Using Speech Recognition to Evaluate Two Student Models for a Reading Tutor

Kai-min Chang, Joseph Beck, Jack Mostow, Albert Corbett
Project LISTEN
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract. Intelligent Tutoring Systems derive much of their power from having a student model that describes the learner's competencies. However, constructing a student model is challenging for computer tutors that use automated speech recognition (ASR) as input, due to inherent inaccuracies in ASR. We describe two extremely simplified models of developing word decoding skills and explore whether there is sufficient information in ASR output to determine which model fits student performance better, and under what circumstances one model is preferable to another.

The two models that we describe are a lexical model that assumes students learn words as whole-unit chunks, and a grapheme-to-phoneme (G→P) model that assumes students learn the individual letter-to-sound mappings that compose the words. We use the data collected by the ASR to show that the G→P model better describes student performance than the lexical model. We then determine which model performs better under what conditions. On one hand, the G→P model better correlates with student performance data when the student is older or when the word is more difficult to read or spell. On the other hand, the lexical model better correlates with student performance data when the student has seen the word more times.

Keywords. Intelligent Tutoring Systems, Student Model, Automatic Speech Recognizer, Knowledge Representation

1. Introduction

Intelligent Tutoring Systems (ITS) derive much of their power from having a student model [16] that describes the learner's proficiencies at various aspects of the domain to be learned. For example, the student model can be used to determine what feedback to give [3] or to have the students practice a particular skill until it is mastered [4]. Unfortunately, language tutors that use automated speech recognition (ASR) as input have difficulty in developing strong models of the student. Much of the difficulty comes from the inaccuracies inherent in the ASR output. Providing explicit feedback based only on student performance on one attempt at reading a word is not viable since the accuracy at distinguishing correct from incorrect reading is not high enough [14].

In previous work, we have been able to use ASR output to estimate a student's overall level of knowledge [1] (e.g. help requests within the use of a reading tutor [2]) and assess interventions (e.g. help selection policy) of a tutoring system [7]. The next question is whether we can construct a student model from the ASR output. Specifically, we would like to model internal knowledge representation of reading and word decoding strategies. Ideally, we would like to construct a complex student model capturing all aspects of reading. For example, Ehri [6] describe the reading process:

> "Reading words may take several forms. Readers may utilize decoding, analogizing, or predicting to read unfamiliar words. Readers read familiar words by accessing them in memory, called sight word reading. With practice, all words come to be read automatically by sight, which is the most efficient, unobtrusive way to read words in text. The process of learning sight words involves forming connections between graphemes and phonemes to bond spellings of the words to their pronunciations and meanings in memory. The process is enabled by phonemic awareness and by knowledge of the alphabetic system, which functions as a powerful mnemonic to secure spellings in memory."

However, training such a complex student model is clearly infeasible due to a sparse data problem. Although we can obtain more data with ASR, the inherent inaccuracies with ASR output must be addressed. Therefore, in the current study we first propose two extremely simplified models of developing word decoding skills and examine whether there is sufficient information *at all* in ASR output to discriminate the two overly simplified models.

More specifically, the two models that we describe are a lexical model that assumes students learn words as whole-unit chunks, and a grapheme-to-phoneme model that assumes students learn the individual letter-to-sound mappings that compose the words. Given the observed student performance data, we map those overt actions to some internal representation of the student's knowledge. Then, we evaluate the two models to determine which model fits student performance data better. Furthermore, we examine under what circumstances one model is preferable to another.

2. Knowledge Tracing

The goal of knowledge tracing is to estimate student's knowledge from their observed actions. Prior work in this area [2] has shown that knowledge tracing [4] is an effective approach for using ASR output to model students.

As illustrated in Figure 1, knowledge tracing maintains four constant parameters for each skill. Two parameters, $L0$ and t, are called learning parameters and refer to the student's initial knowledge and to the probability of learning a skill given an opportunity to apply it, respectively. Two other parameters, *slip* and *guess*, are called performance parameters and account for student performance not being a perfect reflection of his underlying knowledge. The guess parameter is the probability that a student who has not mastered the skill can generate a correct response. The slip parameter is used to account for even knowledgeable students making an occasional mistake.

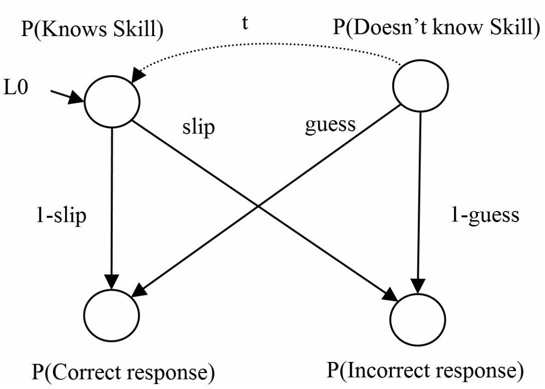

Figure 1. Overview of knowledge tracing. A set of L0, t, slip and guess parameters is estimated for each skill, while the internal knowledge state of a skill is traced for each student.

At each successive opportunity to apply a skill, knowledge tracing updates its estimates of a student's internal knowledge state of the particular skill, based on the skill-specific learning parameters and the observed student performance (evidence). $P(L_n)$ denotes the probability of knowing the skill following the n^{th} encounter,

$$P(L_n) = \begin{cases} L0 & \text{if } n = 0 \\ P(L_{n-1}|evidence) + (1 - P(L_{n-1}|evidence)) * t & \text{if } n > 0 \end{cases} \quad (1)$$

Given the current knowledge state of a student at a particular skill, knowledge tracing then predicts the probability of the student performing the skill correctly, based on the skill-specific performance parameters. $P(O_n)$ denotes the probability of applying the skill correctly at n^{th} encounter,

$$P(O_n) = P(L_{n-1}) * (1 - slip) + (1 - P(L_{n-1})) * guess \quad (2)$$

Prior work on applying knowledge tracing to ASR output [2] demonstrate that the slip and guess parameters, in addition to accounting for variability in student performance, also account for variability in the ASR scoring of student responses.

3. The Lexical and Grapheme-to-phoneme Student Model

We consider two extremely simplified models for how students can learn to decode words. The first is a lexical model, which assumes that students learn words as a whole-unit with no transfer between words. Although the assumed lack of transfer is somewhat naive, it is likely that skilled readers recognize most words by sight [6]. It is less clear, however, whether children learning to read have a similar representation as skilled readers.

The second model is a grapheme-to-phoneme (G→P) model, and assumes that rather than learning whole words, students instead learn sub-lexical units. Specifically, it assumes that students learn the grapheme (letter) to phoneme (sound) mappings that make up words. For example, the word "cat" contains the following G→P mappings: c→/K/, a→/AE/, and t→/T/.

Unlike the lexical model which assumes lack of transfer between words, the G→P model allows students to share the sub-lexical knowledge for words that share G→P mappings. For example, the word "bat" contains the G→P mappings of b→/B/, a→/AE/, and t→/T/, where the last two G→P mappings are shared with the word "cat". The G→P model assumes that knowledge about a→/AE/, and t→/T/ that are learned from reading the word "cat" will transfer to the word "bat".

4. Data Collection

Our data came from 360 students who used the Reading Tutor [9] in the 2002-2003 school year. The students using the Reading Tutor were part of a controlled study of learning gains, so were pre- and post-tested on the Woodcock Reading Mastery Test [15]. The test was human administered and scored.

Over the course of the school year, these students read approximately 1.95 million words (as heard by the ASR). On average, students used the tutor for 8.5 hours. Most students were between six and eight years old, and had reading skills appropriate for their age.

During a session with the Reading Tutor, the tutor presented one sentence (or fragment) at a time, and asked the student to read it aloud. The student's speech was segmented into utterances that ended when the student stopped speaking. Each utterance was processed by the ASR and aligned against the sentence. This alignment scored each word of the sentence as either being accepted (heard by the ASR as read correctly), rejected (the ASR heard and aligned against some other word), or skipped (not read by the student) [11]. For example, in Table 1, the student was supposed to read "The dog ran behind the house." The bottom row of the table showed how the student's performance would be scored by the tutor.

Table 1. Example alignment of ASR output to sentence

Sentence	The	dog	ran	behind	the	house.
ASR output	The	the	ran			
Scoring	Accept	Reject	Accept	Skipped	Skipped	Skipped

Notice that, we used the terms "accepted" and "rejected" rather than "correct" and "incorrect" due to inaccuracies in the ASR. The ASR only noticed about 25% of student misreadings, and scored as read incorrectly about 4% of words that were read correctly. Therefore, "accept" and "reject" were more accurate terms.

5. Experiment 1: Fitting Aggregate Student Performance Data

5.1. Model Representation and Credit Assignment

To determine which of the lexical and G→P models better describes student performance, we fit each model to student performance data as heard by the ASR. First, we split the students into two groups (to create a testing set to be used later). Then, for each model, we estimate the knowledge tracing parameters for each skill using an optimization algorithm[1]. The optimization algorithm performs a gradient search over the space of L0, t, guess and slip to find the best fit of a non-linear curve to all student performance data in the training set, characterized by Equation 1 and 2.

For the lexical model, we simply treat words as skills. Therefore, each student's attempt at reading a word is evidence for knowing the whole word or not. For the G→P model, modeling the student's proficiency at a sub-lexical level is difficult, as we do not have observations of the student attempting to read G→P mappings in isolation. In the current study, we adopt a simple crediting mechanism: if a word is accepted by the ASR, then all of the G→P mappings are credited; otherwise, if a word is rejected, then all of the mappings are debited.

The lexical model has considerably more skills than the G→P model. There are 3210 lexical skills (i.e. words) and in comparison, there are only 295 G→P mappings encountered by students. As a result of this difference in number of skills, the G→P model has substantially more students encountering each skill on average (106 vs. 45).

5.2. Model Fit

Table 2 describes the knowledge tracing parameter estimates for each of the models. Notice that, the knowledge tracing parameters are skill-specific; that is, a set of L0, t, guess and slip is estimated for each skill. To summarize the parameters for a model, we report the *average* across each skill in the model, weighted by the number of times the skill occurred. This weighting is to avoid biasing the model by several skills that occur rarely (e.g. the word "arose" or "bts→/TS/" as in the word "debts").

Table 2. Estimated knowledge tracing parameters (averaged across skills, weighted by the number of times the skill occurred)

Model	L0	T	Guess	Slip	R^2
Lexical	0.32	0.14	0.65	0.08	0.34
G→P	0.49	0.01	0.57	0.10	0.48

As seen in Table 2, the performance parameters (guess and slip) are similar for both models, while the learning parameters (L0 and T) are different. These performance parameters are vastly different than in knowledge tracing done in other ITSs (where typically "guess" is restricted to be less than 0.3 [4]). The reason

[1] Source code is courtesy of Albert Corbett and Ryan Baker and is available at http://www.cs.cmu.edu/∼rsbaker/curvefit.tar.gz

for this difference is the uncertainty introduced by the ASR. This uncertainty is also the reason the performance parameters under both models are similar: the parameters are (mostly) modeling the speech recognition rather than the student. The column labeled R^2 in the table refers to how well the knowledge tracing parameters fit student performance data. The R^2 for the lexical and G→P models are 0.34 and 0.48, respectively, and are significantly different at $p < 0.01$.

At least within the framework of knowledge tracing, student performance is better described by the G→P model than by the lexical model. Thus, the G→P model appears to be a better description of how children between six and eight acquire reading skills.

Notice that, the knowledge tracing's model fit, R^2, fits the *aggregated* student performance data. That is, the performance data of all students are lumped together in order to have more data to estimate knowledge tracing parameters more reliably. Consequently, the estimated knowledge tracing parameters describe aggregated student performance data and are not student-specific.

6. Experiment 2: Fitting Individual Performance Data

Given that the G→P model fit the *aggregate* student performance better, our second goal is to determine which of the lexical and G→P model fit the *individual* student performance data better. Our approach is to treat the problem as a classification problem. For each student, we use knowledge tracing's estimates of his proficiency to predict whether the ASR will accept a word that he attempts to read.

For example, upon encountering the word "cat", we extract a student's proficiency in both the lexical and G→P model. Whereas the lexical model asserts that successful reading of the word "cat" depends on proficiency in only one skill, "cat", the G→P model asserts that it depends on three sub-lexical skills, c→/K/, a→/AE/, and t→/T/. Notice that, the skill proficiency can be estimated in two ways. We may estimate it to be the probability of *knowing* the skill, or the probability of correctly *applying* the skill. Unfortunately, neither of $P(O_n)$ nor $P(L_n)$ is perfect solution. On one hand, by using $P(O_n)$, we run the risk of solely modeling the ASR, even when $P(L_n)$ contains no information (that it is not modeling student knowledge). On the other hand, by using $P(L_n)$, we run the risk of ignoring ASR's tendencies to accept/reject certain words regardless of student's knowledge. One remedy is to evaluate and bound proficiency in both $P(O_n)$ and $P(L_n)$. In the current study, we simply use the $P(O_n)$.

Given students' proficiencies in both the lexical and G→P skills of a word, we train two logistic regression classifiers to predict whether the word will be accepted by the ASR. The first logistic regression classifier is for the lexical model and has one predictor - the corresponding lexical skill for the word. The second logistic regression classifier is for the G→P model and has one predictor for each sub-lexical skill in the word. In the above example, the word "cat" requires only one skill in the lexical model, but three skills in the G→P model. To account for such differences, we train different logistic regression models for different word lengths. That is, for the lexical model, we train a logistic regression for all words with the same word length, totaling 16 models since the longest word tried has a length of 16 characters. For the G→P model, a logistic regression model is trained for all words with the same number of G→P mappings, totaling 12 models since the longest word tried has 12 G→P mappings.

We use the second (testing) half of our data to construct the classifiers, so these data have not been used to perform the knowledge tracing parameter estimates of L0, t, slip, or guess. We then compute the R^2 for each length, and weight the overall result by the number of words of each length. The weighted R^2 suggests whether data can be predicted by our models. As seen in Table 3, the R^2 for the lexical model is essentially the same as the G→P model (0.0861 and 0.0832, respectively). Notice that, the R^2 for individual data are expected to be smaller than R^2 for aggregate data (0.34 and 0.48) since aggregated data are smoother.

Given the two logistic regression models, each model makes separate predictions on the probability that a student will read a word correctly. We then use the probabilistic predictions of the two models as independent variables in a logistic regression model to again predict individual performance data. The combined model achieves an even higher R^2 of 0.109, as seen in Table 3. This finding suggests that, although each model fits individual performance data equally well, there exists some variations in model predictions and each model accounts for unique variance in student performance. It is likely that students use different strategies for different words. That is, students may use the lexical model for some words and the G→P for other words. In our next experiment, we examine which model is preferable under what circumstances.

Table 3. Logistic regression

Model	R^2
Lexical	0.0861
G→P	0.0832
Combined	0.1090

7. Experiment 3: Which model performs better under what conditions

7.1. Model Preferability and Contextual Information

Given that the combined model is better, we want to know under what circumstances one model outperforms another. We do this by correlating various student and word information with Delta, a construct that relates to preferability of a model.

For each word encounter, each model makes separate predictions of the probability that a student will read the word correctly. We can compute the error made by each model by taking the squared difference between a model's probabilistic prediction and the student's observed performance. Then, we define Delta as the lexical model's error minus the G→P model's error. For example, suppose the lexical and G→P model estimate the probability that the student reads a

Table 4. Example of error in model prediction and Delta

Example	Model	Model Prediction	ASR accept	Error	Squared Error	Delta
1	Lexical	0.7	1	0.3	0.09	-0.16
	G→P	0.5	1	0.5	0.25	
2	Lexical	0.7	0	0.7	0.49	0.22
	G→P	0.5	0	0.5	0.25	

word correctly at a particular trial as 0.7 and 0.5, respectively, where in reality, the ASR indeed accepts student's reading. Then, the squared error of the two models are $(1-0.7)^2 = 0.09$ and $(1-0.5)^2 = 0.25$, respectively, and Delta equals $0.09 - 0.25 = -0.16$. Therefore, a negative Delta indicates that the lexical model is performing better than the G→P model. Conversely, a positive Delta indicates that the G→P model is performing better than the G→P model (see Table 4).

As discussed earlier, we want to characterize the students and words for which one model outperforms the other. For information about a student, we include the student's age, grade, and word identification grade as found in the pretest of Woodcock Reading Mastery's Word Identification subtest [15]. For information about a word, we heuristically estimate the word's identification and spelling difficulty from the same Woodcock pretest. The measures give the difficulty estimate of the word in grade equivalent terms. In addition, we include *prior*, the number of prior encounters of the word within the Reading Tutor, and *frequency*, how often the word occurs in a corpus of English text. Finally, we identify the *dolch* [5] and *stop* words. The dolch words are a list of 220 high frequency words that are used in beginning reading programs, whereas the stop words are 36 high frequency words on which errors seldom affect comprehension [10].

7.2. The Correlation Matrix

The correlation between each feature and Delta is shown in Table 5. Despite the small correlation coefficients, all correlations, except grade, are in the expected direction and are statistically significant at $p < 0.01$. We now describe the observed correlations.

One one hand, the G→P model better estimates the student performance data when the student is older or has higher word identification proficiency (correlation of 0.014, and 0.008, respectively). This finding agrees with Ehri's description [6]: the process of skilled reading is enabled by phonemic awareness and by knowledge of the alphabetic system. Moreover, the G→P model also performs better when the word is more difficult. This is seen in the positive correlation of word identification difficulty and spelling difficulty with Delta (0.022 and 0.023, respectively). The direction is intuitive; the more difficult a word is, the more likely is one to decode the word using G→P mappings.

On the other hand, the lexical model better predicts student performance data when the word is more frequently encountered. This is seen in the negative correlation between number of prior encounters, frequency in English text and Delta (-0.014 and -0.016, respectively). The direction is intuitive; the more encountering of a word, the more likely one is to become a skilled reader with that word. Further, we have expected and found similar correlations for the dolch and stop word (correlations of -0.026 and -0.024, respectively).

Table 5. Correlation matrix. **Correlation is significant at $p < 0.01$ (2-tailed).

	Feature	Correlation with Delta (Positive means better fit for the G→P model)
Student	Age	0.014**
	Grade	0.000
	Word identification proficiency	0.008**
Word	Word identification difficulty	0.022**
	Spelling difficulty	0.023**
	Number of prior encounters	-0.014**
	Percent in English text	-0.016**
	Dolch word	-0.026**
	Stop word	-0.024**

8. Conclusion and Future Work

The ASR of a computer tutor for reading provides information about an individual student's reading development. This paper reports using ASR output from a computer tutor for reading to construct two models of how students learn to read words: a lexical model and a grapheme-to-phoneme (G→P) model. First, the two student models are evaluated to determine which model better predicts student performance data. The G→P model outperforms the lexical model in a model where we aggregate across student performance data. The performance difference disappears when we evaluate the models against individual performance data. Nonetheless, when we combine the two student models, the combined model outperforms either model alone. Consequently, we evaluate which model performs better under what conditions. Correlations between model fit and student information (grade, age, etc.), word information (number of prior encounters within tutor, frequency, etc.) are in the expected directions. On one hand, the G→P model better correlates with student performance data when a student is older or when the word is more difficult to read or spell. On the other hand, the lexical model better correlates with student performance data when the student has seen the word more times. There appears to exist sufficient information in the ASR output to determine which model is better under what circumstances.

Despite the initial success, the method for constructing a student model from the ASR output is somewhat crude. Two areas of potential improvement are a better credit model and using cues other than acceptance/rejection of a word. Currently, all of the G→P mappings in a word are blamed or credited. However, if a student misreads a word it is probable that not all of the mappings are responsible. A Bayesian credit assignment approach (e.g. [3]) would overcome this weakness. Similarly, the student's pattern of hesitation before a word contains a useful signal for modeling the student [9]. One possible avenue is to use the amount of hesitation before reading a word as a clue to the strategy the student is using: a short pause suggests a lexical strategy while a longer pause suggests the student is using knowledge of G→P mappings.

References

[1] Beck, J.E., P. Jia, and J. Mostow, *Automatically assessing oral reading fluency in a computer tutor that listens.* Technology, Instruction, Cognition and Learning, 2004. **2**: p. 61-81.

[2] Beck, J.E. and J. Sison. *Using knowledge tracing to measure student reading proficiencies.* in *Proceedings of International Conference on Intelligent Tutoring Systems.* 2004. p. 624-634.

[3] Conati, C., A. Gertner, and K. VanLehn, *Using Bayesian Networks to Manage Uncertainty in Student Modeling.* User Modeling and User-Adapted Interaction, 2002. **12**(4): p. 371-417.

[4] Corbett, A.T. and J.R. Anderson, *Knowledge tracing: Modeling the acquisition of procedural knowledge.* User Modeling and User-Adapted Interaction, 1995. **4**: p. 253-278.

[5] Dolch, E., *A basic sight vocabulary.* Elementary School Journal, 1936. **36**: p. 456-460.

[6] Ehri, L.C. *Learning to Read Words: Theory, Findings, and Issues.* Scientific Studies of Reading, 2005. **9**(2): p. 167-188.

[7] Heiner, C., J.E. Beck, and J. Mostow. *Improving the Help Selection Policy in a Reading Tutor that Listens.* in *Proceedings of International Conference on Computer Assisted Language Learning.* 2004. p. 195-198

[8] Larsen, S.C., D.D. Hammill, and L.C. Moats, *Test of Written Spelling.* fourth ed. 1999, Austin, Texas: Pro-Ed.

[9] Mostow, J. and G. Aist, *Evaluating tutors that listen: An overview of Project LISTEN*, in *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001, MIT/AAAI Press: Menlo Park, CA. p. 169-234.

[10] Mostow, J., Roth, S., Hauptmann, A. G., and Kane, M., *A Prototype Reading Coach that Listens*, in *Proceedings of the Twelfth National Conference on Artificial Intelligence* (AAAI-94), American Association for Artificial Intelligence, Seattle, WA, August 1994, pp. 785-792.

[11] Tam, Y.-C., Beck, J., Mostow, J., and Banerjee, S. *Training a Confidence Measure for a Reading Tutor that Listens*, in *Proc. 8th European Conference on Speech Communication and Technology* (Eurospeech 2003). 2003.p. 3161-3164 Geneva, Switzerland.

[12] Torgesen, J.K., R.K. Wagner, and C.A. Rashotte, *TOWRE: Test of Word Reading Efficiency.* 1999, Austin: Pro-Ed.

[13] Wiederholt, J.L. and B.R. Bryant, *Gray Oral Reading Tests.* 3rd ed. 1992, Austin, TX: Pro-Ed.

[14] Williams, S.M., D. Nix, and P. Fairweather. *Using Speech Recognition Technology to Enhance Literacy Instruction for Emerging Readers.* in *Fourth International Conference of the Learning Sciences.* 2000: Erlbaum.

[15] Woodcock, R.W., *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU).* 1998, Circle Pines, Minnesota: American Guidance Service.

[16] Woolf, B.P., AI in Education, in *Encyclopedia of Artificial Intelligence.* 1992, John Wiley & Sons: New York. p. 434-444.