Knowledge Extraction from Metacognitive Reading Strategies Data Using Induction Trees

Christopher Taylor
Department of Computer Science
The University of Texas at Tyler
Tyler, Texas, USA

Arun Kulkarni
Department of Computer Science
The University of Texas at Tyler
Tyler, Texas, USA

Kouider Mokhtari
School of Education
The University of Texas at Tyler
Tyler, Texas, USA

Abstract—**The assessment of students' metacognitive knowledge and skills about reading is critical in determining their ability to read academic texts and do so with comprehension. In this paper, we used induction trees to extract metacognitive knowledge about reading from a reading strategies dataset obtained from a group of 1636 undergraduate college students. Using a C4.5 algorithm, we constructed decision trees, which helped us classify participants into three groups based on their metacognitive strategy awareness levels consisting of global, problem-solving and support reading strategies. We extracted rules from these decision trees, and in order to evaluate accuracy of the extracted rules, we built a fuzzy inference system (FIS) with the extracted rules as a rule base and classified the test dataset with the FIS. The extracted rules are evaluated using measures such as the overall efficiency and Kappa coefficient.**

Keywords—Metacognitive Reading Strategies; Classification; Induction Tree; Rule Extraction; Fuzzy Inference System

I. Introduction

A student in any field has many ways of learning and being taught. In order to improve these methods, it is important to understand students' levels of metacognitive knowledge and skills. Metacognition, briefly defined as the awareness and understanding of one's thought processes, is fundamental in developing effective reading comprehension and problem solving skills and strategies. These skills show how well students are able to solve complex tasks such as reading comprehension. There is agreement among researchers that variability in reader characteristics can be used to partially explain individual differences in reading comprehension performance [1]. The process of reading is greatly influenced by the beliefs, attitudes, and values that readers possess. We know, for instance, that how students feel and know about their own cognitive and metacognitive abilities and skills affects whether they succeed or fail in school. Indeed, the development of metacognitive beliefs about reading and the understanding of the parameters and complexities involved in reading tend to develop whenever and wherever students receive instruction in reading.

Various methods have been used to gather data on metacognitive awareness. In order for the data to be useful, knowledge must be extracted from a dataset. Knowledge discovery in databases (KDD) is the process of analyzing data to find patterns and useful information that can be used to gain knowledge from the data. Data mining is a step in the KDD process, where rules and patterns can be extracted from data for a given purpose. Data mining involves classification, regression, clustering, and rule generation among many things [2]. Selecting a method to evaluate a dataset depends largely on the type of data to be processed. Using these methods, information can be extracted in the form of rules. A rule states how different attributes are correlated with one another in a dataset. There are several different methods that can be used to extract rules. These include the black-box method, link tracing in neural networks, decision trees, and association rules. In a black-box method the system receives inputs and produces outputs without revealing to the user the complex workings of the algorithm or requiring them to have some knowledge of how to operate it. This can prove a benefit for many fields of work where specific calculations need to be performed on large, complex datasets. Malone et al. [3] have used Kohonen network for data mining and have used Kohonen feature maps to formulate rules. Fung et al. [4] used Support Vector Machines (SVM) to extract rules from datasets by expressing the variable space as hyper-cubes. Ali et al. [5] have shown it is possible to extract useful rules using decision tree induction by suggesting improvements to the existing C4.5 decision tree algorithm. Zhou et al. [6] have shown that neural networks are able to extract rules from datasets by creating ensembles of multiple neural networks that can work together to classify data.

Anderson et al. [7] have used neural networks to identify students' levels of metacognitive awareness using data collected via the Metacognitive Awareness of Reading Strategies Inventory (MARSI). In this work we have used an induction tree to analyze a new MARSI dataset. The main advantage of induction trees is that an induction tree is able to take data from the instrument with little or no modification and process it resulting in clear, simplified rules that do not require a special knowledge or other skill to understand. A fuzzy inference system (FIS) can be built with the extracted rules as a rule-base. The FIS can further enhance the understanding of the study by providing additional information relating to how the different sets of relationships interact with each other. The FIS has been used widely in the medical field to study ailments such as cancer [8], preventing heart attacks [9] and classification of heart data [10]. It has also been applied to other fields, such as image steganography, the process of hiding information in images [11], and gas and oil consumption [12]. The FIS has proven to be a useful tool that is able to classify unknown data quickly where it is impractical to use human experts. In this research work, we seek to uncover relationships among student variables such as perceptions of self as a reader in relation to their levels of metacognitive strategy awareness and perceived use of reading strategies using an ID3 induction tree. Four different induction trees were built using four sets of features. Randomly selected half samples were used to construct the induction trees and remaining half samples were classified to evaluate accuracy of classification. Classification rules in the form of knowledge were extracted from the trees. In order to validate each set of the extracted rules, we built a Fuzzy Inference System (FIS) with the extracted rule as rule base and reclassified test samples. Section 2 describes the methodology used that is the induction tree and the FIS. Section 3 deals with results and discussions, and Section 4 provides conclusions.

II. Method

We applied C4.5 induction trees to extract rules from a dataset consisting of results from an instrument distributed to undergraduate college students to assess their metacognitive awareness and use of reading strategies. These rules were then tested for accuracy using a fuzzy inference system. The purpose was to extract students' metacognitive knowledge about reading using a metacognitive awareness strategies inventory. Specifically, we wanted to identify relationships between categories to better understand how different reading strategies relate to each other and affect overall reading skills. This information will be valuable in helping students understand their metacognitive awareness and creating teaching and learning programs designed to improve on these skills.

A. MARSI Dataset

In this study, we used a set of reading strategies data using the Metacognitive Awareness of Reading Strategies Inventory [13], which was administered to a group of college freshmen and sophomores enrolled in a community college in the south central US. The original dataset had 1811 records, which were pruned down to 1636 to accommodate missing or incomplete data. It is worth noting that the sample size is sufficient in light of the main objective of the study, which is to extract students' metacognitive knowledge about reading using a metacognitive awareness of reading strategies inventory. The size of the data set used is also consistent with similar data sets used in prior studies exploring students' metacognitive knowledge about reading [14]. The instrument consists of thirty questions designed to assess students' level of awareness or perceived use of reading strategies by classifying the questions into different categories. The questions assess what kind of reading strategies a student uses while reading conventional academic texts. Depending on the responses, ranging from 'never or almost never' to 'always or almost always', the students can be placed in three different reading categories and in a combined overall category. These categories cover three broad areas of strategies including (a) Global Reading Strategies (GLOB), which can be thought of as reading strategies used when preparing to read text (e.g., setting purpose for reading, previewing text content, predicting what the text is about); (b) Problem-Solving Strategies (PROB) which are typically used during reading when problems develop in understanding textual information (e.g., checking one's understanding upon encountering conflicting information, re-reading for better understanding); and (c) Support Reading Strategies (SUP), which scaffold or support the process of reading and text understanding (e.g., use of reference materials like dictionaries and other support systems). These three categories of reading strategies contribute to a calculation of a student's overall reading strategy score. Using the scores of these categories, a student can be classified into a 'Low', 'Medium', or 'High' category with respect to their levels of reading strategy awareness and perceived use of reading strategies when reading academic texts.

B. Data Analysis

We used C4.5 induction tree, a variation of the ID3 induction tree, to analyze the datasets. Both the ID3 and C4.5 induction trees were proposed by Quinlan [15]. ID3 and C4.5 are very similar methods, but have a few differences. For example, the C4.5 algorithm allows the usage of both continuous and discrete attributes, whereas the ID3 algorithm has difficulty dealing with continuous data since it is more intensive to find a proper split on this kind of attribute [16]. Tree classifiers use supervised learning methods to organize data results into a hierarchical tree, with each node correlating to a different attribute. The possible values of each attribute become the branches that lead to child nodes. Each node acts as a separate decision, and leads to a class at bottom of the tree, or the leaves. These trees act as multi-stage classifiers and are more efficient than single-stage classifiers since decisions are made at multiple levels and reduce the computational load [17]. By selecting a leaf node and traversing up the tree recording attributes and decision values until the root is reached, the rule can be created by listing those conditions. The ID3 Induction tree algorithm has proven to be effective when working with large datasets that have a large number of features where it is inefficient for human experts to process. These rules are also clear and easy to understand to the average user. Induction trees also have low rates of error when classifying data with noise as long as the noise rate is not extremely high. When dealing with errors in a single attribute or multiple attributes, the tree is still able to find enough information to branch on, even if the error rate of the data are high [18]. While simple decision trees for small datasets can be created quickly by a user, large datasets with many attributes would make user creation less than ideal. Induction trees can handle large datasets with multiple attributes easily with little computational power needed to produce a simple decision tree.

To make a decision tree, the amount of information needed to classify the dataset is calculated. Then, the amount of information needed to classify the dataset after a split using each attribute is calculated. The information gain is defined as the difference between information needed to classify before the split and after the split. The attribute with highest information gain is used for the split at the root node. The process is then repeated with the remaining attributes until all are processed and the tree is grown.

C. C4.5 Decision Tree

We used C4.5 algorithm to extract rules and information from the dataset. In the preprocessing stage we converted attribute values that were continuous to discrete values. After pre-processing, data can be processed with the C4.5 algorithm using the programming language R, a statistical computing environment. Using J48, a function of the R package RWEKA created by Hornik et al., [19], C4.5 decision trees can be generated that take a formula and a dataset as input. Since the data are ready, a formula is now required. This formula tells the algorithm how the dataset attributes relate to one another. Once the data and the formula are prepared, the algorithm can start to calculate information gain. For each attribute, the information gain is calculated. Equation (1) can be used to calculate entropy.

$$entropy(D) = -\sum_{i=1}^{m} p_i \log(p_i) \tag{1}$$

Where D is the observation vector, m is the number of classes, and p_i is the probability that D belongs to a given class. The information gain is calculated by subtracting the difference in entropy from the total amount of information contained in the data using (2).

$$Gain(A) = Info(D) - Info_A(D) \tag{2}$$

Where A is the attribute being processed. The total amount of information can be calculated by using (3).

$$Info_A(D) = \sum_{j=1}^{v} \frac{|D_j|}{|D|} \times Info(D_j) \tag{3}$$

Where v is the number of distinct values of attribute A, and $|D_j|/|D|$ shows the weight value of the j^{th} split of the tree. Once the information gain for each attribute is known, the attribute with the highest information gain becomes the root node in the tree and is the first split. Next, the process is repeated again for the remaining nodes and another node of the tree is created and split upon using the highest available information gain. These nodes branch out using the different conditions of the previous nodes to begin narrowing down the data with each split on information gain creating nodes that are children of their parent node. This process continues until one of a few conditions is met. If all of the records in the list at this point belong to the same class, a leaf node is created that states the class. If there is no information gain on any of the attributes or a new class is encountered for the first time, the algorithm creates a node higher up the tree to represent the expected value of the class at that point. Once all attributes have been processed, the tree is built and rules can be extracted.

Extracting rules from a full decision tree is not a simple task. Simple trees have few leaves that classify all the data. Since a large number of leaves are generated for the datasets used in this project, a selection method is taken to reduce the number of leaves to a reasonable amount. The J48 function has a default value of 25% confidence for pruning, meaning that at least 1/4th of the samples must be correctly classified for the rule to appear in the tree. The strongest rules showing the highest sample collections for each class were taken so that the rule base would be balanced for all the classes. Once the strongest rules for each class have been selected, the rule is created by listing the conditions that occur on the path from the root node to the leaf that represents the resultant class. These rules take on the format of "if *x*=*a* AND *y*=*b* then *class*=*z*" with *x* and *y* being different attributes, a and *b* being possible values for those attributes, and *z* being a possible class. Next, these rules can be used to predict the outcome of samples taken from new data that match the conditions in the rules.

D. Fuzzy Inference System

In order to evaluate the accuracy of extracted rules, a fuzzy inference system (FIS) was built using the extracted rules and the data samples were reclassified using the FIS. A FIS is a system that applies fuzzy logic to map inputs to outputs, functioning similar to an artificial neural network [20]. Fuzzy inference systems attempt to build models that can be used to predict new data. These systems allow us to model the behaviors of complex systems using rules made up of basic logic statements and then use those rules to simulate the effects on new data. In this project, the software MATLAB and the fuzzy logic toolbox is used to build a FIS.

A fuzzy inference system (FIS) essentially defines a nonlinear mapping of the input data vector into a scalar output using fuzzy rules. The mapping process involves input/output membership functions, fuzzy logic operators, fuzzy if-then rules, aggregation of output sets, and defuzzification. A FIS with multiple outputs can be considered as a collection of independent multi-input/single output systems. A general model of a fuzzy logic system (FLS) is shown in Figure 1 [21]. The FLS maps crisp inputs into crisp outputs. It can be seen from Figure 1 that the fuzzy logic system contains four components: the fuzzifier, inference engine, rule base, and defuzzifier. The rule base contains linguistic rules that are provided by experts. It is also possible to extract rules from numeric data. Once the rules have been established the FLS can be viewed as a system that maps an input vector to an output vector. The fuzzifier maps input numbers into corresponding fuzzy memberships. This is required in order to activate rules that are in terms of linguistic variables. The fuzzifier takes input values and determines the degree to which they belong to each of the fuzzy sets via membership functions. The inference engine defines mapping from input fuzzy sets into output fuzzy sets. It determines the degree to which the antecedent part is satisfied for each rule. If the antecedent part

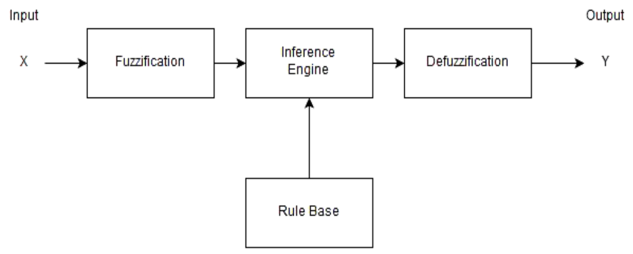

Fig. 1. Block Diagram of a fuzzy inference system (FIS)

of a given rule has more than one clause, fuzzy operators are applied to obtain one number that represents the result of the antecedent for that rule. It is possible that one or more rules may fire at the same time. Outputs of all rules are then aggregated. During aggregation fuzzy sets that represent the output of each rule are combined into a single fuzzy set. Fuzzy rules are fired in parallel, this is one of the important aspects of a FLS. In a FLS, the order in which rules are fired does not affect the output. The defuzzifier maps output fuzzy sets into a crisp number. Given a fuzzy set that encompasses a range of output values, the defuzzifier returns one number, thereby moving from a fuzzy set to a crisp number. Several methods for defuzzification are used in practice. They include: the centroid, maximum, mean of maxima, height, and modified height defuzzifier. The most popular defuzzification method is the cendroid defuzzification method. It calculates and returns the center gravity of the aggregated fuzzy set.

Fuzzy inference systems employ rules. However, unlike conventional expert systems, a fuzzy rule localizes a region of space along the function surface instead of isolating a point on the function surface. For a given input more than one rule may fire, in a FLS multiple regions are combined in the output space to produce a composite region. A general schematic of a FLS is shown in Figure 2 [17].

Creating a FIS in software consist of three primary steps: entering inputs, outputs, and rules. The inputs are the attributes of the dataset and are represented graphically in the FIS by mapping each one with a set of membership functions. For each possible value of each specific input, a membership function showing the degree of membership to a set of values is created and mapped. This fuzzification of values allows us to define how specific inputs relate to the memberships of each of the values. After each input value has been mapped, a graph similar to Figure 3 is produced. Next, the outputs are mapped in a similar fashion, only using the final possible classes for each membership function as seen in Figure 4. Lastly, the rulebase must be entered into the system. These rules are the ones obtained from the C4.5 algorithm when it was applied to the dataset. The preprocessing that was applied to the dataset earlier now proves useful when entering rules in to the FIS, where the rules must be entered using a verbose representation.

With the inputs, outputs, and rules programmed into the FIS, data can now be passed into the system to observe the -

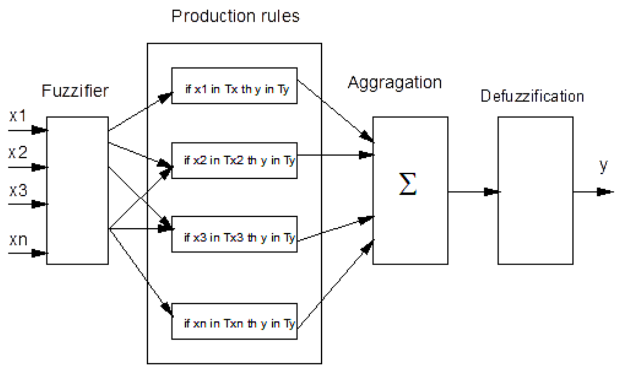

Fig. 2. Schematic diagram of a FIS

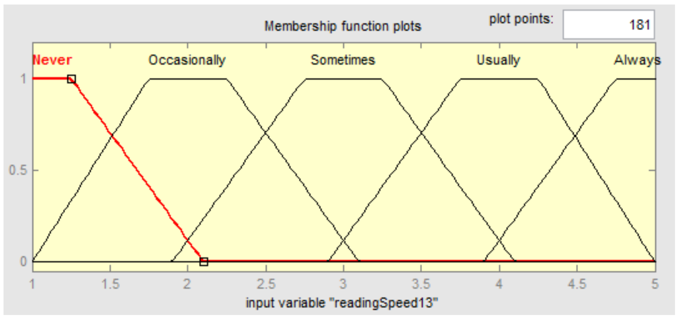

Fig. 3. Example Input Membership Functions

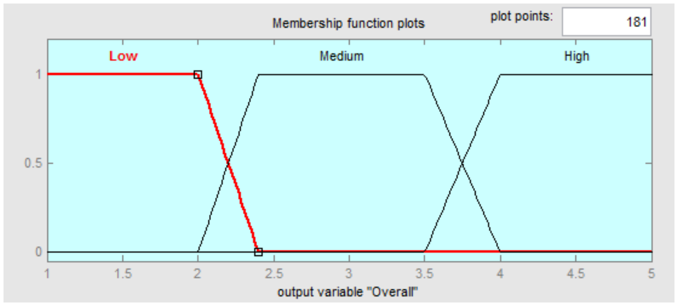

Fig. 4. Example Output Member Functions

behavior of the rules. From here, the fuzzifier component of the FIS evaluates each input into the system and finds the firing strength of each rule. The firing strength of a rule is a measure of how accurately the inputs match the conditions of the rule. Each input is matched against all rules and receives a numerical output for each rule whose conditions are all satisfied. The output of each rule is mapped to the membership functions of the output variable. This creates a shape in the membership function that shows the degree of membership as seen in Figure 5. Several rules could fire for any given input. If this happens, the output of all the rules are aggregated into one result. This aggregation combines all the shapes into a larger shape for the final result which can be seen in the bottom-right of Figure 5. To obtain a single crisp value for this range of values covered by the shape, the center of gravity is calculated. In this FIS, the centroid method was used, calculating the center of the area under the curve of the shape. Once the center of gravity of the aggregated shape is calculated, the range value at that point becomes the crisp value for the input. This crisp value is then plotted on the set of output membership functions to defuzzify it and obtain a final value which is used to determine a class for the input data. Afterwards, the next input can be processed.

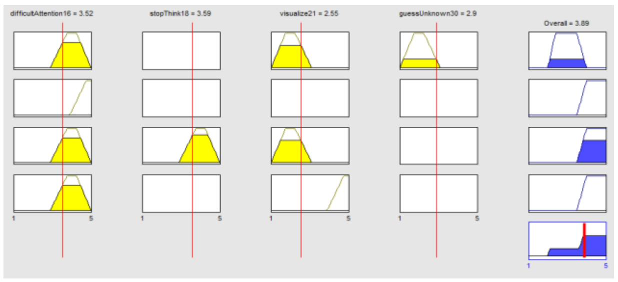

Fig. 5. Shapes from Rule Firing

III. Results and Discussions

Once the system has been implemented, the test data can be run through the FIS to validate the data. By using the command line in MATLAB, it is possible to test each value and the system returns the resulting class. By comparing the predicted classes to the actual classes, an accuracy rating can be determined. Overall accuracy was calculated by taking the number of correct classifications divided by the total number of samples.

A. MARSI Overall Dataset

This dataset consisted of all the MARSI data, having 30 questions as attributes and 1636 records. This dataset determined an aggregated level of reading strategies for the reader by considering all reading strategies to classify them into three categories: low, medium, or high.

1. If (contentFit7 is Never) and (stopThink18 is Never) and (analyse23 is Never) then (Overall is Low)
2. If (contentFit7 is Never) and (stopThink18 is Occasionally) and (analyse23 is Never) then (Overall is Low)
3. If (contentFit7 is Occasionally) and (stopThink18 is Occasionally) and (analyse23 is Never) then (Overall is Low)
4. If (analyse23 is Occasionally) then (Overall is Medium)
5. If (Know3 is Sometimes) and (stopThink18 is Sometimes) and (analyse23 is Sometimes) then (Overall is Medium)
6. If (Know3 is Sometimes) and (analyse23 is Sometimes) then (Overall is Medium)
7. If (analyse23 is Always) then (Overall is High)
8. If (summarize6 is Usually) and (stopThink18 is Usually) and (analyse23 is Usually) then (Overall is High)
9. If (summarize6 is Always) and (analyse23 is Usually) then (Overall is High)

Fig. 6. Extracted Rules for Overall Dataset

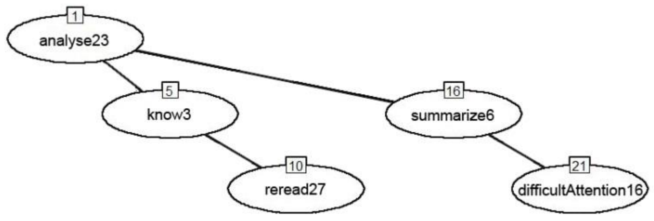

Fig. 7. A Subset of the MARSI Overall Tree

After running this dataset through the C4.5 induction tree, nine rules were selected, three from each class of high, medium, and low, that represented the strongest rules that applied to the largest sections of the data. To find the strongest rules, the rules were ranked by number of samples classified for each class, and the three highest rules that classified the most data were selected. These rules for the MARSI Overall dataset can be seen in Figure 6. These rules were derived from the tree produced from the C4.5 algorithm. The subset of the top layers of the tree is shown in Figure 7. From the tree, we can see that the attribute 'analyse23', corresponding to the 23rd question in the instrument, is the attribute that is the most influential on overall reading strategies. Using the C4.5 tree produced, the data can be run through the classifier again to determine accuracy. After running a confusion matrix on the MARSI overall dataset, it was found to correctly classify the data 83.33% of the time. Those rules were then taken and input into a fuzzy inference system. After processing all the data through the FIS, it was found that the FIS classified data correctly 72.55% of the time. A tool known as a confusion matrix can show induvial accuracies for each classification. The confusion matrix for the data after being run through the FIS is shown in Table 1.

TABLE I. FIS Confusion Matrix for MARSI Overall Dataset

	Low	Medium	High	Row Total
Low	30	35	1	66
Medium	23	425	287	735
High	0	103	732	835
Column Total	53	563	1020	1636

B. MARSI Global Reading Strategies Dataset

The global reading strategies dataset (GLOB) is a subset of the MARSI overall dataset consisting of only thirteen of the thirty questions. After processing this dataset similarly to the first and using the same instruments to analyze the data, this dataset showed an 84.62% accuracy when using the 9 rules found to reclassify the data. After taking those rules and using them in building the GLOB FIS, classification of the data resulted in a 70.29% accuracy.

C. MARSI Problem Solving Strategies Dataset

The problem solving strategies dataset (PROB) is a subset of the MARSI overall dataset consisting of eight of the thirty questions. Applying the same process to the data showed an accuracy of 62.5% after applying the C4.5 algorithm, and an 86.76% accuracy with the PROB FIS built with the extracted rules.

D. MARSI Support Reading Strategies Dataset

The support reading strategies dataset (SUP) is a subset of the MARSI overall dataset consisting of the remaining eight questions. As with the other datasets, the MARSI SUP dataset was processed through the C4.5 induction tree to determine rules and an accuracy of 100% was observed. After taking those rules and using them to build the SUP FIS, the accuracy of the system was shown to be 68.89%.

IV. Conclusions

In this research, the C4.5 induction tree algorithm was applied to four datasets in order to obtain rules that were easily understandable and helped to show patterns in conditions that led to a classification. These rules were then tested for accuracy using a fuzzy inference system that was built for each dataset. The method was applied to the MARSI datasets – overall, global, problem-solving, and support. Rules for the systems were selected manually from a list, choosing rules that had the greatest strength. It was shown that rule generation has varying efficiency depending on the dataset used, however, the rules generated still classified the majority of the data in all datasets with the lowest accuracy being 68.89%.

Looking at the MARSI datasets, it can be seen from Table 2 that there are far more classifications of 'High' than 'Medium', and even fewer 'Low' classifications. This relationship is most likely due to the effect of the students participating in the instrument inflating their own self-assessments. It has been found that people often overstate their abilities and see themselves as above average when they actually score low in areas where they rate themselves [22]. This effect can also alter the results if the student is unskilled in the area they are evaluating themselves on. This can possibly explain how the data for the MARSI datasets are slightly skewed, having few rules generated for the 'Low' classes. Rules selected for the 'Low' classes often contained far fewer samples than the other classes, but since they had the highest sample size in their class, they were selected and used in the system. Even with the inflation of self-assessment effect providing some inaccuracy in the data, the MARSI datasets were still able to produce accurate rules, with some datasets having higher accuracy than others.

TABLE II. TOTALS FOR CLASSIFICATION AMOUNTS

		Dataset			
		Overall	GLOB	PROB	SUP
Class	Low	66	107	25	240
	Medium	735	793	317	806
	High	835	736	1294	590

Our study has shown that induction trees provide an efficient tool for extracting reliable rules from datasets such as MARSI. However, in future research, we plan to apply other methods such as neural networks, support vector machines (SVM), and K-nearest neighbor algorithm for classification and rule extraction. The usage of such methods will enable us to compare methods with respect to efficiency of rule extraction, classification accuracy, and potentially attainment of a best possible rule set. In addition, it is possible to rank attributes by their information content and use a subset of those with the highest information content to improve efficiency and possibly accuracy. The current dataset used for purposes of this study was limited to only thirty attributes and did not have demographic attributes such as age, gender, ethnicity, and student reading ability. Demographic variables such as these, and others, could have been factored into the research to show how they contribute, individually or collectively, to students' awareness and use of metacognitive strategies when reading.

REFERENCES

[1] RAND Reading Study Group, "Reading for Understanding: Toward a Research and Development Program in Reading Comprehension." Arlington, VA: RAND, 2002.

[2] S. Mitra, S. K. Pal, and P. Mitra, "Data mining in Soft Computing Framework: A Survey," IEEE Trans. Neural Networks, vol. 13, no. 1, pp. 3-14, 2002.

[3] J. Malone, K. McGarry, S. Wermter, and C. Bowerman, "Data mining using rule extraction from Kohonen self-organizing maps," Neural Computer and Applications, vol. 15, no. 1, pp. 9-17, 2005.

[4] G. Fung, S. Sandilya, and R. B. Rao, "Rule Extraction from Linear Support Vector Machines," Eleventh ACM International Conference on Knowledge Discovery in Data Mining, pp. 32-40, 2005.

[5] M. Ali, S. Qaseem, L. Rajamani, and A. Govardhan, "Extracting Useful Rules through Improved Decision Tree Induction Using Information Entropy," International ournal of Informaition Sciences and Techniques, vol. 3, no. 1, pp. 27-41, 2013.

[6] Z. Zhou, Y. Jiang, and S. Chen, "Extracting Symbolic Rules from Trained Neural Network Ensembles," AI Communications - Artificial Intelligence Advances in China, vol. 16, no. 1, pp. 3-15, 2003.

[7] J. Anderson, K. Mokhtari, and A. Kulkarni, "Assessing Metacognitive Skills Using Adaptive Neural Networks," Procedia Computer Science, pp. 294-299, 2012.

[8] T. S. Roy, N. Sirohi, and A. Patle, "Classification of Lung Image and Nodule Detection Using Fuzzy Inference System," International Conference on Computing, Communication and Automation, pp. 1204-1207, 2015.

[9] S. Sapna and A. Tamilarasi, "Fuzzy Relational Equation in Preventing Diabetic Heart Attack," International Conference on Advances in Recent Technologies in Communication and Computing, pp. 635-637, 2009.

[10] I. Mala, P. Akhtar, T. J. Ali, and S. S. Zia, "Fuzzy Rule Based Classification for Heart Dataset using Fuzzy Decision Tree Algorithm based on Fuzzy RDBMS," World Applied Sciences Journal, vol. 28, no. 9, pp. 1331-1335, 2013.

[11] S. Sajasi and A. M. E. Moghadam, "A High Quality Image Steganography Scheme Based on Fuzzy Inference System," 13th Iranian Conference on Fuzzy Systems, pp. 1-6, 2013.

[12] A. Behrouznia, M. Saveri, A. Azadeh, S. M. Asadzadeh, P. Pazhoheshfar, "An Adaptive Network Based Fuzzy Inference System-Fuzzy Data Envelopment Analysis for Gas Consumption Forecasting and Analysis: The Case of South America," International Conference on Intelligent and Advanced Systems, pp. 1-6, 2010.

[13] K. Mokhtari and C. A. Reichard, "Assessing Students' Metacognitive Awareness of Reading Strategies," Journal of Educational Psychology, vol. 94, no. 2, pp. 249-259, 2002.

[14] K. Mokhtari, and R. Sheorey, "Reading Strategies of first and second language learners: See how they read," Norwood, MA: Christopher Gordon Publishers, 2008.

[15] J. R. Quinlan, "C4.5 Programs for Machine Learning," Morgan Kaufmann, 1992.

[16] K. Jearanaitanakij, "Classifying Continuous Data Set by ID3 Algorithm", Fifth International Conference on Information, Communications, and Signal Processing, pp. 1048-1051, 2005.

[17] A. D. Kulkarni, Computer Vision and Fuzzy Neural Systems. Upper Saddle River, NJ: Prentice Hall, 2001.

[18] J. R. Quinlan, "Induction of Decision Trees," Mach. Learn., vol. 1, no. 1, pp. 81–106, Mar. 1986.

[19] K. Hornik, C. Buchta, and A. Zeileis, "Open-Source Machine Learning: R Meets Weka", Computational Statistics, vol. 24, no. 2, pp. 225-232, 2009.

[20] J-SR. Jang, "ANFIS: adaptive-network-based fuzzy inference system," IEEE Transactions on Systems, Man, and Cybernetics, vol. 23, no. 3, pp. 665-685, 1993.

[21] A. D. Kulkarni and Sara McCaslin, "Knowledge Discovery From Multispectral Satellite Images," IEEE Geoscience and Remote Sensing Letters, vol. 1, no. 4, pp. 246-250, 2004.

[22] J. Kruger and D. Dunning, "Unskilled and Unaware of It: How Difficulties in Recognizing One's Own Incompetence Lead to Inflated Self-Assessments," Journal of Personality and Social Psychology, vol. 77, no. 6, pp. 1121-1134. 1999.