Automatic Detection of Student Mental Models During Prior Knowledge Activation in MetaTutor

Vasile Rus[1], Mihai Lintean[1], and Roger Azevedo[2]
{vrus, mclinten, razevedo}@memphis.edu
[1]Computer Science Department, The University of Memphis
[2]Psychology Department, The University of Memphis

Abstract. This paper presents several methods to automatically detecting students' mental models in MetaTutor, an intelligent tutoring system that teaches students self-regulatory processes during learning of complex science topics. In particular, we focus on detecting students' mental models based on student-generated paragraphs during prior knowledge activation, a self-regulatory process. We describe two major categories of methods and combine each method with various machine learning algorithms. A detailed comparison among the methods and across all algorithms is also provided. The evaluation of the proposed methods is performed by comparing the prediction of the methods with human judgments on a set of 309 prior knowledge activation paragraphs collected from previous experiments with MetaTutor on college students. According to our experiments, a content-based method with word-weighting and Bayes Nets algorithm is the most accurate.

1 Introduction

This paper describes automatic methods for detecting students' mental models (MM) during interaction with MetaTutor [5], an intelligent tutoring system that teaches students self-regulatory processes during learning of complex science topics. At the beginning of their interaction with MetaTutor, students are given a learning goal, e.g. *learn about the human circulatory system*, and encouraged to use a number of self-regulatory processes that will eventually help with their learning. One of the important self-regulatory processes in MetaTutor is prior knowledge activation (PKA), which involves students recalling knowledge about the topic to be learned.

During prior knowledge activation, students must write a paragraph which is assumed to reflect students' knowledge with respect to the learning goal. Excerpts from PKA paragraphs corresponding to High (H) and Low (L) mental models with respect to the goal of learning about the circulatory system are given in Table 1. The paragraphs are reproduced as typed by students. Entire paragraphs are not shown due to space reasons.

Table 1. Examples of PKA paragraphs for High (H) and Low (L) mental models (MM).

MM	PKA Paragraph
H	Circulatory system is made up of 3 parts: heart, blood and blood vessels. The heart is a muscle which pumps blood in and out to the rest of the body. ... There are 3 types of blood vessels. Artery, veins and capillaries. The arteries carry blood away from the heart, veins to the heart. ...
L	I know that we all have hearts. The heart is the main source of blood. It is the strongest and most important muscle. I know that there are arteries going (coming) out of the heart. ...

Given such a PKA paragraph, the task is to infer the student mental model. We work with three qualitative mental models: low, medium, and high. We view the task of detecting the student mental models as a standard classification 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 will be 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 assortment of the selected learning algorithms provides some diversity in terms of potential weighting and dependency patterns among the features used to model the task at hand, e.g. naïve Bayes assume total independence among features.

In order to find a good method and algorithm for inferring student mental models based on PKA paragraphs, we have investigated two categories of methods and combined them with the above six machine learning algorithms. In one category of methods, called *content-based*, student-generated PKA paragraphs are automatically compared with various sources of knowledge describing the learning goal. The sources can be (1) a collection of pages that describe the goal, (2) a taxonomy that includes the major concepts related to the goal, or (3) ideal/expected paragraphs, written by human experts, describing the learning goal and its subgoals. The second category of methods, called *word-weighting*, maps student-articulated PKA paragraphs onto a set of features in which individual words act as features and the corresponding values are weights derived using distributional information of the words across a corpus of documents (in our case the PKA paragraphs). This latter method resembles traditional text classification models [14] in that it uses individual words as features (some classification models also use the position of the words in the documents). In addition to all the above methods, we also experimented with two baseline algorithms random guessing and uniform guessing, i.e. guessing all the time the dominant category in the training data.

The rest of the paper is structured as follows. *Background* presents the mental models in MetaTutor and previous work on automatic student input assessment. The subsequent section, *Methods*, describes in detail the methods we proposed whereas *Experimental Setup and Results* presents performance figures, lessons learned, and also outlines plans for the future. The *Conclusions* section ends the paper.

2 Background

MetaTutor is an adaptive hypermedia learning environment that is designed to detect, model, trace, and foster students' self-regulated learning about human body systems such as the circulatory, digestive, and nervous systems [5]. Theoretically, it is based on cognitive models of self-regulated learning [1, 17]. The underlying assumption of MetaTutor is that students should regulate key cognitive and metacognitive processes in order to learn about complex and challenging science topics. The design of MetaTutor is based on extensive research by Azevedo and colleagues' showing that providing adaptive

human scaffolding, that addresses both the content of the domain and the processes of self-regulated learning, enhances students' learning about challenging science topics with hypermedia [2, 3, 4, 5, 10]. Overall, their research has identified key self-regulatory processes that are indicative of students' learning about these complex science topics. More specifically, they include several processes related to planning (e.g., generating sub-goals), metacognitive monitoring processes (e.g., feeling of knowing, judgment of learning), learning strategies (coordinating information sources, summarization), and methods of handling task difficulties and demands (e.g., time and effort planning).

2.1 Mental Models

Mental models are mental 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. The acquisition of mental models of complex systems can be facilitated through presenting multiple representations of information such as text, pictures, and video in hypermedia learning environments [12]. 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 [1, 8].

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. A mental model can be categorized qualitatively, and depending on the current state (e.g., simple model vs. sophisticated model), is then used by the hypermedia system to provide the necessary instructional content and learning strategies (e.g., prompt to summarize, coordinate informational sources) to facilitate the student's conceptual shift 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.

2.2 Mental Models 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 [7] focused on 7 mental models of the circulatory system. Azevedo and colleagues [1] extended their mental models classification to 12 to accommodate the multiple representations embedded in their hypermedia learning environment. In this paper, we

have re-categorized our existing 12 mental models of the circulatory system (see [10] for the details) 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.

2.3 Previous Work on Evaluating Natural Language Student Input in Intelligent Tutoring Systems and Automated Essay Grading

Researchers who have developed tutorial dialogue systems in natural language have 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 [9, 16], and the iSTART system, which helps students read text at deeper levels [13]. Systems such as these have typically relied on statistical representations, such as latent semantic analysis (LSA; [11]) and content word overlap metrics [13]. LSA has the advantage of representing 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. More recently, a lexico-syntactic approach, entailment evaluation [15], has been successfully used to meet the challenge of natural language understand and assessment in intelligent tutoring systems. The entailment approach has been primarily tested on short student inputs, namely individual sentences. Both LSA and the entailment approach pose some challenges for evaluating the PKA paragraphs we have to handle. LSA requires the construction of a LSA space based on a large collection of documents from the domain of interest, i.e. the circulatory system. Collecting such tests is a time consuming task. Also, LSA suffers from the text-length confound which means using it for handling paragraph-length texts would lead to high similarity scores, probably resulting in many false positives. The entailment approach has been designed for sentence-to-sentence relation and thus it is not trivial to extend it to handle paragraph-to-paragraph tasks as it requires the use of a syntactic parser which operates on one sentence at a time. We do plan to extend it to handle paragraph-to-paragraph textual relation detection using coreference resolution components that will link concepts across sentences for a paragraph-level meaning representation. For the time being, we opted instead for a set of methods that combine simple textual overlap features with machine learning algorithms to automatically infer student mental models. We take advantage of the goals and subgoals in MetaTutor when choosing the features to be used in our solution to the student mental model detection problem, as explained later.

The problem of detecting student mental models from PKA paragraphs is related to the task of *automated essay scoring* (AES), i.e. automatically evaluating and scoring written texts. The purpose in AES is to improve time, cost, reliability and generalizability of the process of writing assessment. Dikli [19] gives a fairly comprehensive survey of AES systems. AES systems require training , i.e. human-scored written texts, and rely on form and content features to score written texts. They do not really understand the texts or

emulating the human scoring process. One difference between AES and MM detection is that the length of the input is different. Usually, in AES essay-long texts, which are comprised of many paragraphs, are considered while in our task of MM detection we work with smaller, paragraph-length texts. AES systems use the multi-paragraph structure of essays as part of the scoring algorithm while in the MM detection problem this structural information is less important. The content-based components of the AES systems could be used for the MM detection task. Some of our proposed methods resemble some of the content-based methods employed in AES systems (see the word weighting in the vectorial representation used in E-rater, which is described in [19]).

3 Methods

All the methods we implemented, except the baselines, have two major steps. The first step consists of data processing and feature extraction. The details of this step are specific to each method and will be described later. During a second step, we used machine learning algorithms to induce various classifiers for categorizing PKA paragraphs into high, medium, and low mental models. We experimented with the six machine learning algorithms mentioned earlier. It is beyond the scope of this paper to discuss in detail these algorithms (see [14, 18] for details). We used the implementation of the algorithms from WEKA, a machine learning toolkit [18]. The algorithms were run with the default parameters, e.g. SVM was run with the polynomial kernel. There is a large parameter space for these learning algorithms and we plan to tweak these parameters in the future in order to further investigate their behavior for our problem. For this paper, the machine learning phase was used to check the effectiveness of the preprocessing phase and of the chosen set of features and methods.

The performance of all the methods was evaluated using 10-fold cross validation. In k-fold cross-validation the available data set is split into k folds. Then, one fold is kept for testing and the other $(k - 1)$ are used for training. This process is repeated for each fold resulting in k trials. The reported performance is then computed as the average of the individual trials' performances. When $k = 10$ we have 10-fold cross-validation. To further increase the confidence in the estimated values of the reported accuracy, we have run 10-fold cross-validation 10 times, each time with a different seed value, which is an input parameter to k-fold cross-validation evaluation. The seed value affects the way instances in the data set are selected for the individual folds. Thus, for each method and learning algorithm we compute $10 * 10 = 100$ performance scores and then take the average. The advantage of running 10-fold cross-validation 10 times with different seed points is that each instance in the original data set is evaluated 10 times. By comparison, a 100-fold cross-validation would result in each instance being evaluated once. We also ran paired t-tests among different methods and learning algorithms in order to check if differences in performance are statistically significant. We report performance in terms of accuracy and kappa coefficient. Accuracy is the percentage of correct predictions out of all predictions. Kappa coefficient measures the level of agreement between predicted categories and expert-assigned categories while also accounting for chance agreement.

3.1 Content-based Methods

The methods in this category rely on the presence of key concepts related to the learning goal in the student-articulated paragraphs. The key concepts are specified in different ways for the three methods in this category and it is in this aspect that the methods differ. The key concepts are specified in the three methods using the following benchmarks, respectively: (1) expert-created domain taxonomy, (2) original pages of content, and (3) expert-generated ideal descriptions of the learning goal and its subgoals.

For all three methods, 8 features are computed: one feature corresponding to the overall learning goal and one feature for each of the 7 subgoals. The value of each feature represents the percentage of words in the entire benchmark (for the feature corresponding to the overall learning goal) or parts of the benchmark corresponding to subgoals (for subgoal features) that are present in the student-generated PKA paragraphs. For instance, for the *taxonomy-based method* (*tax* in Table 2) a taxonomy of concepts is the benchmark. The overall goal, i.e. learn about the circulatory system, is the top node of the taxonomy (see Figure 1). The seven subgoals are the nodes in the *ideal level* in the Figure 1. The parts of the taxonomy benchmark corresponding to subgoals are the subtrees below the subgoals nodes in the taxonomy. We use nodes in these subtrees to compute the values corresponding to the 7 subgoal-related features. The advantage of the taxonomy-based method is its simplicity and small computational costs as the taxonomy only includes several dozen concepts. The trade-off is the expert associated costs to build the taxonomy. In MetaTutor, the taxonomy was needed for assessing and feedback during another self-regulation process, subgoal generation, and thus there is no extra effort to build the taxonomy specifically for mental model detection.

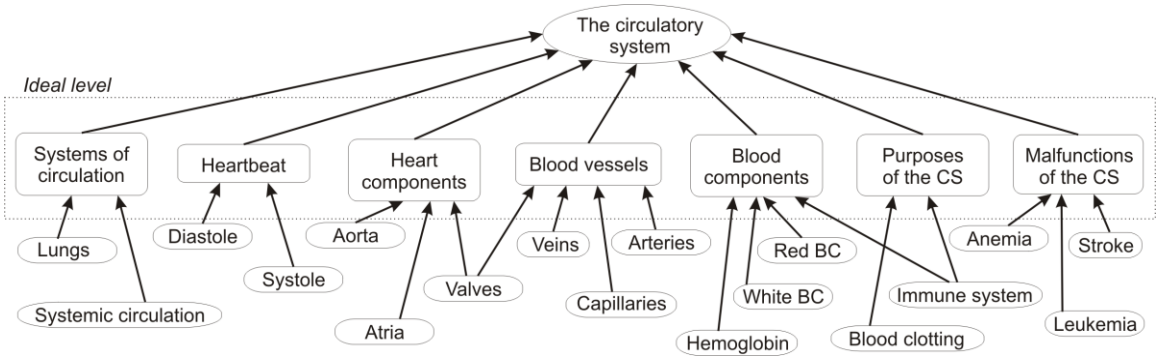

Figure 1. Partial Taxonomy of Topics in Circulatory System.

N-grams methods are very similar to the taxonomy-based method. Instead of using the taxonomy to identify key concepts relevant to the learning goal or subgoals, we used the subset of content pages related to the overall goal or subgoals, respectively. The values for the features are computed as the percentage of N-grams, i.e. sequences of N consecutive words, in the benchmark, or parts of it for subgoal features, that are present in the PKA paragraphs. In this method, it is necessary to know which page is relevant to which subgoal. An expert mapped each individual page onto each subgoal. Also, to generate the N-grams the pages and PKA paragraphs are pre-processed: stop words are eliminated and the remaining words are lowercased and stemmed. Stop words are very

frequent words such as determiners, e.g. *the*. Stemming is the process of mapping all morphological variation of a word to its base form, e.g. *hearts* and *heart* are mapped to *heart*. We used both unigrams (*uni*) and bigrams (*bi*) to compute content overlap. We also experimented with a combined method in which both bigrams and unigrams are used (*uni-bi*). Bigrams have the advantage (over unigrams) of capturing some word order, i.e. syntactic information. The N-grams methods have the advantage of needing no extra structures, e.g. expert-built taxonomies, to generate the features. We simply used the original content pages about the circulatory system from Encarta, which are used in MetaTutor. On the other hand, there is need for an expert to specify which content page is relevant to which subgoal. The biggest disadvantage of the N-gram method is their use of too much content to compare against, e.g. bigrams from all the content pages for the overall goal feature, as opposed to a set of well-selected key concepts from a taxonomy as is the case with the taxonomy-based method.

In the last method in this category, called *expectation-based*, we started by asking domain experts to generate ideal descriptions for each of the seven subgoals. These descriptions are short textual paragraphs comprising of 5-7 sentences. The collection of all paragraphs for the 7 subgoals is used to derive the eighth feature corresponding to the overall learning goal. The values of the features are generated using unigram and bigram overlap between the ideal paragraphs and the student PKA paragraphs. In this method (labeled *ip* - ideal paragraphs - in Table 2), there is no need for creating a crisp taxonomy of concepts and decide which concepts is directly related to which concept. The effort to create the ideal paragraphs is less compared to building a taxonomy for instance.

3.2 Word-weighting Methods

In this category of methods, we select from each paragraph all the words that have minimum 4 letters (when all words were used performance results were slightly worse), excluding the stop words. The selected words are then converted to lower case and stemmed. The resulting set of words is used to describe the paragraphs, i.e. they are the features. Each feature is weighted using *tf-idf* (term frequency-inverted document frequency), which captures the importance of the corresponding feature for a given paragraph. Inverted document frequency (idf) is computed as the inverse of document frequency, which is the number of documents a term occurs in from a collection of documents. In our case, document frequency is the number of prior knowledge-paragraphs a term occurs in. Term frequency, *tf*, is the number of occurrences of a term/word in a document, i.e. a PKA paragraph. As a result, a total of 1038 features are extracted and used to describe each instance in data set. Other weighting schemes, besides tf-idf, could be used but the tf-idf proved to be successful in a number of other applications [6] which is the reason we chose it.

4 Experimental Setup and Results

4.1 The Dataset

In this paper, we have experimented with an existing dataset consisting of 309 mental model essays collected from previous experiments by Azevedo and colleagues (based on

[2, 3]). The dataset consisted of entries from senior high school students and non-biology college majors. 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 for this paper: 139 low mental models, 70 intermediate mental models, and 100 high mental models. The coders included a nurse practitioner and a high school biology teacher.

4.2 Results

We report results for all combinations of methods and learning algorithms mentioned earlier. In Table 2, rows correspond to methods and columns to learning algorithms. An analysis of 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 = 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 = 45%, kappa = 0).

Table 2. Performance results as accuracy(%)/kappa values

Method	NB	BNets	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
Uni	63.65/0.45	62.97/0.44	67.57/0.45	70.03*/0.52	64.65/0.43	64.52/0.43
Bi	66.14/0.47	64.75/0.46	70.09/0.49	70.64*/0.52	63.40/0.41	63.56/0.41
uni-bi	65.43/0.47	63.63/0.45	68.79/0.46	70.22/0.52	68.93/0.49	68.89/0.49
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
ip-uni-bi	64.94/0.45	64.53/0.46	67.05/0.43	66.83/0.46	65.40/0.46	65.66/0.46

Based on a more careful analysis of the results in Table 2, we found that given a method the choice of the machine learning algorithm is important. Looking at the results within each group of methods one can notice the relative large range of the performance figures. For instance, the accuracy values for the tf-idf method vary most from 57.70% for naive Bayes to 76.31% for Bayes Nets. For Bayes Nets the Weka's default K-2 search algorithm was used. This variability indicates that this method is more sensitive with respect to the choice of the machine learning algorithm. We call such methods less stable. One possible explanation for the variability of the tf-idf method could be its large number of features used (1038) relative to the number of instances (309). This is not unusual for text classification as, for instance, a typical naive Bayes method [14] uses not only all the words in the documents to be classified but also their positions leading to a very large number of features. The last three groups of methods in Table 2 also show variability but

they seem more stable as the range of the values is somehow smaller. The most stable methods are the ideal paragraph-based methods and the unigram/bigram methods. As unigram/bigram methods provide better results than the paragraph-based methods we could say that the former offer the best of performance and stability across various machine learning schemes. We plan to conduct a study on the stability of the tf-idf method once more PKA paragraphs are available from future MetaTutor experiments. Given its best performance overall, if we can show that this method is also stable if more training data is available - as we suspect - it would be a very important finding.

5 Conclusions

We presented and evaluated several methods for detecting student mental models in the intelligent tutoring system MetaTutor. We have found that a tf-idf method combined with a Bayes Nets algorithm provides the best accuracy and kappa values. Bigram-based methods combined with Logistic Regression or Support Vector Machines provide competitive results. In addition, bigram-based methods seem to be less sensitive to the choice of the machine learning algorithm compared to the tf-idf method. It is believed that tf-idf methods would be more stable if more training data would be available.

Acknowledgments

This research was supported by funding from the National Science Foundation awarded to R. Azevedo (0133346, 0633918, and 0731828) and V. Rus (0836259). We thank Amy Witherspoon, Emily Siler, Michael Cox, and Ashley Fike for data preparation.

References

[1] 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*.

[2] Azevedo, R., Greene, J.A., & Moos, D.C. (2007). The effect of a human agent's external regulation upon college students' hypermedia learning. *Metacognition and Learning, 2*(2/3), 67-87.

[3] Azevedo, R., Moos, D.C., Greene, J.A., Winters, F.I., & Cromley, J.C. (2008). Why is externally-regulated learning more effective than self-regulated learning with hypermedia? *Educational Technology Research and Development, 56*(1), 45-72.

[4] Azevedo, R., & Witherspoon, A. Self-regulated learning with hypermedia. (in press). In Graesser, A., Dunlosky, J., & Hacker D. (Eds.), *Handbook of metacognition in education*, in press. Manwah, NJ: Erlbaum.

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

[6] Baeza-Yates, R. & Ribeiro, B. *Modern Information Retrieval*, Addison-Wesley, 1998.

[7] Chi, M. T. H., Siler, S., Jeong, H., Yamauchi, T., & Hausmann, R. Learning from human tutoring. *Cognitive Science*, 2001, 25, p. 471-534.

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

[9] Graesser, A.C., Hu, X., & McNamara, D.S. Computerized learning environments that incorporate research in discourse psychology, cognitive science, and computational linguistics. In Healy, A., ed., *Experimental Cognitive Psychology and its Applications*, 2005, p. 59-72. Washington, D.C.: APA.

[10] Greene, J.A. & Azevedo, R. A macro-level analysis of SRL processes and their relations to the acquisition of sophisticated mental models. *Contemporary Educational Psychology*, 2009, 34, p. 18-29.

[11] Landauer, T., McNamara, D.S., Dennis, S. & Kintsch, W. (Eds), *Latent Semantic analysis: A road to meaning*, 2007, Mahwah, NJ:Erlbaum.

[12] Mayer, R. *The Cambridge handbook of multimedia learning*, 2005, NY: Cambridge University Press.

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

[14] Mitchell, T. *Machine Learning*, McGraw Hill, 1997.

[15] Rus, V., McCarthy, P.M., Lintean, M.C., Graesser, A.C., & McNamara, D.S. Assessing student self-explanations in an intelligent tutoring system, In D. S. McNamara and G. Trafton (Eds.), *Proceedings of the 29th the Annual Conference of the Cognitive Science Society*, 2007.

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

[17] Winne, P. & Hadwin, A. The weave of motivation and self-regulated learning. In Schunk, D. & Zimmerman, B. (Eds.), *Motivation and self regulated learning: Theory, research and applications*, 2008, p. 297-314. NY: Taylor & Francis.

[18] Witten, I. H. & Frank, E. *Data Mining: Practical machine learning tools and techniques, 2nd Edition*, Morgan Kaufmann, San Francisco, 2005.

[19] Dikli, S. An Overview of Automated Scoring of Essays. *Journal of Technology, Learning, and Assessment*, 2006, 5(1).