

For practical purposes, this may be justified. Interoperability with existing training infrastructure may be more important than supporting additional macro-adaptation, and adding interactivity and micro-adaptation is already an improvement over IMI Level-1 and Level-2 content (Campbell, Cantrell et al. 2006). Introducing new macro-adaptations would require design time and expense. Nonetheless, it is reasonable to ask whether a transformation process is available that at least has the potential of producing more sophisticated intelligent tutoring systems that maintain persistent learner models. Two emerging technologies are now available that should make this possible.

The first such technology is the Experience API (ADL 2013), or XAPI, which is a successor to SCORM and part of the Training and Learning Architecture (TLA). TLA introduces the notion of a *Learning Record Store* which tracks data reported by the XAPI. In the long run, the TLA will enable stand-alone web applications to operate independently of an LMS and still report results to an LMS or other system. As a consequence, an XAPI version of the intelligent tutors described in this paper could persistently store a learner model.

The second such technology is the Generalized Intelligent Framework for Tutoring (Sottolare, Brawner et al. 2012, Sottolare, Goldberg et al. 2012), or GIFT. GIFT includes components (a domain module, a pedagogical module and a learner module) that provide the logic and data structures for building highly adaptive intelligent tutors. GIFT is also intended to integrate with LMS and other systems. In the future, we plan to use GIFT as a target for transformation, after we have perfected the methods described in this paper.

ACKNOWLEDGEMENTS

The research described herein has been sponsored by the U.S. Army Research Laboratory - Human Research & Engineering Directorate (ARL-HRED). Statements and opinions expressed in this paper do not necessarily reflect the position or the policy of the United States Government, and no official endorsement should be inferred.

The authors also thank Keith Brawner of the Army Research Lab for his guidance, Brent Smith of Engineering and Computer Simulations, Inc. for providing combat medic content and advice, and Dr. Xiangen Hu for his comments concerning *AutoTutor Lite* and applications of semantic technology.

REFERENCES

- ADL. (2013). "SCORM 2004, 4th Edition." Retrieved June 19, 2013, from <http://www.adlnet.gov/scorm/scorm-2004-4th>.
- ADL. (2013). "Training & Learning Architecture (TLA): Experience API (XAPI)." Retrieved June 19, 2013, from <http://www.adlnet.gov/tla/experience-api>.
- AIPD (2012). Combat Lifesaver Course: Student Self-Study, Army Institute for Professional Development. Subcourse IS0871.

- Bernard, R. M., P. C. Abrami, E. Borokhovski, C. A. Wade, R. M. Tamim, M. A. Surkes and E. C. Bethel (2009). "A meta-analysis of three types of interaction treatments in distance education." Review of Educational Research **79**(3): 1243-1289.
- Campbell, C., G. Cantrell, T. Generalao, A. Sawyer and J. Takitch (2006). Interactive multimedia instruction for US Army training. World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education.
- Carterette, B. and E. M. Voorhees (2011). Overview of information retrieval evaluation. Current challenges in patent information retrieval. M. Lupu, K. Mayer, J. Tait, A. J. Trippe and W. B. Croft. Berlin Heidelberg, Springer: 69-85.
- Corbett, A. (2001). "Cognitive computer tutors: Solving the two-sigma problem." User Modeling 2001: 137-147.
- Dodds, P. and J. D. Fletcher (2004). "Opportunities for New "Smart" Learning Environments Enabled by Next-Generation Web Capabilities." Journal of Educational Multimedia and Hypermedia **13**(4): 391-404.
- Durlach, P. and R. Spain (2013). Forms of Adaptation in a Framework for Instructional Technology. To appear.
- Fletcher, J. D. (2011). DARPA Education Dominance Program: April 2010 and November 2010 Digital Tutor Assessments. Alexandria, VA.
- Gagné, R. M. and M. D. Merrill (1990). "Integrative goals for instructional design." Educational Technology Research and Development **38**(1): 23-30.
- Graesser, A., P. Chipman, B. Haynes and A. Olney (2005). AutoTutor: An Intelligent Tutoring System with Mixed-Initiative Dialogue. IEEE Transactions on Education, IEEE. **48**: 612-618.
- Graesser, A., P. Penumatsa, M. Ventura, Z. Cai and X. Hu (2007). "Using LSA in AutoTutor: Learning through mixed-initiative dialogue in natural language."
- HowToSmile. (2013). "HowToSmile.org: All the Best Science & Math Activities." Retrieved June 19, 2013, from <http://howtosmile.org/>.
- Hu, X., Z. Cai, L. Han, S. D. Craig, T. Wang and A. C. Graesser (2009). AutoTutor Lite. IOS Press.
- Hu, X., Z. Cai, P. Wiemer-Hastings, A. C. Graesser and D. S. McNamara (2007). Strengths, limitations, and extensions of LSA. Handbook of Latent Semantic Analysis. T. Landauer, D. McNamara, S. Dennis and W. Kintsch. Mahwah, NJ, Lawrence Erlbaum Associates.
- Landauer, T. K. and S. Dumais (2008). "Latent semantic analysis." Scholarpedia **3**(11): 4356.
- Manning, C. D., P. Raghavan and H. Schütze (2008). Introduction to information retrieval, Cambridge University Press Cambridge.
- Mitkov, R., L. A. Ha and N. Karamanis (2006). "A computer-aided environment for generating multiple-choice test items." Natural Language Engineering **12**(2): 177-194.
- Person, N., A. Olney, S. D'Mello and B. Lehman (2012). Interactive Concept Maps and Learning Outcomes in Guru. Proceedings of the Twenty-Fifth International Florida Artificial Intelligence Research Society Conference. Association for the Advancement of Artificial Intelligence.
- Robson, R. and F. Ray (2012). Applying Semantic Analysis to Training, Education, and Immersive Learning. Interservice/Industry Training, Simulation, and Education Conference, Orlando, Florida.
- Robson, R. and D. Regan (2008). Using SCORM to Convert, Repurpose, Redesign, and Improve Existing Content. DevLearn 2008, eLearning Guild.
- Rustici Software. (2012). "SCORM Cloud." Retrieved June 19, 2013, from <http://scorm.com/scorm-solved/scorm-cloud-features/>.
- Sottolare, R. A., K. W. Brawner, B. S. Goldberg and H. K. Holden (2012). "The Generalized Intelligent Framework for Tutoring (GIFT)."
- Sottolare, R. A., B. S. Goldberg, K. W. Brawner and H. K. Holden (2012). A Modular Framework to Support the Authoring and Assessment of Adaptive Computer-Based Tutoring Systems (CBTS). The Interservice/Industry Training, Simulation & Education Conference (I/ITSEC), NTSA.

VanLehn, K. (2011). "The Relative Effectiveness of Human Tutoring, Intelligent Tutoring Systems, and Other Tutoring Systems." Educational Psychologist **46**(4): 197-221.

Verbert, K. and E. Duval (2008). "ALOCOM: a generic content model for learning objects." International Journal on Digital Libraries **9**(1): 41-63.

Wolfe, C. R., C. L. Widmer, V. F. Reyna, X. Hu, E. M. Cedillos, C. R. Fisher, P. G. Brust-Renck, T. C. Williams, I. D. Vannucchi and A. M. Weil (2013). "The development and analysis of tutorial dialogues in AutoTutor Lite." Behavior research methods: 1-14.

Woolf, B. P. (2010). Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning, Morgan Kaufmann.