Assessing Student Proficiency in a Reading Tutor that Listens

Joseph E. Beck, Peng Jia, and Jack Mostow

joseph.beck@cmu.edu
http://www.cs.cmu.edu/~listen
Project LISTEN
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213. USA.

Abstract. This paper reports results on using data mining to extract useful variables from a database that contains interactions between the student and Project LISTEN's Reading Tutor. Our approach is to find variables we believe to be useful in the information logged by the tutor, and then to derive models that relate those variables to student's scores on external, paper-based tests of reading proficiency. Once the relationship between the recorded variables and the paper tests is discovered, it is possible to use information recorded by the tutor to assess the student's current level of proficiency. The major results of this work were the discovery of useful features available to the Reading Tutor that describe students, and a strong predictive model of external tests that correlates with actual test scores at 0.88.

1 Introduction and Motivation

Project LISTEN's Reading Tutor is an intelligent tutor that listens to students read aloud and helps them learn how to read. Target users are students in first through fourth grades (approximately 6- through 9-year olds). The Reading Tutor uses speech recognition technology to (try to) determine which words the student has read incorrectly and provide help.

Constructing a student model for the Reading Tutor is a challenging task. Most student models are structured according to the domain content or a model of how students solve procedural problems [1]. Previous work at constructing student models in computer tutors for language learning has focused on understanding students' typed input [5]. Although the Reading Tutor uses mouse for some input, requiring typing would not work well since non-readers cannot write.

Our goal with this work is to use fine-grained data generated by student-Reading Tutor interactions to provide assessment of students' reading performance that rely on empirical knowledge that we can derive from data. Previous work [3] has used external tests to validate the accuracy of a user model. This prior work used the student model's estimates of the student's proficiencies to predict what his score would be on an exam. Correlations between predicted and actual scores reached 0.81.

We are instead starting with student data and using external tests to *derive* a student model. If we can accurately predict how a student would perform on a paper test, we can use that prediction to direct the tutor's decision making. Such automated assessments can be used to help adapt the Reading Tutor's functionality by selecting stories for the student at an appropriate level of difficulty.

2 Approach

In the 2000-2001 school year, 88 students in grades one through four (i.e. 6- through 9-year olds) used the Reading Tutor from late October through early June. There were 37 first graders, 18 second graders, 17 third graders, and 16 fourth graders.

We tested students individually 4 times. In October and April, students were measured for fluency and the Woodcock Reading Mastery Test (WRMT) [7]. We also tested students' fluency in January and May. We measured fluency by having each student read 3 grade-level passages and counting the number read correctly, and then taking the median of those 3 numbers. The WRMT is a battery of tests designed to assess the student's reading proficiency across a broad spectrum.

In this paper, we use these test scores to relate student interaction data logged by the Reading Tutor to the paper tests for purposes of automatically assessing the student. Specifically, we predict student fluency and Word Identification (WI) from the WRMT. WI measures the student's skill at correctly reading words in English. A 2.4 means a student demonstrates word identifications skills at the level of 4 months into the second grade.

The Reading Tutor logs when a student reads a story, a sentence, and a particular word. It also logs when students request help on a word. From this information we can define a series of measures that describe how students are performing while using the Reading Tutor. One measure is the interword latency [6], defined as the time from when a student finishes speaking the *i-1*th word in the sentence until he begins to **correctly** pronounce the *i*th word. Note that some words do not have defined latencies. If a student never reads word *i-1* in the sentence, then word *i* does not have a latency.

We defined several features based on latency:

1. Total percentage of words having a defined latency
2. Percentage of words read fluently (latency of 10ms)
3. Percentage of words read disfluently (latency >5000ms)
4. Median of all latencies
5. Mean of all latencies

We also defined features based on the student's help request behavior

1. Percentage of sentences for which the student requested sentence help
2. Percentage of words about which the student requested help

The features about latency and help requests were defined for all words the student encountered. We also computed those features just for words that are on the Dolch list [4] of 220 frequent words. We then computed those features just for words that are not on the Dolch list. We also used as features each student's grade and gender, and the percentage of words the student read the Reading Tutor accepted as correct.

Since the Reading Tutor's logs were designed primarily for debugging rather than educational data mining, certain types of interactions were not logged in a parseable form. See [2] for a description of problems with the logging procedure. In spite of relatively minor warts with the logging (which have been fixed in later versions of the Reading Tutor), we were able to define and extract many potentially useful descriptors of student performance from our database. We now turn to using these data to predict scores on the WRMT and fluency tests.

We only consider data about student performance in the Reading Tutor from within a window of time before a paper-test is administered. We experimented with a variety of window sizes: 1 week, 2 weeks, 4 weeks, 8 weeks, 12 weeks, and all data before the test was administered to explore tradeoffs between timeliness of data and noisiness of estimate. Data that are more recent better describe a student's changing state of knowledge. However, if the window is too small, then we our estimates of the parameters may be noisy (i.e. a version of bias-variance tradeoff). Once we have a specified window size, we collect data on all of the student measures from within that window and use those data to construct a set of features representing the student's performance within the Reading Tutor.

Each training instance consists of one paper test score and 62 features computed from student performance during its associated time window.

We aggregated the data for all 4 fluency tests and for both WI tests together. Since there are 88 students and 4 fluency tests, combining the data together provides 352 instances to train a model of fluency. With only 2 WI tests, there are 176 instances for training a model of the WI component of the WRMT. Due to students missing some tests and our losing low-level Reading Tutor data from one student, we only had 344 fluency and 173 WI test scores to serve as labels.

We conducted experiments using Weka, a public domain set of datamining tools written in Java, and used its model tree and linear regression algorithms to make predictions. Model trees are a combination of decision trees and linear regression: first the training data are partitioned as in a decision tree, but the leaf nodes contain a linear model for making predictions. Model trees handle non-linearities in data by splitting the data into regions that can be better modeled with linear techniques. For both techniques we used the default settings in Weka.

3 Results

We now relate our model's predictions to how students actually performed on the paper-based tests. These results are from a 10-fold cross validation of 87 students. We used window sizes of 1 week, 2 weeks, 4 weeks, 8 weeks, 12 weeks, and all data before the test. The best result for predicting fluency was a correlation of 0.86 from using a model tree with a 12 week window. Model trees did better than linear regression for all window sizes except for 1 week.

For Word Identification, the pattern between window size and performance was less clear. Model trees outperform Weka's linear regression, and performance improves somewhat as window size increases. Using an 8-week window, the model

tree's predictions correlate at 0.87 with the actual test scores while Weka's linear regression correlates at 0.82.

A truism in datamining and machine learning is to start with simple models first, and then see if using more complex models is needed and justified. However, we have found that which software you use for your simple models can make a noticeable difference. We performed the majority of our work in Weka since it is free and the source code is available. Having source code is a great advantage for researchers since it simplifies conducting experiments that software designers may not have thought of. However, we have found that Weka and SPSS disagree on how well linear models fit our data. Figure 1 shows how SPSS and Weka compare in model accuracy. SPSS gives a maximum correlation of 0.88 with the actual test scores, compared to 0.87 with Weka's model trees. This difference in correlations is negligible; however, models generated by SPSS are relatively insensitive to window size, which is a good feature. This difference in performance between regression techniques is linked to Weka's default behavior that first prunes variables before building its linear models. For regression it is easy to disable this pruning. For model trees, turning off the pruning requires a source code modification that we have not yet completed.

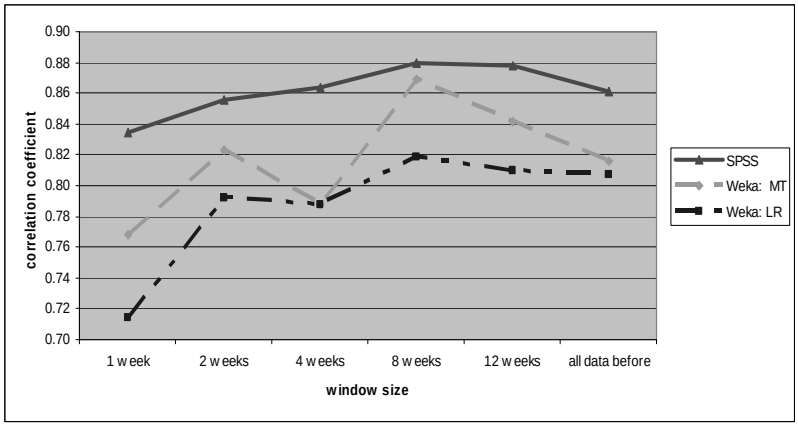

Figure 1. Performance at predicting word identification test

4 Conclusions

The approach of constructing a user model by relating fine-grained features to existing, external measures is a promising one. It makes sense to bootstrap from the extensive effort that has been spent psychometrically validating instruments such as the WRMT. For domains such as reading, where it can be difficult to determine which student performance measures are important, and it is impractical to make the user interface transparent to show the student's "problem-solving steps," constructing a student model in this manner is a sensible procedure. However, this approach

would not be practical for modeling finer-grained aspects of student-knowledge (e.g. whether a student knows *ph* makes the sound /f/ in the word "phone").

We have determined which variables are important in assessing student knowledge at a coarse level. Viewing the variables as broad categories, both latency and help request features provided useful information for predicting a student's level of reading ability. Both variables combined did better than either one alone.

The most useful single variable was the percentage of words that had a latency defined. This feature is somewhat different from, and outperformed, the percentage of words the speech recognizer heard the student say correctly. Since latency is only defined for 2 successive words read correctly [6], it is not defined for the first word of the sentence or for isolated words read correctly. Thus, the student's ability to string multiple words in a row together seems to have some predictive power above and beyond just saying those words correctly in isolation.

Acknowledgements

This work was supported in part by the National Science Foundation under Grant No. REC-9979894. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government. We thank members of Project LISTEN who contributed to this work, especially Susan Rossbach for conducting the field studies, Andrew Cuneo for constructing the database from the logfiles, and June Sison for commenting on a draft of this paper.

References (see www.cs.cmu.edu/~listen for LISTEN publications)

1. Anderson, J.R., *Rules of the Mind*. 1993: Lawrence Erlbaum Assoc.
2. Beck, J.E., Jia, P., Sison, J. and Mostow, J., Predicting student help-request behavior in an intelligent tutor for reading. Proceedings of the *Ninth International Conference on User Modeling*. 2003
3. Corbett, A.T. and Bhatnagar, A., Student Modeling in the ACT Programming Tutor: Adjusting a Procedural Learning Model With Declarative Knowledge. Proceedings of the *Sixth International Conference on User Modeling*. 1997
4. Dolch, E., A basic sight vocabulary. *Elementary School Journal*, 1936. **36**: p. 456-460.
5. Michaud, L.N., McCoy, K.F. and Stark, L.A., Modeling the Acquisition of English: an Intelligent CALL Approach". Proceedings of the *Eighth International Conference on User Modeling*. 2001
6. Mostow, J. and Aist, G., The Sounds of Silence: Towards Automated Evaluation of Student Learning in a Reading Tutor that Listens. Proceedings of the *Proceedings of the Fourteenth National Conference on Artificial Intelligence*. p. 355-361. 1997
7. Woodcock, R.W., *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU)*. 1998, Circle Pines, Minnesota: American Guidance Service.