Computational Aspects of the Intelligent Tutoring System MetaTutor

Vasile Rus[1], Mihai Lintean[1], Zhiqiang Cai[2], Amy Witherspoon[2], Arthur C. Graesser[2], and Roger Azevedo[2]

Departments of Computer Science[1] and Psychology[2], The University of Memphis, USA

ABSTRACT

We present in this chapter the architecture of the intelligent tutoring system MetaTutor that teaches students meta-cognitive strategies while learning about complex science topics. The emphasis is on the natural language components. In particular, we present in detail the natural language input assessment component used to detect students' mental models during prior knowledge activation, a meta-cognitive strategy, and the micro-dialogue component used during subgoal generation, another meta-cognitive strategy in MetaTutor. Subgoal generation involves subgoal assessment and feedback provided by the system. For mental model detection from prior knowledge activation paragraphs, we have experimented with three benchmark methods and six machine learning algorithms. Bayes Nets in combination with a word-weighting method provided the best accuracy (76.31%) and best human-computer agreement scores (kappa=0.63). For subgoal assessment and feedback, a taxonomy-driven micro-dialogue mechanism yields very good to excellent human-computer agreement scores for subgoal assessment (average kappa=0.77).

INTRODUCTION

We describe in this chapter he architecture of the intelligent tutoring system MetaTutor with an emphasis on two components that rely on natural language processing (NLP) techniques: (1) *detection of students' mental models* during prior knowledge activation (PKA), a meta-cognitive strategy, based on student-generated PKA paragraphs, and (2) the micro-dialogue component which handles *subgoal assessment and feedback generation* during subgoal generation (SG), another meta-cognitive strategy in MetaTutor.

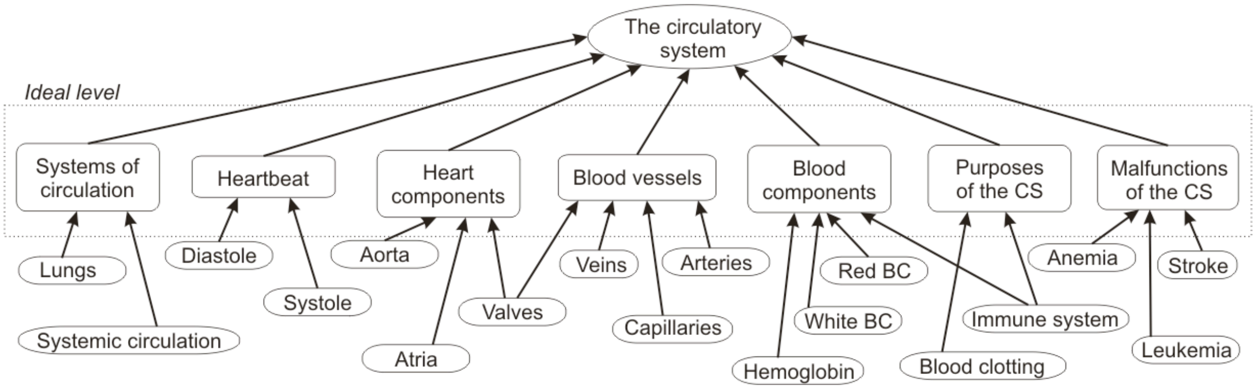
The current MetaTutor is a complex system that consists of nine major logical components: pre-planning, planning, student model, multi-modal interface (includes agents), feedback, scaffolding, assessment, authoring, and a system manager that coordinates the activity of all components. We present details about the role of each of these components and how they are implemented with various underlying technologies including dialogue processing, machine learning methods, and agents technology. We will describe in-depth the two NLP-based tasks of mental model detection, which is part of the student module, and subgoal generation, which is part of the planning module.

During prior knowledge activation, which occurs at the beginning of a student-system session, students are asked to write a paragraph describing their prior knowledge with respect to the learning goal. The task is to infer from these PKA paragraphs the *mental model* of the students. A mental model characterizes a student's level of understanding with respect to a subject matter. We regard this problem as a text categorization problem. The general approach is to combine textual features with supervised machine learning algorithms to automatically derive classifiers from expert-annotated data. The parameters of the classifiers were derived using six different algorithms: naive Bayes (NB), Bayes Nets (BNets), Support Vector Machines (SVM), Logistic Regression (LR), and two variants of Decision Trees (J48 and J48graft, an improved version of J48). These algorithms were chosen because of their diversity in terms of patterns in the data they are most suited for. For instance, naive Bayes are best for problems where independent assumptions can be made among the features describing the data. The diversity of the selected learning algorithms allows us to cover a wide range of patterns that may be hidden in the data.

The role of the subgoal generation strategy in MetaTutor is to have students split the overall learning goal, e.g. *learn about the human circulatory system*, into smaller learning units called *subgoals*. The subgoals must be specified at the ideal level of specification, i.e. not too broad/general or too narrow/specific. If student-generated subgoals are too specific or too general the system must provide appropriate feedback in natural language such that students will be able to re-state the subgoal in a form closer, if not identical, to the ideal form. The system uses a set of ideal subgoals, generated by subject matter experts, to assess the student-generated subgoals. In our work reported here, we have seven ideal subgoals associated with the general goal of learning about the human circulatory system. The subgoals can be seen in the second level of nodes in Figure 1. A taxonomy of goals/subgoals and concepts related to the subgoals was chosen as the underlying scaffold for the subgoal assessment and feedback mechanism (see Figure 1). A taxonomy can capture general/specific relations among concepts and thus can help us drive the feedback mechanism. For instance, a student-generated subgoal can be deemed too general if the subgoal contains concepts above the ideal level in the taxonomy. Similarly, a subgoal can be deemed too specific if it contains concepts below the ideal level in the taxonomy. We present the details of our taxonomy-driven subgoal assessment and feedback model and report results on how well the system can assess student-articulated subgoals.

The rest of the chapter is structured as follows. *Previous Work* presents prior research on intelligent tutoring systems with natural language interaction focusing on student input assessment and dialogue management. Next, the architecture of the MetaTutor system is presented. The subsequent section, *Prior Knowledge Activation*, presents our methods to address the task of detecting mental models during prior knowledge activation whereas the *Subgoal Generation* section describes in detail our taxonomy-based subgoal assessment and feedback generation method as well as the experiments and results obtained. The *Conclusions* section ends the chapter.

Figure 1. Partial Taxonomy of Topics in Circulatory System



PREVIOUS WORK ON INTELLIGENT TUTORING SYSTEMS WITH NATURAL LANGUAGE

Intelligent tutoring systems with natural language input have been developed at a fast pace recently (VanLehn et al. 2007). We discuss prior research on assessment of natural language student input and on dialogue management in intelligent tutoring systems because these two topics are most related to our work presented here.

Researchers working on intelligent tutoring systems with natural language input explored the accuracy of matching students' written input to a pre-selected stored answer: a question, solution to a problem, misconception, or other form of benchmark response. Examples of these systems are AutoTutor and Why-Atlas, which tutor students on Newtonian physics (Graesser et al. 2005; VanLehn et al. 2007), and the iSTART system, which helps students read text at deeper levels (McNamara et al. 2007). Systems such as these have typically relied on statistical representations, such as latent semantic analysis (LSA; Landauer et al. 2005) and content word overlap metrics (McNamara et al. 2007). LSA has the advantage of representing the meaning of texts based on latent concepts (the LSA space dimensions, usually 300-500) which are automatically derived from large collection of texts using singular value decomposition (SVD), a technique for dimensionality reduction. However, LSA cannot tell us whether a concept or a text fragment is more specific or more general than the other, which is what we need to handle student input and provide feedback during subgoal generation in MetaTutor. In our approach, we rely on a taxonomy of concepts which explicitly embeds specific/general relations among concepts or phrases. More recently, a lexico-syntactic approach, entailment evaluation (Rus et al., 2007), has been successfully used to meet the challenge of natural language understanding and assessment in intelligent tutoring systems. The entailment approach has been primarily tested on short student inputs, namely individual sentences. It could be extended to handle paragraph-size texts but not in a straightforward manner as it requires the use of a syntactic parser which operates on one sentence at a time. As both LSA and the entailment approach have some challenges with handling longer texts, such as the PKA paragraphs, we opted instead for a set of methods that combine textual features with machine learning algorithms to automatically infer student mental models. Another reason we opted for machine learning based methods is due to the importance of goals and subgoals in MetaTutor. As we will see later, we choose features for our machine learning models which are tied to the set of subgoals in MetaTutor.

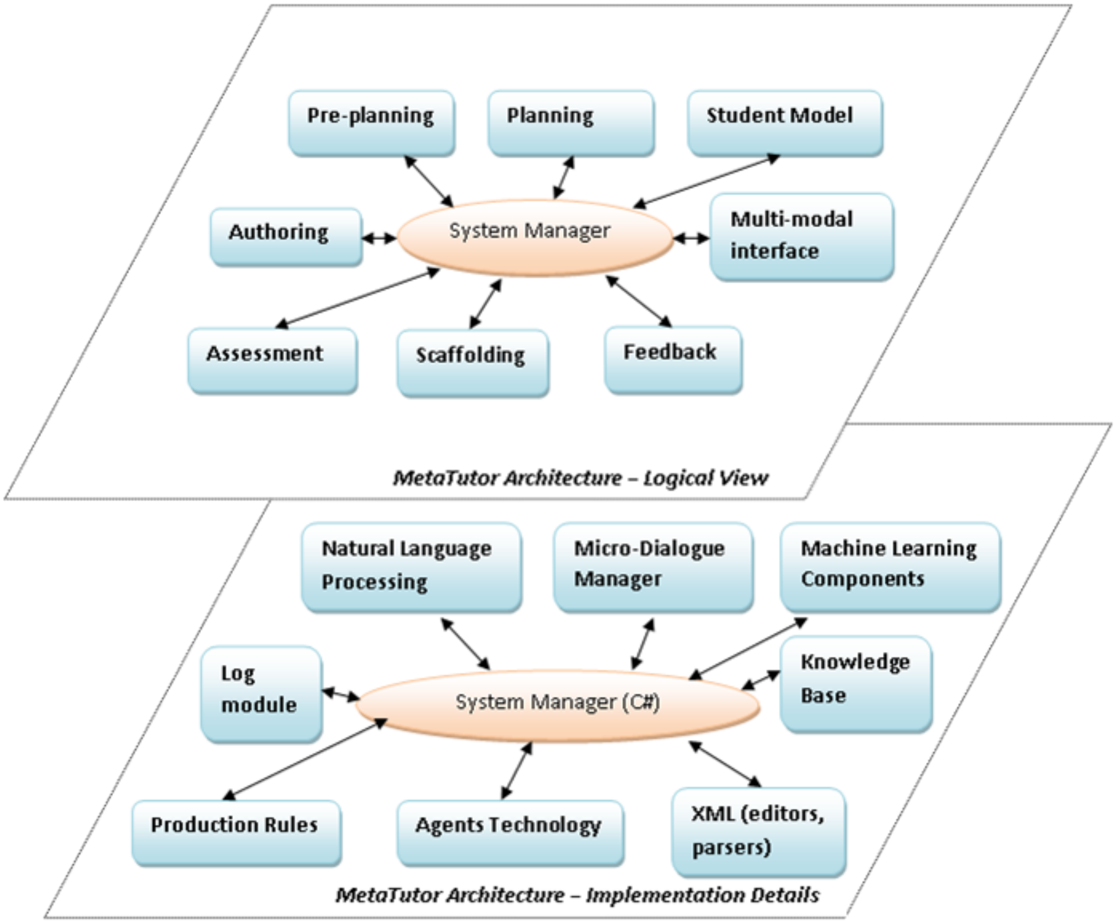
Dialogue is a major component of natural language intelligent tutoring systems. Various dialogue management models have been proposed in intelligent tutoring systems. These models are usually built around instruction and human tutoring models. The dialogue models can be described at various levels. For example, at one level the AutoTutor dialogue management model (Graesser et al. 2005) can be described as a misconception-expectation model. That is, AutoTutor (and human tutors for that matter) typically has a list of anticipated expectations (good answers) and a list of anticipated misconceptions associated with each challenging question or problem in the curriculum script for a subject matter. Our micro-dialogue management model for providing feedback during subgoal generation resembles at some extent the misconception-expectation model in AutoTutor in that we do have a set of ideal/expected subgoals. However, our dialogue management method relies on a taxonomy of concepts to manage the dialogue turns as opposed to a flat set of expectations or misconceptions. There is need for a taxonomy because we must identify general/specific relations in the student input with respect to the ideal subgoals, as already mentioned.

THE ARCHITECTURE OF METATUTOR

The current MetaTutor is a complex system that consists of nine major logical components (see top part of Figure 2). The implementation details of the system in terms of major technologies used are shown at the bottom of the figure. A screenshot of the main view of the system is shown in Figure 3.

The architecture of the MetaTutor system is open; new modules can be easily accommodated, and major changes can be made to any of the existing modules without redesigning the system from scratch. For instance, if a more advanced micro-dialogue manager is developed in the future then the current micro-dialogue manager component can be replaced (in a plug-and-play manner) without affecting the functionality of the overall system, as long as the interface with the other modules is maintained. If changes to the interface with other modules are needed then such changes must be propagated throughout the system to the connected modules-but this is still less cumbersome than redesigning from scratch. One other advantage of the current architecture is the decoupling of processing and data. This feature allows easy transfer of MetaTutor from one domain to another without changes in the processing part. All the domain-specific information as well as other configurable information (e.g., the verbal feedback the agents provide) is maintained in external, separate files that can be easily edited by domain experts, dialogue experts, or cognitive scientists. The architecture is also reconfigurable in that some modules can be turned on and off. To run a version of MetaTutor without pedagogical agents (PAs) for comparison purposes and in order to evaluate the role of PAs in self-regulated learning (SRL) modeling and

Figure 2. Overview of MetaTutor's architecture

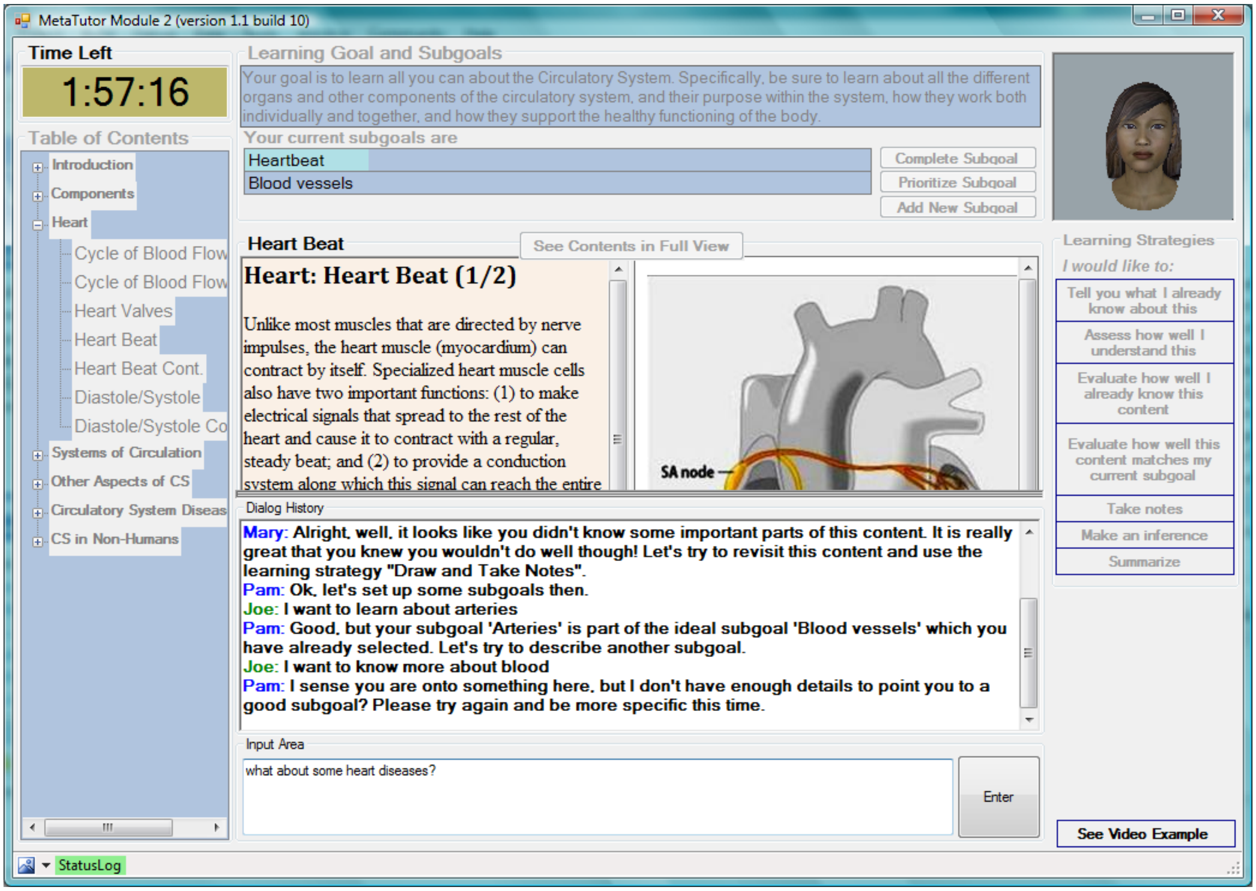


scaffolding, the Scaffolding module can turn off (not call) the Agents Technologies implementation module and rely only on the other modules for scaffolding purposes. For instance, it can simply call the Multi-modal Interface module to display the feedback the agents were supposed to utter.

We present next detailed descriptions of MetaTutors' components. The pre-planning component collects student demographic information, delivers a short quiz, and prompts students to activate prior knowledge in the form of a paragraph summarizing their knowledge on the topic to be studied, e.g., the circulatory system. In addition, pre-planning calls other modules, such as the assessment module, to evaluate the quiz responses and the student-generated paragraph, i.e., the prior-knowledge activation (PKA) paragraph. The student model module is also called to update the model based on the quiz results and evaluation of the PKA paragraph. The planning module handles the multi-step, mixed-initiative process of breaking the overall learning goal into more manageable sub-goals. It relies on the micro-dialogue manager module (see bottom part of Figure 2, Implementation Details), which handles the multi-turn interaction between the system and the student. The purpose of this call is to determine a set of accurate sub-goals. The planning module calls other modules, such as the student model module, to update variables related to sub-goal generation that are part of the student model. It calls the assessment module to assess each student-articulated sub-goal and then the feedback module to generate appropriate feedback.

The student model component maintains and updates close to 100 variables that we deem important to assess the students' mastery of the subject matter (student mental model) and SRL processes (student SRL model). One of the designing principles of the existing MetaTutor system was to collect and store in log files everything that might be related to shifts in understanding and meta-cognitive behavior in students. Every attempt was made to create an exhaustive set of variables to be tracked within the log files.

Figure 3. Sample Screenshot of MetaTutor.



Your goal is to learn all you can about the Circulatory System. Specifically, be sure to learn about all the different organs and other components of the circulatory system, and their purpose within the system, how they work both individually and together, and how they support the healthy functioning of the body.

Heart: Heart Beat (1/2)

Unlike most muscles that are directed by nerve impulses, the heart muscle (myocardium) can contract by itself. Specialized heart muscle cells also have two important functions: (1) to make electrical signals that spread to the rest of the heart and cause it to contract with a regular, steady beat; and (2) to provide a conduction system along which this signal can reach the entire

Mary: Alright, well, it looks like you didn't know some important parts of this content. It is really great that you knew you wouldn't do well though! Let's try to revisit this content and use the learning strategy "Draw and Take Notes".
Pam: Ok, let's set up some subgoals then.
Joe: I want to learn about arteries
Pam: Good, but your subgoal 'Arteries' is part of the ideal subgoal 'Blood vessels' which you have already selected. Let's try to describe another subgoal.
Joe: I want to know more about blood
Pam: I sense you are onto something here, but I don't have enough details to point you to a good subgoal? Please try again and be more specific this time.

what about some heart diseases?

Variables include the scores on quizzes given throughout a session as well as assessment of the PKA paragraphs and summaries of content pages that students write. The student model module is called by other modules as they need to retrieve or update information regarding students' level of understanding of the subject matter and SRL behavior. The assessment module evaluates various student inputs (textual, actions on the interface, time-related behavior) and sends evaluation results to other components that need these results. It uses information provided by the knowledge base module and various functions provided by the natural language processing and machine learning modules (see Figure 2). For instance, to assess a student-generated sub-goal the natural language processing module is called with the sub-goal taxonomy, which is retrieved from the knowledge base, and the student-articulated sub-goal as input parameters. The output from the natural language processing module is a vector of feature-values that quantifies the similarity between the student sub-goal and each of the ideal sub-goals in the taxonomy. The vector is then passed to a classifier in the machine learning module that classifies the student-articulated sub-goal into one of the following categories: too general, too specific, or ideal.

The scaffolding module handles the implementation of pedagogical strategies. It relies on the knowledge base, XML parser, and production rules modules of the implementation architecture. The production rules encode conditions which are monitored by the system. Through a polling mechanism, all the rules are checked at specified time intervals, e.g., every 30 seconds (this value will be calibrated based on data), to see if the conditions of a rule are met. When they are, the corresponding rule is triggered. If multiple rules fire simultaneously, a random or uniform policy (implemented using a round-robin algorithm) can be implemented. The default policy in the current system is uniform firing. The best policy is yet to be determined. The feedback module handles the type and timing of feedback provided through the PAs and other interface elements. It uses the knowledge base, XML parser, and production rules modules in the implementation. The authoring module serves the designer of the system, the subject-matter experts, and the cognitive scientists that use the system. It relies on XML editors and text editors to make changes to various configurable items in the knowledge base. The multi-modal interface module handles the complex interface between the students/experimenter/developer and MetaTutor.

The system manager controls the operation of the entire system, assuring proper communication and sequencing among all components. The Log module in the implementation view records every single action by the user and the system such that post-experiment analyses can be performed. The knowledge base module includes the content pages and other knowledge items needed throughout the system, such as the sub-goal taxonomy used during sub-goal generation in the planning module. The agents' technology module handles the four agents we have used in MetaTutor: Gavin the Guide, Mary the monitoring agent, Pam the planner, and Sam the strategizer.

MENTAL MODEL DETECTION

Self-regulation is most important when students engage in tasks that challenge them. Research has shown that complex science topics, such as the circulatory system, are difficult for students to understand (Azevedo et al., 2008; Chi et al., 2001). Often, students acquire declarative knowledge of these topics, but lack the conceptual understanding, or mental model, necessary to be successful (Azevedo, 2005; Chi, Siler, & Jeong, 2004). Mental models are cognitive representations that include the declarative, procedural, and inferential knowledge necessary to understand how a complex system functions. Mental models go beyond definitions and rote learning to include a deep understanding of the component processes of the system and the ability to make inferences about changes to the system. One way the acquisition of mental models of complex systems can be facilitated is through presenting multiple representations of information such as text, pictures, and video in a multimedia and hypermedia learning environment (Mayer, 2005). Therefore, hypermedia environments, such as MetaTutor, with their flexibility in presenting multiple representations, have been suggested as ideal learning tools for fostering sophisticated mental models of complex systems (Azevedo, in press; Goldman, 2003; Kozma, 2003).

Detecting mental model shifts during learning is an important step in diagnosing ineffective learning processes and intervening by providing appropriate feedback. One method to detect students' initial mental model of a topic is to have them write a paragraph. Cognitively, this activity allows the learner to activate their prior knowledge of the topic (e.g., declarative, procedural, and inferential knowledge) and express it in writing so that it can be externalized and amenable to computational methods of analysis. The mental model can be categorized qualitatively, and depending on the current state (e.g., simple model vs. sophisticated model), is then used by the system to provide the necessary instructional content and learning strategies (e.g., prompt to summarize, coordinate informational sources) to facilitate the student's conceptual jump to the next qualitative level of understanding. Along the way, students can be prompted to modify their initial paragraph and thereby demonstrate any subsequent qualitative changes to their initial understanding of the content. This qualitative augmentation is a key to an intelligent, adaptive hypermedia learning environment’s ability to accurately foster cognitive growth in learners. This process continues periodically throughout the learning session to examine qualitative shifts during learning.

Mental Model Coding. Due to their qualitative nature, most researchers develop complex coding schemes to represent the underlying knowledge and most often use categorical classification systems to denote and represent students' mental models. For example, Chi and colleagues' early work (Chi et al., 2001) focused on 7 mental models of the circulatory system. Azevedo and colleagues (Azevedo, 2005; Azevedo et al., 2008) extended their mental models classification to 12 to accommodate the multiple representations embedded in their hypermedia learning environments. We have re-categorized our existing 12 mental models of the circulatory system into 3 categories of low-, intermediate-, and high-mental models of the circulatory system. The rationale for choosing the 3-category mental models approach was to enhance the ability of determining students' mental models shifts during learning with MetaTutor and because the 12 mental models approach would have been too detailed of a grain size to yield reliable classifications and thus to accurately assess "smaller" qualitative shifts in students' models. Furthermore, with more mental models we would have needed substantially more instances to train our classifiers.

Experiments and Results. We have experimented with an existing dataset consisting of 309 mental model essays collected from previous experiments by Azevedo and colleagues (based on Azevedo, Cromley, & Seibert, 2004; Azevedo, 2005; Azevedo et al., 2008). These mental model essays were classified by two experts with extensive experience coding mental models. Each expert independently re-coded each mental model essay into one of the three categories and achieved an inter-rater reliability of .92 (i.e., 284/309 agreements) yielding the following new dataset: 139 low mental models, 70 intermediate mental models, and 100 high mental models.

Each item in the dataset is mapped onto a set of 8 features. There is one feature corresponding to each of the seven subgoals and one feature corresponding to all subgoals. Each feature represents the semantic overlap between the student-generated PKA and a benchmark. The benchmark can be the nodes in a taxonomy corresponding to a subgoal (feature 1-7) or all subgoals (feature 8), ideal PKA paragraphs generated by experts (there is an ideal PKA paragraph for each of the 7 subgoals; their union represents the benchmark for feature 8), or content pages corresponding to pages relevant to each of the subgoals. The relevance of each page to a subgoal has been identified by human experts. Each of the benchmarking methods (taxonomy, ideal paragraphs, content pages) can be used in combination with several semantic similarity methods. In our case, we used a cosine similarity measure based on tf-idf (term frequency-inverted document frequency) vector representations of student PKA paragraphs and the benchmarks or simple normalized overlap measures based on unigrams or bigrams.

We report results, as accuracy and kappa values, for the best combinations of methods and learning algorithms mentioned above. In Table 1, rows correspond to methods and columns to learning algorithms. A quick look at the results revealed that a tf-idf method combined with Bayes Nets leads to best overall results in terms of both accuracy and kappa values. The second best results were obtained using a combination of unigrams and/or bigrams with SVM or LR. Both SVM and LR are called function-based classifiers as they are both trying to identify a function that would best separate the data into appropriate classes, i.e. mental model types in our case. For the random baseline we obtained (accuracy=0.31, kappa=-0.06 - a kappa close to 0 means chance) based on averaging over 10 random runs while for the uniform baseline, i.e. predicting all the time the dominant class, which is the Low mental model class, we obtained (accuracy=0.45, kappa=0).

Table 1. Performance results as accuracy/kappa values

Dataset	NaïveBayes	BayesNet	SVM	LR	J48	J48graft
tf-idf	57.70/0.35	**76.31•/0.63•**	64.12/0.42	54.21•/0.28	68.22•/0.50•	71.19•/0.55•
Tax	61.44/0.39	61.93/0.37	67.18•/0.44	**69.61•/0.50•**	62.23/0.40	62.65/0.40
Ip-uni	66.39/0.48	66.14/**0.48**	67.83/0.45	65.62/0.44	65.85/0.47	65.88/0.47
Ip-bi	61.42/0.38	65.18•/0.43	**67.21**•/0.44	67.05•/0.45•	62.14/0.40	62.37/0.40

• - statistically significant improvement compared to NaïveBayes

SUBGOAL GENERATION

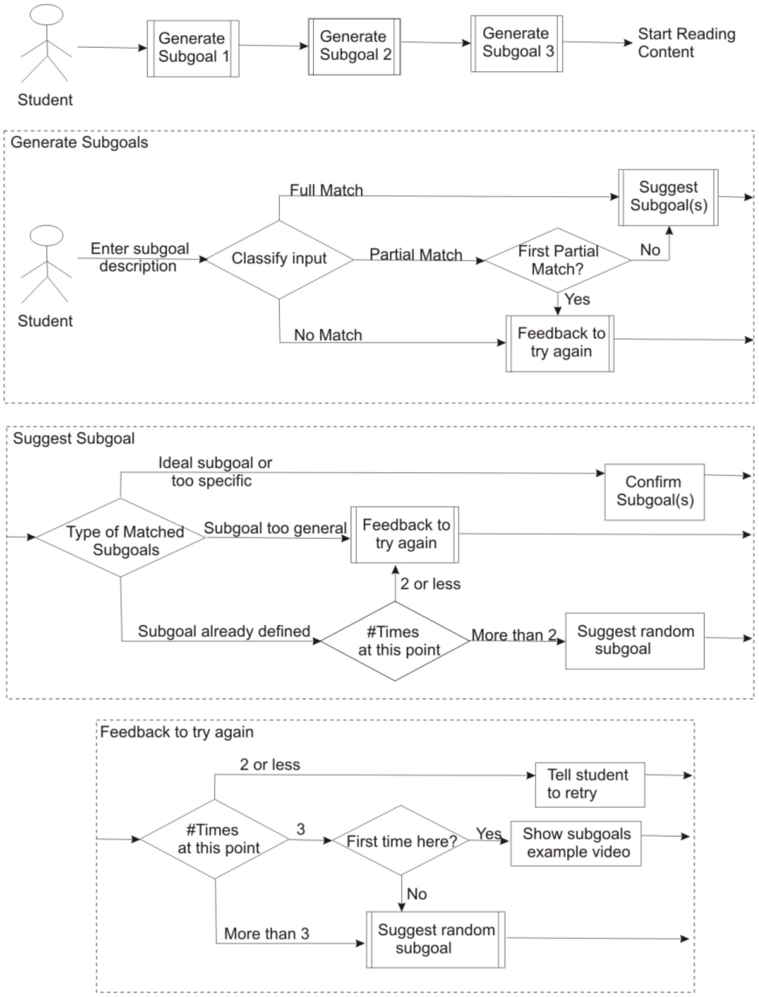
Subgoal generation is a critical step in complex learning and problem solving (Anderson & Labiere, 1998; Newell, 1994). Multi-phase models of self-regulated learning (SRL; Azevedo & Witherspoon, in press; Pintrich, 2000; Winne & Hadwin, 2008; Zimmerman, 2006) include subgoal generation as key element of planning. According to time-dependent SRL models, self-regulatory processes begin with the forethought, planning, and activation phase. During this initial phase of learning, learners create subgoals for their learning session; activate relevant prior knowledge of the content (stored in long-term memory) and perceptions about the specific learning task and the context in which they will be learning. Subgoal generation is an important phase in learning about complex science topics with non-linear, multi-representational hypermedia environments whereby the learner may be asked to spent a substantial amount of time creating a deep conceptual understanding of the topic (as measured by a sophisticated mental model). As such, asking the learner to create subgoals forces him/her to partition an overall learning goal set by the experimenter, human or computerized tutor, or teacher into meaningful subgoals that can be accomplished by integration multiple representations of information in a relatively short period of time. For example, the overall learning goal of *you have two hours to learn about the parts of the human circulatory system, how they work together, and how they support the human body* can be used to create the following subgoals - *learn about the parts*, *learn about how the systemic and pulmonary systems work together*, *functions of the circulatory system*, etc.

An intelligent tutoring system whose goal is to model and scaffold subgoal generation should include a component able of first assessing student generated subgoals and then provide appropriate feedback to help the student set an ideal set of subgoals.

In MetaTutor, a taxonomy-driven dialogue management mechanism has been implemented to handle subgoal assessment and feedback generation (see Figure 4). We organized hierarchically in a taxonomy the overall learning goal, its seven ideal subgoals as identified by human experts, and relevant keywords associated with each subgoal. In this *subgoal taxonomy* (see Figure 3), the top node is the most general while the leaves (lowest level nodes) are the most specific. The taxonomy was semi-automatically generated from the set of seven ideal subgoals and other sources such as WordNet (Miller, 1995). A student subgoal is assessed by extracting and comparing its key concepts, i.e. words or sequences of words, with entries in the taxonomy. The assessment is performed using the following dimensions:

- **Full or partial match**. If all the key words that describe a subgoal in the taxonomy are present in the student subgoal then we have a full match. Otherwise, if only some of the subgoal's key words are present in the student's input, a partial match occurs.
- **Single or multiple matches**. When a student subgoal is associated with more than one subgoal we have multiple matches. That is, the student input points to two or more different subgoals. An example of a multiple matches student subgoal is *I would learn about heart valves.* The concept of *valves* is associated with the subgoals of *heart components* and *blood vessels* (see Figure 1).
- **Specific, general, or perfect match**. An example of a perfect match is when the exact concept in the taxonomy is found in the student subgoal as in *I want to know more about blood vessels (major ones)*. This student subgoal matches the ideal subgoal *blood vessels*. In other words, the student subgoal is not too general, not too specific, and it contains all and only the words of the subgoal concept in the taxonomy. When a student subgoal only mentions concepts below the ideal level (see Figure 1) in the taxonomy, the subgoal is deemed too specific. An example of a specific student subgoal is *I need to learn more about veins.* It is too specific because *veins* is below *blood vessels*, the ideal level, in the taxonomy.

Figure 4. Overview of the Subgoal Generation process in MetaTutor.



- **Relevant or Irrelevant**. When a student subgoal does not match any concept in the taxonomy we have an irrelevant subgoal. Otherwise, we have a relevant subgoal.

Based on student subgoal assessment, a particular type of feedback will be generated. The feedback is specific to each combination of values for the above dimensions of assessment, i.e. depending on whether the subgoal is deemed a full or partial match, single or multiple matches, specific or general or perfect match, and relevant or irrelevant. For instance, when a student-articulated subgoal is deemed full match, relevant, too general, single match, then the following multi-turn dialog will take place:

> Student: I want to learn about the circulatory system
> **System**: *Alright, can you be a little more specific about what you would like to learn about the circulatory system?*
> Student: learn about the valves of the heart
> **System**: *That's good, but it's a little too specific level for our objectives today. Let's consider a broader sub-goal. Did you want to learn about 'Heart components' or 'Blood vessels'?*

Student: Heart components
System: *Wonderful! We've just setup a good subgoal. Please choose another subgoal.*

For each system turn we have a list of generic templates with placeholders for important items such as subgoals as in the following example: *Very good! Let's try to set a subgoal that covers 'Y'. How about we try to learn about 'X'.* which is used when a student-articulated subgoal *(Y)* is assessed as full match, relevant, too specific, and single match.

To evaluate our subgoal assessment method, we have experimented with a set of 258 student-generated subgoals collected from a classroom experiment in which students were asked to generate 3 subgoals for the overall learning goal of *learning about the circulatory system*. The generated subgoals were then rated by an expert with respect to which ideal subgoals students specified. The expert used the following three scores to rate each student-generated subgoal 0 - subgoal not specified, 1 - subgoal partially specified, 2 - subgoal fully specified. We compared the human judgments with computer judgments and report the results, in terms of kappa scores, in Table 2. The results are reported for each individual subgoal and also as an average over the seven subgoals.

Table 2. Kappa scores for the automated method for assessing student-generated subgoals.

Subgoal	Kappa
Bloodflow	0.76
Heartbeat	0.75
Heart components	0.76
Blood vessels	0.95
Blood components	0.77
Purposes of CS	0.69
Malfunctions	0.75
Average	0.77

CONCLUSIONS

We presented and evaluated two components of the intelligent tutoring system MetaTutor. We have found that a tf-idf method combined with the Bayes Nets algorithm provides best accuracy and kappa results on the task of mental model detection. A taxonomy-driven method to handle subgoal generation yielded very good human-computer agreement scores for subgoal assessment.

ACKNOWLEGEMENTS

The research presented in this paper has been supported by funding from the National Science Foundation (Early Career Grant 0133346, 0633918, and 0731828) awarded to the R. Azevedo and (RI 0836259, RI 0938239) awarded to Dr. Vasile Rus. We thank Jennifer Cromley, Daniel Moos, and Jeffrey Greene for data collection and analysis. We also thank Siler, Michael Cox, and Ashley Fike for data preparation.

REFERENCES

Anderson, J.R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Azevedo, R. (in press). The role of self-regulation in learning about science with hypermedia. In D. Robinson & G. Schraw (Eds.), *Current perspectives on cognition, learning, and instruction*.

Azevedo, R., & Witherspoon, A.M. (in press). Self-regulated use of hypermedia. In D. J. Hacker, J. Dunlosky, and A. C. Graesser (Eds.). *Handbook of metacognition in education.*, Mahwah, NJ: Erlbaum.

Azevedo, R., Cromley, J.G., & Seibert, D. (2004). Does adaptive scaffolding facilitate students' ability to regulate their learning with hypermedia? *Contemporary Educational Psychology*, 29, 344-370.

Azevedo, R. (2005). Computer environments as metacognitive tools for enhancing learning. *Educational Psychologist*, 40, 193-197.

Azevedo, R., Witherspoon, A., Graesser, A.C., McNamara, D.S., Rus, V., Cai, Z., & Lintean, M. (2008). MetaTutor: An adaptive hypermedia system for training and fostering self-regulated learning about complex science topics. *Annual Meeting of Society for Computers in Psychology*, Chicago, IL.

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

Chi, M. T. H., Siler, S., & Jeong, H. (2004). Can tutors monitor students' understanding accurately? *Cognition and instruction*, 2004, 22, p. 363-387.

Goldman, S. (2003). Learning in complex domains: When and why do multiple representations help? *Learning and Instruction*, 13, 239-244.

Graesser, A.C., Hu, X., & McNamara, D.S. (2005). Computerized learning environments that incoporate research in discourse psychology, cognitive science, and computational linguistics. In A. Healy (Eds.). *Experimental Cognitive Psychology and its Applications*, 59-72. Washington, D.C.: APA.

Kozma, R. (2003). The material features of multiple representations and their cognitive and social affordances for science understanding. *Learning and Instruction*, 13(2), 205-226.

Landauer, T., McNamara, D.S., Dennis, S., & Kintsch, W. (Eds). (2005). *Latent Semantic analysis: A road to meaning*. Mahwah, NJ:Erlbaum.

Mayer, R. (2005). *The Cambridge handbook of multimedia learning*. NY: Cambridge University Press.

McNamara, D.S., Boonthum, C., Levinstein, I.B., & Millis, K. 2007. Evaluating selfexplanations in iSTART: comparing word-based and LSA algorithms. *Handbook of LSA*. In Landauer, T., D.S. McNamara, S. Dennis, and W. Kintsch (Eds.). NJ: Erlbaum, 227-241.

Miller, G. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):3941.

Newell, A. (1994). *Unified Theories of Cognition.* Harvard University Press.

Pintrich, P.R. (2000). The role of goal orientation in selfregulated learning. In M. Boekaerts, P. Pintrich, and M. Zeidner (Eds.). *Handbook of self-regulation.*, San Diego, CA: Academic Press, 451–502.

Rus, V., McCarthy, P.M., Lintean, M., Graesser, A.C., & McNamara, D.S. (2007). Assessing student selfexplanations in an intelligent tutoring system. *Proceedings of the 29th annual conference of the Cognitive Science Society.* In D. S. McNamara and G. Trafton (Eds.).

VanLehn, K., Graesser, A.C., Jackson, T., Jordan, P., Olney, A., & Rose, C. (2007). When are tutorial dialogues more effective than reading? *Cognitive Science*, 31(1):3– 62.

Winne, P., and Hadwin, A. (2008). The weave of motivation and self-regulated learning. In D. Schunk and B. Zimmerman (Eds.), *Motivation and self-regulated learning: Theory, research, and applications*, Mahwah, NJ: Erlbaum, 297–314.

Zimmerman, B. (2006). Development and adaptation of expertise: The role of self-regulatory processes and beliefs. In K. Ericsson, N. Charness, P. Feltovich, and R. Hoffman (Eds.). *The Cambridge handbook of expertise and expert performance*. New York: Cambridge University Press, 705-722.