AIED 05 WORKSHOP 12

Amsterdam, 18-22 July, 2005

Student Modeling for Language Tutors

WORKSHOP

12th International Conference on Artificial Intelligence in Education, Amsterdam, the Netherlands

Introduction

Student modeling is of great importance in intelligent tutoring and intelligent educational assessment applications. However, student modeling for computer-assisted language learning (CALL) applications differs from classic student modeling in several key ways, including the lack of observable intermediate steps (behavioral or cognitive) involved in successful performance. This workshop will focus on student modeling for intelligent CALL applications, addressing such domains as reading decoding and reading and spoken language comprehension. Domains of interest include both primary (L1) and second language (L2) learning. Hence, the workshop will address questions related to student modeling for CALL, including what types of knowledge ought such a model contain, with what design rationale, and how might information about the user's knowledge be obtained and/or inferred in a CALL context?

The first workshop on Student Modeling for Language Tutors is taking place at the AIED2005 conference. Constructing student models for language tutors is more challenging than for classical computer tutors for several reasons:

1. It is difficult to determine the reasons for successes and errors in student responses. In classic ITS domains (e.g., math and physics), the interaction with the tutor may require students to demonstrate intermediate steps. For performance in language domains, much more learner behavior and knowledge is hidden, and having learners demonstrate intermediate steps is difficult or perhaps impossible, and at any rate may not be natural behavior. (How) Can a language tutor reason about the cause of a student mistake? (How) Can a language tutor make attributions regarding a student's knowledge state based on overt behavior?
2. Cognitive modeling is harder in language tutors. A standard approach for building a cognitive task model is to use think-aloud protocols. Asking novices to verbalize their problem solving processes while trying to read and comprehend text is not a fruitful endeavor. How then can we construct problem solving models? Can existing psychological models of reading be adapted and used by computer tutors?
3. It may be difficult to accurately score student responses. For example, in tutors that use automated speech recognition (ASR), whether the student's response is correct cannot be determined with certainty. In contrast, in classic tutoring systems scoring the student's response is relatively easy. How can inaccuracies in scoring be overcome to reason about the students' proficiencies?

We should like to thank the following people for organizing the workshop and for reviewing the submissions:

Peter Fairweather. IBM T.J. Watson Research Center
Lewis Johnson. USC / Information Sciences Institute
Stephen A. LaRocca. Army Research Laboratory (ARTI)
Lisa N. Michaud. Wheaton College
Jack Mostow. Carnegie Mellon University

Sherman Alpert (co-chair). IBM T.J. Watson Research Center
Joseph E. Beck (co-chair). Carnegie Mellon University

Table of Contents

Modeling Student Knowledge in an Oral Reading Companion

Sherman R. Alpert, Peter G. Fairweather, Bill Adams, Jennifer Lai

IBM T.J. Watson Research Center
Yorktown Heights, NY USA
{salpert, pfairwea, whadams, jlai}@us.ibm.com

Abstract
Guided oral reading has been shown to have positive pedagogical value; our Reading Companion provides a shared reading experience in which students read on-screen books aloud guided by the Companion, which offers scaffolded modeling of expert skill and feedback based on speech recognition. A student model is maintained for each student, which tracks student performance and decoding knowledge in terms of rules and word features. Decoding rules (or, more accurately, heuristics) involve mapping sequences and patterns of letters and letter categories to particular sounds. An example letter sequence heuristic makes "ph" sound like /f/; an example letter category pattern informs us that the pattern VCe at the end of a syllable usually makes the V have its long sound and makes the final "e" silent. An example word feature might be the fact that a word contains a specific consonant blend. We describe how the tutor maps spoken words to rules and word features (collectively referred to as linguistic facts), and how information about these data are maintained in the student model. The Reading Companion includes a sophisticated post hoc reporting facility, which provides a view into the student model, allowing teachers to gain insights into students' strengths and weaknesses, and facilitating targeted individualized interventions. The Reading Companion is Web-based and accessible via an ordinary browser.

Using a student model to improve a computer tutor's speech recognition

Joseph E. Beck[1], Kai-min Chang[2], Jack Mostow[3], and Albert Corbett[4]
[1]Center for Automated Learning and Discovery
[2]Language Technologies Institute
[3]Robotics Institute
[4]Human Computer Interaction Institute
School of Computer Science
Carnegie Mellon University
joseph.beck@cmu.edu

Abstract. Intelligent computer tutors can 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. This paper reports using ASR output from a computer tutor for reading to compare two models of how students learn to read words: a model that assumes students learn words as whole-unit chunks, and a model that assumes students learn the individual letter→sound mappings that make up words. We use the data collected by the ASR to show that a model of letter→sound mappings better describes student performance. We then compare using the student model and the ASR, both alone and in combination, to predict which words the student will read correctly, as scored by a human transcriber. Surprisingly, majority class has a higher classification accuracy than the ASR. However, we demonstrate that the ASR output still has useful information, and that classification accuracy is not a good metric for this task, and the Area Under Curve (AUC) of ROC curves is a superior scoring method. The AUC of the student model is statistically reliably better (0.670 vs. 0.550) than that of the ASR, which in turn is reliably better than majority class. These results show that ASR can be used to compare theories of how students learn to read words, and modeling individual learner's proficiencies may enable improved speech recognition.

1 Motivation and Introduction

Intelligent Tutoring Systems (ITS) derive much of their power from having a student model [1] 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 [2] or to have the students practice a particular skill until it is mastered [3] Unfortunately, language tutors have difficulty in developing strong models of the student. Much of the difficulty comes from the inaccuracies inherent in automated speech recognition (ASR). 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 [4]. Due to such problems, student modeling has not received as much attention in computer assisted language learning systems as in classic ITS [5], although there are exceptions such as [6].

A common approach to developing cognitive models for use in an ITS is to use think-aloud protocols [7, 8]. In a think-aloud study [7], participants verbalize their thinking while solving a problem. Such verbalizations are then used to construct a cognitive model of how the participants were solving the task. This approach has also been used to develop cognitive models for ITS [8]. Unfortunately, due to the speed of the reading process, think-aloud methodology is not well suited to modeling reading.

There have been efforts to develop cognitive models that describe the reading process. For example, [9] developed a parallel distributed processing model that was able to simulate many aspects of human performance. A major drawback of this approach is the models are designed for individual word reading and not for reading connected text. Furthermore, rather than observing the reader's behavior with each word to model this

particular reader, these studies use simulated input to try to mimic known human behavioral characteristics.

The goal of this paper is to first quickly compare two models of how children learn to read, and then to use the better model to improve the ability of the ASR to listen accurately to children.

We first describe our approach to collecting and representing our data, and describe two candidate models of children's reading. We then compare which model better fits student performance as scored by the ASR. Finally to determine whether the student model can improve listening accuracy, we compare the effects of combining the student model and the ASR to better predict how a human transcriber judges words as read correctly or incorrectly.

2 Approach to Constructing the Student Model

In this Section we discuss the data used for experiments, our statistical framework for modeling, and the two models of reading we are investigating.

2.1 Data collected and representation

We collected data from 541 students working with a computer tutor that helps children learn how to read. Over the course of the school year, these students read approximately 4.1 million words (as heard by the ASR). 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 scores each word of the sentence as either being accepted (heard by the ASR as read correctly), rejected (the ASR heard and aligned some other word), or skipped (not read by the student). We use the terms "accepted" and "rejected" rather than "correct" and "incorrect" due to inaccuracies in the ASR. The ASR only notices about 25% of student misreadings, and scores as incorrectly read about 4% of words that were read correctly. Therefore "accept" and "reject" are more accurate terms.

One problem is determining how to score each word in the sentence text. As an example, suppose the student is trying to read the sentence "They are formed over millions of years and once depleted will take millions of years to replenish," and misreads "depleted," and stops reading after "will." Clearly the word "depleted" was read incorrectly, but what about the words "take" through "replenish?" It is odd to score these words as incorrect, since the student did not try to read them. However, the student stopped reading the sentence for some reason. Since his true reason for stopping is unknown, we assume the student had difficulty with the next word in the sentence where he stopped reading. So in the above example, the student would be considered to have misread "take."

Our heuristic for scoring the sentence words was:

1. For each utterance
    a. Start = position of first accepted word
    b. End = 1+position of last accepted word
    c. Use the ASR's accept/reject decision to score all words from Start through End as correctly or incorrectly read.
    d. Even if the ASR accepted a word, if the student hesitated more than 300ms, score that word as incorrect.
2. For each sentence word *w*
    a. Find the first utterance where *w's* position is between Start and End
    b. Use the ASR's score for *w* from that utterance. If nothing is aligned against *w*, score it as incorrectly read.

c. If a student requested help on *w* before it was accepted by the ASR, mark it as incorrectly read.
d. If *w* is not contained within any utterance, then it is not scored since the student did not attempt to read the word.

To continue the above example, if the student's second attempt at reading the sentence consisted of "take millions of years to replenish," then all of the sentence words would be accepted as read correctly except for "depleted" (since it was misread) and "take" (since in the first utterance that contains this sentence word nothing was aligned against the sentence word).

After using this methodology to combine utterances, and removing students who were not part of the official study, we were left with 360 students and 1.95 million sentence words that students attempted to read. 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.

2.2 Knowledge tracing

Now that we have determined how to score student attempts at reading a word as correct or incorrect, we must map those overt actions to some internal representation of the student's knowledge. Prior work in this area [10] has shown that knowledge tracing [3] is an effective approach for using ASR output to model students.

The goal of knowledge tracing is to map observable student actions while performing a skill (whether the student's response is correct or incorrect) to internal knowledge states (whether the student knows the skill or not). 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 students initial knowledge and to the probability of learning a skill given an opportunity to apply it, respectively. Two parameters, slip and guess, are called performance parameters and used to account for student performance not being a perfect reflection of underlying knowledge. The guess parameter is the probability that a student who has not mastered the skill can generate a correct response. For example, on a multiple-choice test with four response choices, a student with no knowledge still has a 25% chance of getting the question correct. The slip parameter is used to account for even knowledgeable students making an occasional mistake. For example, a student who when asked to multiply 4 and 3, could accidentally hit the keys in the wrong order and type "21."

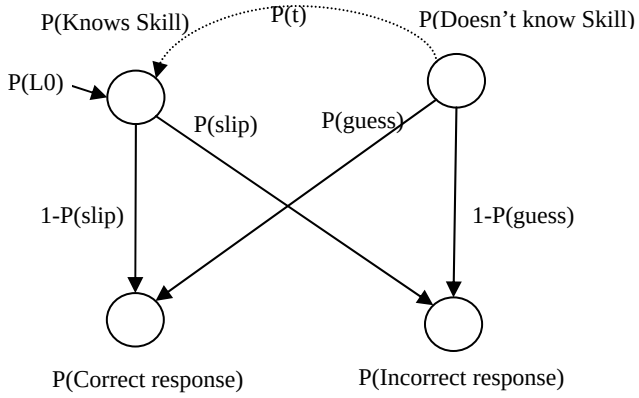

Figure 1. Overview of knowledge tracing

For each student and for each skill, knowledge tracing is maintains the probability that the student knows the skill. Knowledge tracing updates its estimates of P(knows) based on student performance. The approach is that whenever a student has an opportunity to apply a skill, observe whether the student performed the skill correctly or incorrectly.

The probability of P(knows) can then be computed via Bayes's rule [3]. In addition, the transition probability accounts for the expected increase in student knowledge due to the opportunity to practice the skill.

Knowledge tracing distinguishes between a student knowing a skill and getting it correct. P(knows skill) refers to the model's estimate of the student's internal knowledge. P(correct response) is derived from P(knows skill) and the performance parameters: P(correct response) = P(knows skill) * 1 – P(slip) + 1 – P(knows skill) * P(guess). Prior work on applying knowledge tracing to ASR output [10] demonstrated that the slip and guess parameters, in addition to accounting for variability in student performance, can also account for variability in the ASR scoring of student responses. Therefore, knowledge tracing is an appropriate modeling framework for this task.

2.3 Lexical and graphophonemic models

We considered two possible models for how students could learn to decode words. The first is a lexical model, which assumes that students learn words as a whole-unit, and there is no transfer between words. Although the assumed lack of transfer is somewhat naïve, it is likely that skilled readers recognize most words by sight. It is less clear, however, whether children learning to read have a similar representation as skilled readers.

The second model is a graphophonemic model, and assumes that rather than learning whole-words, students instead learn subword units. Specifically, it assumes that students learn the grapheme (letter) to phoneme (sound) mappings that make up words. For example, the word "chemist" contains the following grapheme→phoneme mappings: ch→/K/, e→/EH/, m→/M/, i→/IH/, s→/S/, and t→/T/. The grapheme→phoneme model is abbreviated as g→p model.

Given these two possible models, the next task is to determine which model is better described by our data under the knowledge tracing framework.

2.4 Evaluating the lexical and g→p models

To determine which model of student reading, lexical or g→p, better described student performance, we fit each of them to the student performance data as heard by the ASR (described above). First we split the students into two groups (to create a testing set to be used later). For students in the training set, we ordered each student's performance data chronologically. Then, for each model, we estimated the knowledge tracing parameters for each skill based on the student performance data.

For the lexical model, we simply treated words as skills. So each student attempt at reading a word was evidence for knowing the whole word or not. For the g→p model, we considered all of the g→p mappings in the word. If the word was accepted as correct, then all of the mappings were credited; if it was rejected as incorrect then all of the mappings were debited.

The lexical model had considerably more skills than the g→p model. There were 3210 lexical skill (i.e. words); in comparison there were only 295 g→p mappings encountered by students. As a result of this difference in number of skills, the g→p model had substantially more students encountering each skill on average (106 vs. 45). Table 1 describes the knowledge tracing parameter estimates for each of the models. These parameters are 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").

Note that 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 ITS (where typically "guess" is restricted to be less than 0.3). The reason for this difference is the uncertainty introduced

by the ASR scoring. This uncertainty is the reason the performance parameters under both models are similar: the parameters are (mostly) modeling the speech recognition rather than the student. Thus, the agreement in parameter estimates between the two models is not surprising. The column labeled R^2 in the table refers to how well the knowledge tracing parameters fit each skill.

Table 1. Mean knowledge tracing parameters

	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

At least within the framework of knowledge tracing, student performance is better described by the g→p model (R^2 of 0.48) than by the lexical model (R^2 of 0.34). Thus, the g→p model appears to be a better description of how children at this age acquire reading skills.

3 Leveraging the Student Model to Improve Speech Recognition

Although the g→p model is a useful way of viewing student performance and it provides a reasonable *description* of how students learn how to read, we would like to use the g→p student model to make *predictions* about how students will behave in order to improve the speech recognition system. For example, if the student model believes that the student has mastered the g→p mappings of the word "cat," but the ASR believes the student misread the word, perhaps we should ignore the ASR output and instead credit the student with reading the word correctly.

To evaluate possible improvements to the ASR, we had a skilled human transcribe a sample of the student utterances throughout the year. We followed the same protocol for aligning the sentence text against the transcription as we did for the ASR output, and similarly computed regions of the sentence we thought the student was attempting to read and counted the student's first attempt at reading the word. There were two differences from the prior procedure.

First, we excluded cases where the student requested help on the sentence word before reading it. Since the goal of this experiment was to evaluate whether a student model could improve ASR performance, help was a confound since it could be detected by neither the ASR nor the transcriber. Therefore, we simply excluded such trials.

Second, we insisted that the ASR and transcriber agree about which word the student was trying to read. Sometimes the ASR would be confused by background noise and get off-track. We were not trying to improve performance in these cases, so simply excluded them from the data.

Our approach was to treat the problem as classification. We used the second, testing, half of our data, so these data are not the ones used to perform the knowledge tracing parameter estimates of L0, t, slip, or guess. For the students in the testing set, we ran their data for the year through the knowledge tracing equations to determine skill estimates for each student for each g→p mapping. While tracing through a student's performance for the year, if a particular word had been transcribed, we recorded: 1) the student's knowledge at that point in time (before updating the knowledge tracing estimates of the student's knowledge for this attempt), 2) the ASR's accept/reject decision, and 3) whether the transcriber thought the student said the word correctly. The transcriber's scoring was the outcome variable for the classifier. We considered several types of features for the classifier:

1. The relative difficulty of the word for the student. We pretested students at the beginning of year on a variety of tests, including the Woodcock Reading Mastery's [11] Word Identification subtest, which gives a student's proficiency at reading words in grade equivalent terms (e.g. 3.2 means second month of the 3rd grade). We also had a heuristic that estimates the difficulty of the word on the same grade equivalent scale. The difference between these scores is the relative difficulty of the word for this student.
2. The student's proficiencies at the g→p mappings in the word. Since words have a variable number of mappings, we needed some way to get a constant number of features per word. We settled on extracting the student's proficiency on the following g→p mappings in the word: the first, the last, the one with the lowest proficiency, and the one with the highest proficiency. We also computed the student's mean proficiency across all the g→p mappings in the word, and product of P(knows) for a all the mappings in the word. In addition to computing those six features for P(knows), we also computed them for P(correct) (according to the guess and slip parameter estimates for the skill).
3. The ASR's accept/reject decision for this word.

The testing set contained nearly 1 million sentence words heard by the ASR. However, only 8,818 of those words were transcribed. Furthermore, these data were highly imbalanced, with 369 (4.2%) instances of students misreading a word and 8449 (95.8%) instances of correct reading. Although it may seem unusual for students to read 95.8% of words correctly (not counting those words on which they requested help), this level of performance is appropriate for material to help children learn to read.

For input to the classifier, we used several combinations of the above three groups of features: relative word difficulty, knowledge tracing features, the ASR, ASR + relative word difficulty, and ASR + relative word difficulty + knowledge tracing features.

The relative word difficulty is in essence a simple student model: how hard is this word for a student of this general reading proficiency; the knowledge tracing model is a more nuanced view since it accounts for variations in the student's knowledge. Thus, we can compare the relative benefit of using different levels of knowledge about the student.

Table 2 shows the results of the classification procedure. All results were generated using Weka's [12] REPTree fast decision tree learner's default settings, with bagging (10 bags) and a 20-fold cross validation.

The five most salient items from Table 2 are:

1. No approach did noticeably better than baseline (maximum difference 1.85%).
2. Majority class outperformed the ASR.
3. The only classifiers that beat majority class used the knowledge tracing features as inputs.
4. The student model was not able to improve classification accuracy by much. In fact, the best performer only classified 6 more cases correctly than the majority classifier.
5. Having the ASR as a feature hurt performance.

However, perhaps classification accuracy is not the best metric to use. Even though majority class outperforms the ASR, would it really be a superior scoring system to always assume the student read the word correctly? As a thought experiment, pretend that we had simply scored all student reading as being correct rather than using the ASR at all. It would have been impossible to apply knowledge tracing (or other student modeling approaches). Therefore, we would never have been able to get the small improvement in classification accuracy over majority class that we obtained by adding the knowledge tracing estimates. Although the improvement is very slight, it does exist. Given that the student model was

built from the ASR output, it must contain some signal that is being overlooked by a simple majority classifier.

Table 2. Classifier accuracy for predicting transcription

Features	Classifier accuracy
Knowledge tracing	95.88%
ASR + relative difficulty + knowledge tracing	95.84%
ASR + knowledge tracing	95.84%
Majority class	95.82%
Relative difficulty	95.79%
ASR + relative difficulty	95.78%
ASR (baseline)	94.03%

Furthermore, the primary goal of the ASR in a computer tutor is not to get high classification accuracy, it is to serve as a means to construct a student model to enable the tutor to select appropriate feedback and customize instruction for the student. It is unclear how assuming that the student is always correct can accomplish these modeling or teaching goals.

Perhaps lower classification accuracy is better if it enables the tutor to better model the student? One method of accomplishing this goal is to give different penalties to different types of classification mistakes. Unfortunately, it is difficult to specify *a priori* a good evaluation function that would lead to a good student model. For example, we could try different penalties for different types of classification mistakes, then compute how well we can model the student, and iterate. However, this approach is not computationally efficient. Furthermore, our ability to model the student depends on the domain being taught, known models for how students acquire the skills, etc. Therefore, it would be difficult to transfer research results tuned for one system to others.

One approach that sidesteps the problems of non-generalizable results and inventing penalties for various classification mistakes is to examine the Receiver Operator Characteristic (ROC) curves of the classifiers [13, p. 361]. Specifically, we investigate the Area Under Curve (AUC) of the ROC curves. AUC is a measure of the classifier sensitivity: how well does the classifier do at distinguishing instances of each target class.

Classifiers with a higher AUC are better than those with a lower AUC. A random (or majority) classifier will have an AUC of 0.5.

Table 3 shows the AUC for the classifiers shown in Table 2. The lower and upper bounds for the AUC were computed via SPSS's 95% confidence intervals making no parametric assumptions.

All of the AUCs are reliably superior to 0.5 (i.e. better than majority class). Therefore each set of features is able to distinguish the difference in likelihoods of the student making a mistake under different circumstances. Interestingly, all of the student models, even the simple model of relative word difficulty, were reliably superior to just using the ASR. The more complicated knowledge tracing model did not outperform the simpler model that just used word difficulty. However, there may be some slight gain from combining them with ASR (0.686 vs. 0.678). Therefore, it seems likely that both the simple and more complicated student model contain some independent information about the student's chances of reading a word correctly.

4 Contributions

This work extends prior work on testing models of reading in several ways. First, it applies the models to individual's data rather than to aggregate performance (as in [9]). Second, it examines students learning to read *in vivo* in the classroom rather than using simulated data (as in [9, 14]).

This work extends prior work on using ASR output to build student models (e.g. [10]). First, it considers using the student model to aid speech recognition. Also, rather than simply assuming a model of how children acquire reading skills, this paper examines the ability of the ASR to help select competing cognitive models reading (lexical and g→p models).

Table 3. ROC for various feature combinations

Features	AUC	Lower bound	Upper bound
ASR + knowledge tracing + difficulty	0.686	0.656	0.716
ASR + knowledge tracing	0.678	0.649	0.708
ASR + difficulty	0.678	0.648	0.707
Just difficulty	0.670	0.641	0.699
Just knowledge tracing	0.670	0.640	0.700
Just ASR	0.550	0.518	0.583

Compared to existing work on user modeling for (generally dialog) systems that use ASR (e.g. [15] and [16]) this work describes a richer model of the user. Two advantage of an ITS over many other systems that use ASR as input are that users work with the system for an extended length of time (8.5 hours in our study), and the system has a better idea of what the user is trying to do. Both of these features make for stronger user models.

5 Conclusions and Future Work

The ASR of a computer tutor for reading provides information about an individual student's reading development. The content of this information is sufficient to choose which of two possible models of reading development better describes the students using the tutor. Specifically, children learning to read are better modeled using subword properties (grapheme→phoneme mappings) than by treating words as atomic units.

The ASR is also powerful enough to construct a student model based on the student's past actions that can predict how the student will perform next—even when judged by a human transcriber.

Determining what constitutes good ASR performance in an ITS is complex, and classification accuracy can be misleading. Instead, AUC is a better metric for the actual task of the ASR: to provide a signal to customize instruction to the student. Using AUC as an outcome measure, the student model was able to improve the ability to hear the student's reading.

The method for constructing a student model from the ASR output is somewhat crude. Two areas of 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. [2]) would overcome this weakness. Similarly, the student's pattern of hesitation before a word contains a useful signal for modeling the student [17]. 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 his knowledge of g→p mappings.

One open question is rather than using the ASR to compare two competing models of reading, is to instead ask whether the ASR be used to determine for *which words* and for *which students* a particular model is appropriate. For example, it is likely as students become more familiar with a word they will treat it as an atomic unit (as in the lexical model), and rely on their knowledge of grapheme→phoneme mappings for less familiar words. In the future, we would like to study the ASR's capacity to detect such transitions.

Finally, this paper does not resolve the best method for combining the information contained in the student model (historical, averaged, data) and the ASR (current, noisy, data). For example, we demonstrated that for a new utterance, the ASR does not do as good a job at determining which words the student read correctly than the student model—even though the student model does not use any information from the current attempt! An obvious conclusion is to use the student model to second guess the ASR for the current interaction. Less obvious is how the student model should be updated. Should the student's estimates be decreased (according to the ASR's scoring) or increased (according to the student model's scoring)? If the latter option is chosen, there is a positive feedback mechanism built into the student model which could lead to instability: once the student begins to demonstrate knowledge (or lack of knowledge), his scores will have a built-in tendency to further increase (decrease). Intuitively, this mechanism does not sound like a good one. Perhaps it is necessary to decouple the scoring of the student's responses with one set of rules for determining feedback (student model + ASR), but just using the ASR to update the student model?

Acknowledgements

This work was supported by the National Science Foundation, under ITR/IERI Grant No. REC-0326153. Any opinions, findings, and 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. The author also thanks John Helman for transcribing many, many, student utterances.

References

1. Woolf, B.P., *AI in Education,* in *Encyclopedia of Artificial Intelligence*. 1992, John Wiley &Sons: New York. p. 434-444.
2. 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.
3. 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.
4. 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. p. 115-120: Erlbaum.
5. Heift, T. and M. Schulze, *Student Modeling and ab initio Language Learning.* System, the International Journal of Educational Technology and Language Learning Systems, 2003. **31**(4): p. 519-535.
6. Michaud, L.N., K.F. McCoy, and L.A. Stark. *Modeling the Acquisition of English: an Intelligent CALL Approach"*. in *Eighth International Conference on User Modeling*. 2001. p.: Springer-Verlag.

7. Newell, A. and H. Simon, *Human Problem Solving*. 1972, Englewood Cliffs, N.J.: Prentice-Hall.
8. Anderson, J.R., *Rules of the Mind*. 1993: Lawrence Erlbaum Assoc.
9. Harm, M.W., B.D. McCandliss, and M.S. Seidenberg, *Modeling the successes and failures of interventions for disabled readers.* Scientific Studies of Reading, 2003. **7**(2): p. 155-182.
10. 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.
11. Woodcock, R.W., *Woodcock Reading Mastery Tests - Revised (WRMT-R/NU)*. 1998, Circle Pines, Minnesota: American Guidance Service.
12. Witten, I.H. and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. 2000: Morgan Kaufmann.
13. Hand, D., H. Mannila, and P. Smyth, *Principles of Data Mining*. 2001, Cambridge, Massachusetts: MIT Press.
14. Seidenberg, M.S. and J.L. McClelland, *A Distributed, Developmental Model of Word Recognition and Naming.* Psychological Review, 1989. **96**: p. 523-568.
15. Horvitz, E. and T. Paek. *Harnessing Models of Users' Goals to Mediate Clarification Dialog in Spoken Language Systems*. in *Eighth Conference on User Modeling, Sonthofen*. 2001. p. Sonthofen, Germany.
16. Singh, S., D. Litman, M. Kearns, and M. Walker, *Optimizing Dialogue Managment with Reinforcement Learning: Experiments with the NJFun System.* Journal of Artificial Intelligence Research, 2002. **16**(5): p. 105-133.
17. Mostow, J. and G. Aist. *The Sounds of Silence: Towards Automated Evaluation of Student Learning in a Reading Tutor that Listens*. in *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*. 1997. p. 355-361 Providence, RI: American Association for Artificial Intelligence.

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.

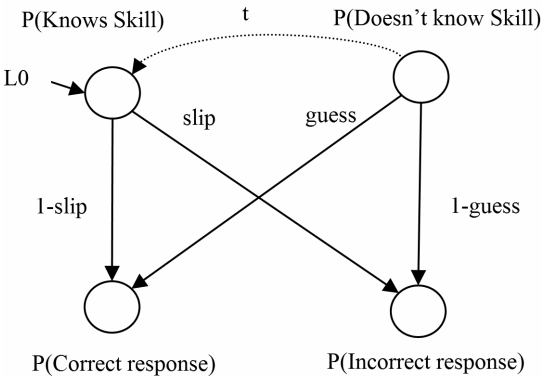

Figure 1. Overview of knowledge tracing. A set of L0, t, slip and guess parameters is estimated for each skill, while the internal knowledge state of a skill is traced for each student.

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.

Extensions to a Histogram-Based Student Modeling Approach to Facilitate Reading in Morphologically Complex Languages

Violetta CAVALLI-SFORZA
Language Technologies Institute
Carnegie Mellon University
violetta@cs.cmu.edu

Mohamed MAAMOURI
Linguistic Data Consortium
University of Pennsylvania
maamouri@ldc.upenn.edu

Abstract. We propose an approach to student modeling in the context of a project aimed at aiding readers negotiate authentic texts in languages where reading is particularly difficult due to the morphological complexity of the language, among other factors. We focus on Modern Standard Arabic as an example of such a language. Our approach extends existing tools for modeling reading skills, text difficulty, and curricula developed for English. We explore the extensions necessary for supporting Arabic morphology.

Introduction and Objectives

In this paper we describe our intended approach to student modeling for language tutoring in the context of a project titled "Teaching and Learning Linguistically Complex Languages", recently funded by the United States Department of Education under the Title VI International Research and Studies Program. The project aims to support foreign language learning and to enhance cross-cultural understanding by producing substantive textual and lexical learning materials and computer-based instructional tools that aid learners in reading authentic materials in languages that present special difficulties for reading. The specific goals of the project are:

(1) Providing readers with tools to negotiate the complex morphology of target languages;
(2) Enabling learners to read authentic texts containing unfamiliar and difficult words;
(3) Enabling teachers to prepare texts for classroom use and to test students' reading ability; and
(4) Creating easy Internet access to all tools and materials for teachers and learners.

While the tools themselves will be designed to address multiple languages, they will be implemented specifically to support Modern Standard Arabic (MSA), a less-commonly taught but critical language of high priority in Middle Eastern and North African studies. MSA's writing system and morphosyntactic structure present special challenges for the reader, particularly with respect to word identification and lookup in a dictionary. The planned tools address dictionary lookup, text preparation, and assessment of word recognition. To substantiate the claim that the tools do indeed generalize beyond MSA, they will be evaluated with a second less-commonly taught language, Nahuatl, spoken in southern United States and northern Mexico, which presents comparable – though different – word

identification and lookup challenges. For MSA, where we have access to substantial textual resources, we will also use the REAP technology, developed at Carnegie Mellon University's Language Technology Institute, to intelligently select texts to be presented to readers based on models of curriculum, text difficulty, student reading skills, and possibly topic interest. The REAP tools were originally developed to improve reading skills through individualized reading practice in English as a first or foreign language and will need to be extended to account for the special challenges presented by reading Arabic texts.

Work on the project will only begin in July of 2005, therefore this paper and our participation in the workshop has two primary objectives: 1) to describe the problem we are attempting to solve and the tools and approaches we are planning to use; and 2) to elicit feedback and learn from other workshop participants with respect to the student modeling component of our reading facilitation tools. Our proposed approach to student modeling, which is heavily based on REAP's histogram-based approach, does not attempt to address all aspects of language learning. Rather, it focuses on modeling specific aspects of reading skill in languages where even the basic process of word recognition presents special challenges due to the writing system and/or the morphosyntax of the language.

1. Reading Arabic Texts: Challenges and Tools

Modern Standard Arabic (MSA) is the primary, if not the only, formal written language used throughout the Arab world and is classified at the highest level of difficulty (level 4) in the United States Foreign Service Institute chart, requiring longer times for mastery than many other languages. Reading in Arabic presents special challenges due to its script. Learners of MSA – the main focus of Arabic teaching in the U.S. and elsewhere, and the only form of written Arabic – face difficulties in word recognition, word disambiguation, and the acquisition of decoding skills, which are important components of reading skill [1] [2]. Authentic Arabic texts lack short vowels and other diacritics that distinguish words and mark grammatical functions. Moreover, Arabic has a rich inflectional and derivational morphology that adds prefixes and suffixes and alters the stem of words according to syntactic context, and utilizes a number of particles (conjunctions, prepositions and pronouns) that attach to words as prefixes and suffixes.

The aforementioned linguistic complexities result in significant reading difficulties. In order to understand the precise meaning of a text, learners who are trying to read materials must insert short vowels and other diacritics themselves on the basis of limited vocabulary knowledge and on the basis of grammatical rules they have not yet completely internalized. To accomplish this, they must be able to recognize letter and word boundaries, decode unvocalized words, and identify and comprehend these words. For example, the word f^{c}lm/ فعلم is composed of the particle fa/ف and one of several possible words written as ^{c}lm/علم (such as: ^{c}ilm "science or knowledge", ^{c}alam "flag", ^{c}allam " (he) taught", ^{c}ulima "it was learned"), and it may play a different role in the sentence depending on unwritten vowels and other diacritic signs. Learners must bring knowledge of vocabulary, root-and-pattern morphology with complex derivational and inflectional rules, syntax, and contextual interpretation to produce correct and meaningful vocalization, to reach final word recognition, and even to look up a word in an MSA dictionary. It is worth noting that MSA presents reading difficulties even for schoolchildren in Arabic speaking countries. Their native language is a spoken dialect of Arabic, whose pronunciation, vocabulary and syntax can differ widely from MSA. Often MSA is their first written language and their first second language (in some areas of the Maghreb region, French has played this role at times). Arabic

schoolbooks begin with almost full diacritics and gradually decrease their use through the school years until they are entirely omitted by the end of middle school.

Arabic instruction is challenging for teachers and institutions as well as for learners. Though lately there has been an increasing demand for Arabic instruction in the U.S., and more educational institutions are beginning to offer introductory courses in Modern Standard Arabic (MSA) and some Arabic dialects, Arabic is still not a widely taught language. At higher levels of instruction, there is a shortage of pedagogically sound instructional materials and an insufficient number of teachers who have both the linguistic and technical skills and time to develop such resources, yet exposure to accessible, motivating and authentic materials is key in language learning. Technology is therefore increasingly being used to supplement the model of the teacher-fronted classroom and to foster learner autonomy by adapting instruction to the needs of individual students who may have specific career or academic objectives that require more rapid attainment of advanced language proficiency for better cross-cultural understanding.

To address the above challenges, our project will develop the following tools together:

(1) **Dictionary Lookup Tool:** enables language learners to look up the citation form of an arbitrarily inflected word in a morphologically complex language;
(2) **Reading Facilitation Tool:** enables language learners who encounter an unfamiliar word in electronic text to easily obtain a morphological analysis of that word, together with the dictionary entry for the citation form of that word;
(3) **Word Recognition Assessment Tool**: aids in the assessment of learners' reading ability, specifically the ability to choose the correct morphological analysis (and in languages where this is relevant, the diacritics) for each word, and its corresponding English gloss;
(4) **Text Preprocessing Tool:** designed to help teachers produce texts for use in the Word Recognition Assessment Tool.

The four tools will make use of the Buckwalter Morphological Analyzer, which has been partially developed at and is currently distributed by the Linguistic Data Consortium (LDC) at the University of Pennsylvania (www.ldc.upenn.edu). The project also leverages LDC's extensive and expanding Arabic language resources and in particular the Penn Arabic Treebank to provide a large database of texts for learners and teachers to chose from. The mission of the LDC is to continually collect and make available to the scientific community large quantities of linguistic resources, both text and speech, for Arabic and other languages. In addition to large quantities of raw Arabic text (LDC currently has more than 600 million-words of newswire text and adds 80 million words annually to its collection), it has already published three segments of the Arabic Treebank, with a fourth one close to completion. The treebank contains morphologically and syntactically annotated MSA text including newswire from the Agence France Presse, and the middle eastern newspapers *Al-Hayat* (distributed by Ummah Arabic News Text), *An-Nahar*, and most recently the Tunisian daily *Assabah*.[1] Several more segments of the Arabic Treebank are planned.

2. Supporting Reading Progress with REAP

In addition to building the aforementioned four tools to support reading practice, creation of prepared texts, and word recognition assessment, we plan to interface them with technology

[1] By end of Spring 2005, the Arabic Treebank will contain a total of 791,681 tokens representing about 1 million words after cliticization. The annotated corpora include complete vocalization including case endings, lemma IDs, more specific part-of-speech tags for verbs and particles, and an English gloss for each word.

developed by project REAP (http://orleans.lti.cs.cmu.edu/Reap/) to intelligently select texts for readers from an existing pool of materials [3] [4] [5]. REAP is funded by the U.S. Department of Education and includes researchers from Carnegie Mellon University's Language Technology Institute and the University of Pittsburgh's Learning Research and Development Center. The project aims to find appropriate authentic documents for students learning to read. It shares with our project the concern that too often students are given prepared texts, which has two disadvantages: first, the student is not exposed to examples of real language, that is, the language used in everyday written communication; second, the students all get the same texts to read, regardless of individual reading skills and interests. The REAP project, which was motivated by the desire to improve reading skills through individualized reading practice in the context of an English and ESL classroom/curriculum, is based on L1 reading research, but can be used for L2 reading as well. REAP has developed tools to a) retrieve texts from the Internet or from pre-existing collections that match different curriculum levels, b) model students' reading ability, and c) select texts that are suited to students' reading ability but also move them towards a higher level of reading skill (as defined by the curriculum) and/or pertain to topics of interest to the student or the teacher's lesson focus.

2.1 The REAP Approach to Student, Text, and Curriculum Modeling

There are four types of models in REAP: a curriculum model, two kinds of student models, and text models. REAP defines a reading curriculum with degrees of text difficulty in terms of vocabulary that a student should know at different curriculum levels. The student's knowledge is modeled as two histograms of words: 1) the **passive model**, which consists of all the words the student has read using the system, along with word frequencies – this can be considered exposure to words; and 2) the **active model**, which includes only the words for which the student has somehow demonstrated knowledge. Finally, texts are modeled by a histogram of word frequencies.

In order to present the reader with appropriate texts, a search engine is first used to look for texts that match that curriculum level/reading difficulty and may include other criteria, such as topic-specific vocabulary. For English REAP, the search is performed offline over the web, but it can also performed on a limited collection of texts in real time. To match documents to a student's level, the system then looks at words in the student's active and passive model and the words in the retrieved documents, selecting those documents that contain some subset of known words and some percentage of new words (the stretch). Stretch size can be experimentally manipulated. Once a set of documents appropriate to the student's reading level has been selected, they can also be ranked according to other criteria, e.g. words the student doesn't know but should in order to achieve curriculum level, or frequency of occurrence of these words, or topic of interest for a particular lesson.

2.2 Extending REAP for Arabic

The REAP project tools were developed primarily with English in mind. REAP currently uses unknown vocabulary, excluding named entities, as the sole criterion for modeling curriculum, student knowledge and text difficulty, although some extensions may be undertaken for other linguistic phenomena, and especially English constructions. The bare word models are extended with part-of-speech information. Words with multiple POS are

considered different words and, in fact, word cohorts – e.g., 'read' 'reading' 'reader' – raise issues in choosing documents for the student. This is currently a topic of research, to which experience with MSA's complex derivational morphology can contribute. Knowledge of vocabulary is certainly very important for Arabic learners as well, but must be modulated by other considerations. Morphologically, English is a (relatively) impoverished language, so a number of extensions will be needed in order to capture those aspects of Arabic writing and morphosyntax that make it difficult to decode and identify words and understand the role they play in a sentence. We envision the following major differences and extensions in the treatment of curriculum, text, and student models when applying REAP tools to Arabic.

Treatment of Named Entities: In English, names seldom affect comprehension. In Arabic, however, where there is no capitalization to distinguish proper nouns from regular words, identifying named entities is an important part of word recognition and text comprehension. Many adjective and noun forms are used as names, and their identification as proper nouns depends on knowledge of morphology and syntactic structure. A further problem is posed by the transliteration of foreign names into Arabic script: sometimes the resulting words are easily identified as foreign because they do not fall into the inflectional/derivational patterns of Arabic, but sometimes they do. To what extent it is desirable or feasible to model this problem remains to be determined and is likely to be of secondary priority: the best strategy could well be, at least initially, to make evident in the texts their special nature of named entities (they are specially tagged in the Arabic Treebank), allowing readers to focus on more general and pervasive morphosyntactic phenomena.

Modeling of Morphological Knowledge: Curriculum, texts, and student models, and the tools that operate on them, will need to be augmented with knowledge of inflectional morphological patterns. At this stage of our thinking, such patterns are best represented as collections of morphological features, including part-of-speech, and their surface realization for different categories of words, notably derivational patterns and words containing weak consonants. Included in morphological knowledge categories will be those closed parts of speech that attach themselves to words (e.g. direct object pronouns, conjunctions and prepositions), their effects on the surface realization of words (e.g. the preposition 'ل' causes an initial 'ا' to be elided), and constraints governing their attachment,. Modeling of derivational morphology skills (as exemplified by the patterns 'teach', 'teacher' in English, ᶜallam and muᶜallim in Arabic) will need to be left for later, since the Arabic electronic lexicon and morphological analyzer underlying the tools are stem-based and do not attempt to recover derivations from Arabic roots.[2] There is wisdom in using stem-based lexicons: while derivational patterns are quite regular, their accompanying derived meanings are often not.

Modeling Syntactic Context: To the extent made possible by the Arabic Treebank syntactic representation, we will model the syntactic context that affects morphological realization of words. While we do not expect to be able to cover the entire grammar, we will be able to model certain (local) phenomena, for example the omission of the definite article in all but the last term of a construct state ('iDafa'), or the rule that a verb preceding its subject does not need to agree with it in number (and even not in gender).

Updating the Active Student Model: While the passive student model can be updated by considering which words and morphosyntactic structures are present in texts the students have been exposed to, the active student model must be updated based on the knowledge demonstrated by the student. In REAP's use with English, knowledge is demonstrated by answering a question about a word; for Arabic, we will need to obtain this information from the Word Recognition Assessment Tool and/or the Reading Facilitation Tool.

[2] Ongoing work may however make this possible at a later date [6] [7].

The use of REAP tools with a morphologically complex language such as Modern Standard Arabic gives rise to an exciting synergy between projects. On one hand, REAP tools will aid the proposed LDC tools to select texts for learners according to pedagogically sound criteria; project team members will interact with language teachers at the University of Pennsylvania and the University of Pittsburgh to develop a curriculum defining levels of text difficulty. On the other hand, the addition of morphosyntactic analysis in modeling the curriculum, text difficulty, and learner ability, provides an opportunity to extend REAP tools in ways not afforded by their application to the English language alone.

3. Background and Qualifications of Authors

Neither of the authors is an expert in the area of student modeling per se, however they both bring relevant and complementary skills to the task. **Violetta Cavalli-Sforza** is a Visiting Researcher at Carnegie Mellon University's Language Technology Institute (CMU-LTI). As a doctoral student in Intelligent Systems at the University of Pittsburgh, and as a staff member and researcher at CMU-LTI, she worked on different aspects of tutoring systems and natural language processing. Her most recent research has focused on machine translation and Arabic morphology generation [6] [7] [8], some of which is being performed in Morocco through National Science Foundation and Fulbright fellowships. She is fluent in four languages, has studied a few more, is a permanent student of Arabic and is well acquainted with the difficulties in learning to read MSA. **Mohamed Maamouri** is a Senior Research Administrator and head of the Arabic Treebank project at LDC. Maamouri is a recognized Arabic language specialist, with significant experience in Arabic reading research, literacy research, and foreign language teaching and learning pedagogy [9] [10]. For over fifteen years he was the director of the Bourguiba Institute of Modern Languages in Tunisia where he started the well-known MSA summer intensive courses. Subsequently he worked as a senior researcher and the associate director of the International Literacy Institute, in the Graduate School of Education at the University of Pennsylvania. For the past three years, Maamouri has been leading the Arabic projects at LDC, where he has overseen and managed the preparation of extensive annotated corpora in Arabic.

References

[1] Perfetti, C. A. (1986). *Reading Ability*. Oxford University Press. New York.

[2] Perfetti, C.A & Hart, L. (2001). The Lexical Quality Hypothesis. In Verhoeven, L., Elbro, C. & P. Reitsma (Eds.), *Precursors of functional literacy*. John Benjamins. Amsterdam/Philadelphia.

[3] Brown, J. & Eskenazi, M. (2004). Retrieval of Authentic Documents for Reader-Specific Lexical Practice. In *Proceedings of InSTIL/ICALL Symposium*. Venice, Italy.

[4] Collins-Thompson, K. & Callan, J. (2004). Information Retrieval for Language Tutoring: An Overview of the REAP Project (poster description). In *Proceedings of the Twenty Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Sheffiel*d, UK.

[5] Collins-Thompson, K. & Callan, J. (2004). A Language Modeling Approach to Predicting Reading Difficulty. In *Proceedings of the HLT/NAACL 2004 Conference*. Boston, USA.

[6] Cavalli-Sforza, V., Soudi, A., & Mitamura, T. (2000). Arabic Morphology Generation Using a Concatenative Strategy." In *Proceedings of the 1st Meeting of the North American Chapter of the Association for Computational Linguistics* (NAACL 2000), 86-93. Seattle, USA.

[7] Cavalli-Sforza, V. & Soudi A. (2003). Enhancements to a Morphological Generator to Capture Arabic Morphology. In *Proceedings of the Eighth International Symposium on Social Communication*. Center of Applied Linguistics, Santiago de Cuba, Cuba.

[8] Soudi, A., Cavalli-Sforza, V., & Jamari, A. (2002). A Prototype English-to-Arabic Interlingua-based MT System, In *Proceedings of the Workshop on Arabic Language Resources and Evaluation – Status and Prospects*. Third International Conference on Language Resources and Evaluation (LREC 2002). Las Palmas de Gran Canaria, Spain.

[9] Maamouri, M. (1998). *Language Education and Human Development: Arabic Diglossia and its Impact on the Quality of Education in the Arab Region*, Preliminary Copy. Discussion paper prepared for The World Bank, The Mediterranean Development Forum, Marrakech, 3-6 September 1998.

[10] Maamouri, M. (forthcoming). Arabic Literacy. In *Encyclopedia of Arabic Language and Linguistics*, Vol. 2. Brill Academic Publishers. Leiden, The Netherlands.

Sequencing Vocabulary Instruction: Artificial vs. Real Users

Samuel R.H. Joseph
University of Hawai'i, USA
srjoseph@hawaii.edu

Stephen H. Joseph
University of Sheffield, UK
s.joseph@sheffield.ac.uk

Michael H. Joseph
University of Leicester, UK
mhj1@leicester.ac.uk

Abstract. There are various widely researched strategies that appear to be helpful in some, but not necessarily all vocabulary learning situations. However, an early report suggested that an extremely simple strategy, in which only the ordering of the material presented is varied, might have very substantial effects on learning and recall. These observations have been used as the basis of many subsequent developments, but rarely been subject to rigorous examination and replication. We have recently been examining both the theoretical foundation, and the practical implementation, of this latter approach. In this paper we present a comparison of data obtained using virtual users, operating in accordance with the underlying theory of memory, with the earlier experimental data obtained with real users.

1. Introduction

Sequencing of vocabulary instruction has tended over the years to be a somewhat imprecise art. With a few exceptions most examples in the psychology of learning literature have promoted a variety of heuristics to optimise the retention of vocabulary. One prominent exception is the work of Atkinson [1], which not only presents a detailed mathematical approach, but also a theoretical framework relating to the nature of memory. This work appeared to demonstrate that a teaching programme designed on the basis of that framework could dramatically improve retention rates in paired associate learning. Although the simple model of short-term memory employed in the work does not completely explain all subsequent experimental results, the original result remains a powerful demonstration of the possibilities of building a teaching system based on a well-defined model of memory. We have thus been led to examine the original model, to try to understand how it works and to see how it can be used to help support the design of learning programmes. After reviewing some related work this paper gives an overview of the Atkinson model, followed by our own analysis and the results of simulations using artificial users that precisely embody Atkinson's memory model.

2. Learning Vocabulary

One might argue today that the Atkinson Model is a limited model of vocabulary learning, because since it's creation in the 60's and 70's an extensive literature has developed on the many different factors that can affect vocabulary learning. For example there are results to indicate that associating sentences with vocabulary or requiring learners to perform generative tasks improves retention [5,10]. Other experiments have confirmed the widely known memory boosting effect of mnemonic strategies [11,15] as well as indicating that a scripted pair-learning/testing format can provide additional benefits [9,11]. Conversely, some studies have indicated that visual repetition of vocabulary items correlates negatively with performance [8], and emphasize the positive effects of meta-cognitive strategies such as "Self Initiation" and "Selective Attention". There is also a great deal of evidence to support the notion that

"distributed" practice is more effected than "massed" [4,6]. In addition de Groot [7] has shown that methods designed to encourage deep processing of vocabulary reduced retention loss two to three weeks after initial presentation.

As a result the benefit of replicating Atkinson's approach may not be immediately apparent. One might argue that the Atkinson model would be a poor choice for instructional design, since it does not explicitly handle long term memory decay, interference between items, or phonological encoding that might allow the advance prediction of errors. However, Atkinson obtained very striking improvements in vocabulary recall by using relatively simple strategies based upon the sequencing and frequency of representation of individual items during learning. His best results were obtained with an algorithm that required information on the difficulty of individual items for the target population, and the approach was based upon an explicit theory of memory function. Firstly it is important to determine whether such results are reproducible. If they are, then this alone could have important implications for practical vocabulary teaching, although naturally further work would be required in order to combine Atkinson's model with the other factors necessary to make a complete instructional approach. In addition we believe that Atkinson's method of model formation and testing is likely to complement a purely empirical approach, which can show whether one procedure is superior to another, but not why.

3. Paired Associate Paradigm

The paired-associate learning task is a standard procedure for assessing human explicit memory. For example, randomly paired elements such as words and letter strings are presented to subjects, who are then asked to recall one half of the pair from the corresponding other half, after which different types of feedback may be made available [19]. Rizzuto & Kahana [16] provide a summary of some different approaches to modelling of the paired associate learning task, as well as their own auto-associative neural network model. Nesbit & Yamamoto [14] showed that grouping together similar paired associates in sub-lists, caused subjects to generate more practise errors, but overall retention was better (around 20% over 64 test subjects).

In this paper we focus on the approach presented by Atkinson and Crothers [2] that was based on a model incorporating concepts of both short and long term memory. In their original study the predictions of different models of the day were compared with the results of a variety of different paired associate experiments, using tri-grams, Greek letters, digits and normal letters. Having demonstrated the explanatory superiority of a three state model that distinguished between long-term and short-term memory, as well as including interference based forgetting, Atkinson [1] showed how the model could be applied to vocabulary learning.

4. Atkinson Model

The Atkinson Model takes a multiple state memory model that effectively distinguishes between long-term memory (LTM) and short-term memory (STM). It is comparable to the Knowledge Tracing model of Anderson & Corbett [3]; the difference being an additional short-term memory state. In Atkinson's model paired associate items comprising of cue and response may be in LTM (state P), a permanent state, or in STM (state T) a temporary state where the association may be forgotten, becoming unknown (state U). The assumption is that a learner will give a correct response when presented with any cue from a paired associate item that they have in either state P or T. Conversely, if that item is in state U they will give an incorrect response.

	P	T	U
P	1	0	0
T	x	1-x	0
U	y	z	1-y-z

	P	T	U
P	1	0	0
T	0	1-f	f
U	0	0	1

Fig. 1: Presented Item (left), and Other Item (right) Probability Transition Matrices (P=Permanent, T=Temporary, U= Unknown), showing probability of transition from one memory state to another

The matrices in figure 1 show the probability of transition from one state to another with the left hand column being the state before presentation and the top row being the state after presentation. The presented item transition matrix in figure 1 is applied whenever an item is presented, e.g. if the presented item is currently in state U, then the likelihood of transferring to state P is y. The transition matrices are defined in terms of a number of parameters, x, y, z, and f which indicate how difficult it is to learn or how easy it is to forget each item. The second matrix is applied to those items that are not being presented, on each presentation that leads to an incorrect response. The implication is that interference from other items in the unknown state can cause an item to drop out of short-term memory. There is also a fifth parameter g which defines the probablity that a subject already has the item in state P before the start of the experiment.

Atkinson [1] created a teaching system based on this model that would choose items for presentation that were most likely to be transferred into the P state[1]. Given the sequences of correct and incorrect responses from the user so far, the model would estimate the likelihood of each item being in a particular state. Given knowledge of the transition parameters, the system could then infer which item, if presented next, would most likely be transferred to state P. The assumption was that items in state P would remain in permanent store and thus be available for subsequent recall a week later, while items in state T would not. In experiments using German-English word pairs Atkinson's optimal strategy condition significantly outperformed subjects selecting their own study order ("self selection"), and random presentation (fig 2). Atkinson's experimental procedure involved presenting seven sets of 12 German cue words in round robin fashion. Each list of German cues numbered 1...12 was projected onto the wall in turn, and the subjects were presented with the number of a cue on a teletype. The subjects would then type in what they believed was the English response, and the teletype would respond with the correct response. A delayed test session a week later was in a similar format, except no feedback was given. There were in fact two types of Atkinson algorithm, with one setting the x,y,z,f, and g parameters equal across all items; the other allowing them to vary. It was this latter algorithm, referred to as the "optimal (unequal)" approach that proved the most effective. The former or "optimal(equal)" approach performed similarly to the self-selection condition

[1] The precise equation is P(U)*P(U->P)+P(T)*P(T->P)

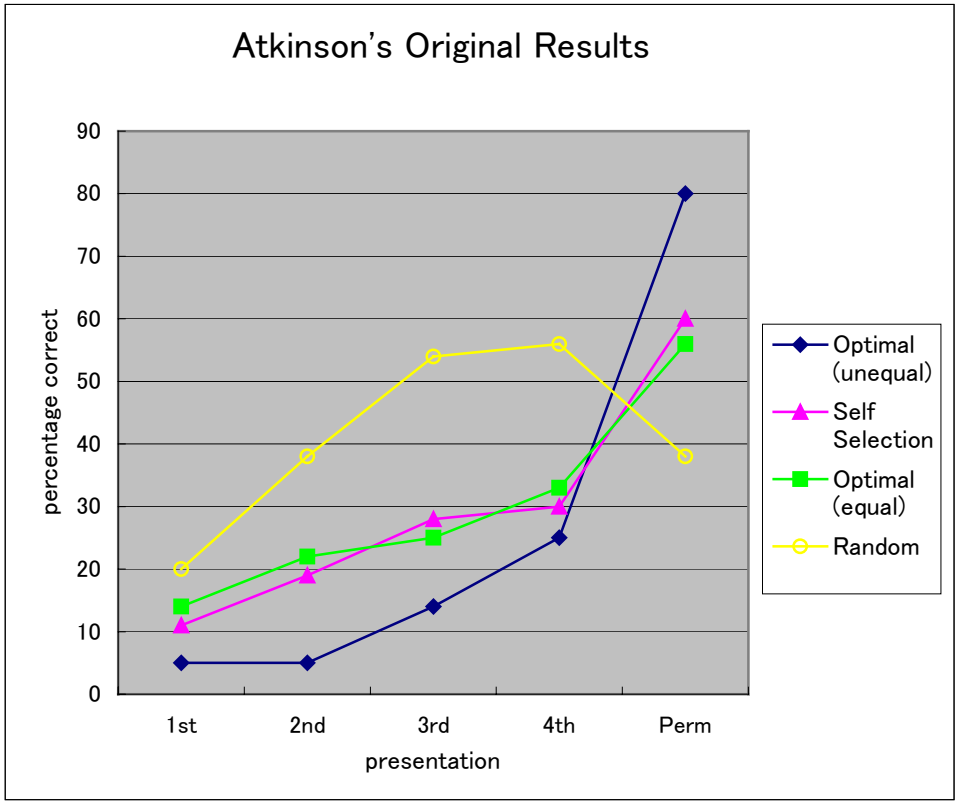

Fig. 2: Results reproduced from Atkinson's 1972 paper showing the percentage correct for the 1st, 2nd, 3rd, and 4th sets of 84 presentations (4 round robin repetitions of 7 lessons of 12 words) for each of the different experimental conditions (see text for more details)

The results also showed a clear inverse relation between the performance during training, and subsequent test, e.g. the random condition subjects performed best during instruction, but worst at test, while the optimal (unequal) condition subjects performed badly during training but were the best during subsequent recall. The contrast between training and recall is most remarkable under the optimal (unequal) condition (see fig 2). The performance of the subjects in the optimal (unequal) condition is in fact extraordinarily low during training. One possible explanation would be that the algorithm was re-presenting extremely difficult items again and again, such that most of the subjects responses were incorrect.

Setting the parameters for Atkinson's model requires pilot studies to be performed on the same word pairs. A minimization algorithm must then be employed to find the parameters that best fit the observed experimental data. In previous studies [12] we were unable to replicate Atkinson's results precisely. This may well be due to the lack of detailed information about the procedures used in the optimization process. There were other differences in our experimental setup such as the use of Japanese words as opposed to German, and alternative blocking of the lessons which must also be expected to have influenced our results. In Atkinson's experiments sets of 7 lessons were presented in round-robin order, so a subject would see one item from lesson 1, then one from lesson 2 etc. Our initial studies presented all items from lesson 1 before moving to lesson 2.

Subsequent communications with the original author have cleared up many of the ambiguities and we are confident that our current user studies come much closer to replicating the original experiments. Organising and managing human studies takes time and overhead, and in the meantime we have employed a mixture of analysis and simulated users to try and understand the Atkinson model in more detail.

5. Analysis of the Atkinson Model

The Atkinson model has influenced various authors, but with the exception of Katsikopoulos [13] few have replicated the algorithms in full. For example, while Seigel & Misselt [18] refer to Atkinson's work, they reject the use of a theoretical model in favour of a selection of heuristics based on instructional design strategies. Van Bussel [19] created a modified version of Atkinson's original procedure called the "a priori knowledge (APK) sequencing procedure", which more frequently presents items that are difficult to learn, as measured by the number of mistakes made by a particular user. This fits in with other work such as that of Schneider et al [17] suggesting that focusing on difficult tasks can lead to better retention. However Van Bussel's approach overlooks the fact that the most effective Atkinson procedure does not necessarily present the most difficult items more frequently, since it may in fact avoid presenting them at all if there are other items that have a higher likelihood of entering the permanent state. Van Bussel's results are extremely interesting however, showing that performance between the APK and fixed presentation strategies can only be distinguished if the users' self-regulated versus externally regulated learning styles are taken into account. As a result we plan to incorporate the same learning styles questionnaire used by Van Bussel into our current human studies.

The relationships between the original model, the heuristic alternatives, and the modified model remain unclear. The customary approach to the modelling problem is to formulate analytical or numerical solutions to the particular conditions of the experiment, and compare the predictions and results according to some chosen measure. We have found that it is possible to understand how the model works, and thus demonstrate some of its general properties, and so compare them with the operation of other systems, by considering the following points:

1. Paired associates can be thought of as being in one of three states:

    i. Un-tried: not yet presented to the subject
    ii. E-tried: presented and most recent response was erroneous
    iii. C-tried: presented and most recent response was correct

2. Let us first ignore the T state and any forgetting processes, then if g is the probability of a prior known and y is the probability of transition from U to P (P(U->P)) then the merit (i.e. the Probability of a transition to P if the word is tried) of the different types of word in the Optimal (Equal) condition are:

    i. Un-tried: (1-g).y = P(Unknown) * P(U->P)
    ii. E-tried: (1-y).y = P(U->U) * P(U->P)
    iii. C-tried: 0

3. If g<y, indicating the probability of learning an item is greater than the probability of already knowing it, Un-tried words will have the highest merit. Thus an Optimal (Equal) approach will present all the Un-tried items, followed by the E-tried items, continuing until everything has been responded to correctly - i.e. a correct response will lead to dropping that item from consideration for subsequent presentation. However one should note that there is no guarantee that the remaining E-tried items will be presented again with uniform frequency at any point, since they are all equally likely to be selected for presentation.

4. If g>y, indicating the probability of knowing an item is greater than the probability of learning it, E-tried words will have the highest merit. Thus an Optimal (Equal) approach will present Un-tried items until an erroneous response is received, after which it would focus on that item until it was responded to correctly. However it is important to note that the round-robin operation would prevent the subject from being presented the same item more than once every seven presentations so there would be a chance for items on other lists to enter the E-tried state.

5. Continuing with the same assumptions the merit of the different types of word in the Optimal (Unequal) condition now depend also on the individual variation of their parameters:

    i. Un-tried - easy to learn words (high y) and non-obvious (low g) will be tried first
    ii. E-tried - words of middling difficulty (y=0.5) will be favoured
    iii. C-tried - as above

6. Thus in the Optimal (Unequal) condition easy to learn and non-obvious words will be presented first, while C-tried items will be dropped as before. Sufficiently obvious and difficult to learn words may conceivably be excluded altogether. Once the set of suitably easy to learn and non-obvious Un-tried items have been presented then E-tried items will start to be presented with a general emphasis on those items with middling difficulty.

7. If we now include the T state. The merit of the different types of word in the Optimal (Equal) condition are:

    i. Un-tried - (1-g).y = P(Unknown) * P(U->P)
    ii. E-tried - (1-y-z).y + z.x = P(U->U) * P(U->P) + P(U->T) * P(T->P)
    iii. C-tried - if first round 0, or possibly non-zero if failure frequency is low[2]

8. Since forgetting only reduces T and increases U without changing P, the merit of an E-tried item will go up or down depending on the relative value of P(U->P) and P(U->T)

9. If x > y, i.e. P(T->P) > P(U->P), more recent E-tried items will have a higher merit, and the thus the Optimal (Equal) system will be likely to re-present recent E-tried items.

10. If y > x, i.e. P(U->P) > P(T->P), older E-tried items will have a higher merit, and the thus the Optimal (Equal) system will be likely to re-present older E-tried items over more recently presented ones.

11. It is interesting to note the difference between 9&10 and 4 above, in as much as while they both try to re-present E-tried items the effects of 9&10 wear off so that while the system described in 4 would keep repeating an item until it received a correct response,

[2] We remark that the effect of the T state will only be detectable if the frequency of presentation of a paired associate is of the same order as the forgetting rate; for the data this would only be so in the latest stages of the rehearsal, since as Atkinson's results show, the majority of presentations lead to incorrect responses and thus a high forgetting rate.

the parameters in 9&10 might be such that the system oscillated between presenting E-tried and Un-tried items.

12. Continuing with the same assumptions the merit of the different types of word in the Optimal (Unequal) condition now depend also on the individual variation of their parameters:

    i. Un-tried - easy to learn words (high y) and non-obvious (low g) will be tried first [as before]
    ii. E-tried - words of middling difficulty (y=0.5) will be favoured, as will words that are easy to learn using the T state (high x and z)
    iii. C-tried - are more likely to be repeated if easily forgotten (high f) and are likely to pass through the intermediate T state (high z)

13. Thus in the Optimal (Unequal) condition we are likely to see similar effects as 7 above, however certain categories of items will have a much higher likelihood of repeated presentation - those that are easily forgotten, and those that are likely to be learnt by passing through the intermediate T state. All this will be in combination with the effects described in 9&10.

The consequence of these considerations is that the Atkinson algorithm is not necessarily approximated by repeatedly presenting items in proportion to how many times the user has answered them incorrectly. The likelihood of an item being presented depends much more on the actual parameter settings associated with that item.

6. Using Artificial Users

In order to test our understanding of the Atkinson model we developed simulated users that embody precisely the theoretical basis of the Atkinson model. Specifically these users maintain a set of three states (P, T & U) and the transition of items between the three states takes place as described in the Atkinson model. Although the simulation did not model feedback to the user explicitly, the ability of the artificial user to learn an item implies that some sort of feedback must be present. Using these artificial users we were able to replicate a subset of Atkinson's original results (figure 3). We created seven lessons, each consisting of 12 pairs of nonsense words, and gave each pair x, y, z and f parameters selected randomly from a Uniform distribution between 0 and 1, with the additional constraint that $y+z < 1$. The g parameter was selected from a Uniform distribution between 0 and 0.4 to achieve similar first round success levels as seen in Atkinson's original results.

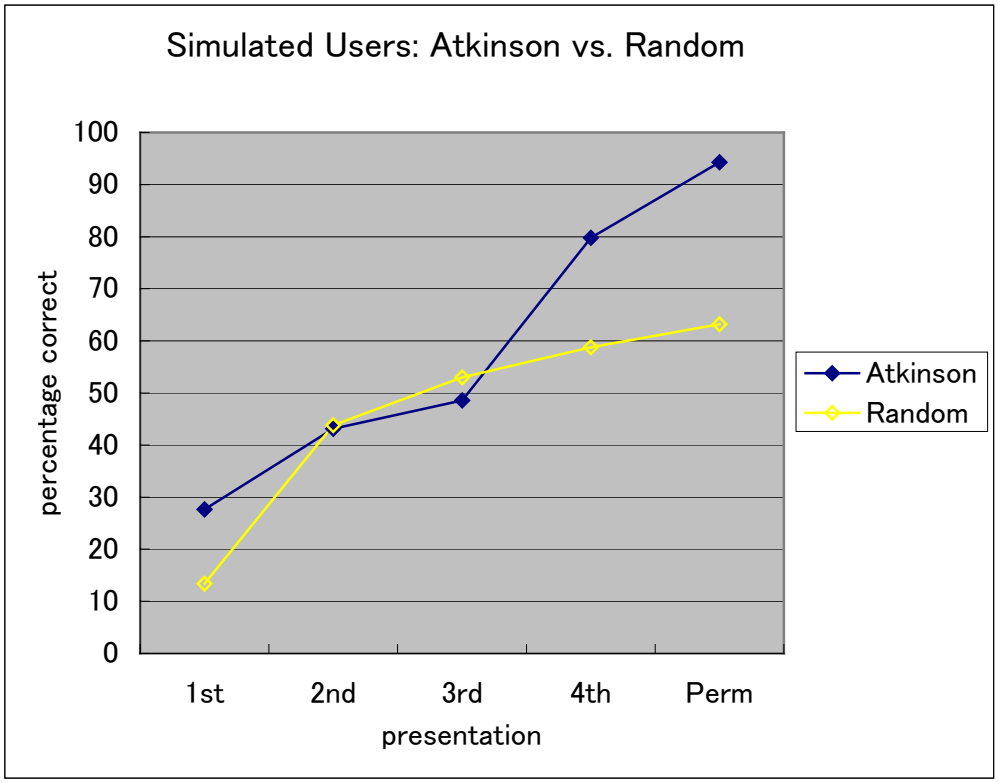

Fig. 3: Results generated by 5 artificial users that embody Atkinson's theoretical , showing the percentage correct for the 1st, 2nd, 3rd, and 4th sets of 84 Training presentations (4 round robin repetitions of 7 lessons of 12 words) for both of the different experimental conditions. The Perm presentation simulates the results of Test a week later, showing the percentage of items in the permanent state at the end of Train.

An inspection of an individual run bears out the predictions of our analysis, with the first 14 items presented all having y values in excess of 0.6. Interestingly all but one of them was responded to incorrectly, and half of them were never presented again. It was also clear that some items were repeated as much as 10 or 11 times, and these items tended to be easy to forget (high f), or ones that were easier to learn via the T state ($z > y$ and $x > y$).

An ANOVA showed that the differences between the number of items in the permanent state after the first Training presentation and the final testing round are significant, $F(1,4) = 13.919$, $p < 0.01$, $F(1,4) = 41.424$, $p < 0.01$, and confirm the potential efficacy of employing the Atkinson algorithm as opposed to a purely random presentation order. However the results indicate that the % correct order of the results at Test roughly reflect the ordering during Train, the inverse of Atkinson's original results. For these results to have mirrored Atkinson's original we would have expected the Random condition to do better on 4th round of training, and then worse in terms of the number of items in the Permanent state.

The extreme behaviour of the Atkinson algorithm, whereby many items are only presented once, and others are repeated again and again, does go some way towards explaining the remarkably high error rates in Atkinson's optimal condition, along with the subsequent high performance at test. The algorithm is presenting items that can be learnt on a single trial, where the user likely makes a mistake, but the algorithm anticipates that the item has been learnt and need not be presented again, thus the high error rate – i.e. the items that the user would answer correctly are not presented again, and the algorithm focuses on other items that are easily forgotten, or are learnt via the short term memory route. Items are thus transferred to the Permanent state without being presented any more times than necessary.

7. Discussion

The artificial user studies not only provide an essential check that the algorithms are correctly implemented, but also test the model under ideal conditions. The improved performance of real users under the Optimal (Unequal) condition may not be due to the validity of the Atkinson model for those users. For the artificial user, however, the improvement *due only to the model* can be measured accurately, and so separated out from that which is a by-product of the condition for the real user. It appears from our results that the effects of the Optimal (Unequal) condition on the real user are not entirely due to the model. Specifically the inverse relationship between performance at test and train present in Atkinson's experiments does not appear in our simulations or interpretation of the model.

This conclusion remains provisional, however, until the completion of our trials with real users, and of more extensive investigations of artificial users. It would seem reasonable to expect that real users cannot be completely modelled by a three state memory model with constant transition rates. However, the contribution of this model is hard to elucidate in the complex context of real teaching programmes on real subjects. The facility for the creation of larger numbers of artificial users, with appropriate distributions of parameters, and subsequent testing of their performance under various programmes, will be of great assistance in critiquing the model and furthering its development. In this way we hope to extend the range of real responses that can be modelled, and perhaps explore the limitations of what is possible with such models.

Acknowledgements

Many thanks to the anonymous reviewers for insightful feedback, and to Luke & Aya Joseph for support during the writing of this paper.

References

[1] Atkinson, R. (1972) Optimizing the learning of a second language vocabulary. Journal of Experimental Psychology, 96, 124- 129.
[2] Atkinson, R. & Crothers, E. (1964) A comparison of paired-associate learning models having different acquisition and retention axioms. Journal of Mathematical Psychology, 1, 285-315.
[3] Corbett, A. and J. Anderson, (1995) Knowledge tracing: Modeling the acquisition of procedural knowledge. User modeling and user-adapted interaction, 4: p. 253-278.
[4] Cull, W.L. (2000) Untangling the benefits of multiple study opportunities and repeated testing for cued recall Applied Cognitive Psychology 14 (3): 215-235
[5] Grace, C. (1998) Retention of word meanings inferred from context and sentence-level translations: Implications for the design of beginning-level CALL software. Modern Language Journal 82 (4): 533-544.
[6] Greene, R (1989) Spacing effects in memory: evidence for a two-process account. Journal of Exp. Psy.: Learning, Memory & Cognition, 15 (3): 371-377.
[7] de Groot AMB, Keijzer R (2000) What is hard to learn is easy to forget: The roles of word concreteness, cognate status, and word frequency in foreign-language vocabulary learning and forgetting Language Learning 50 (1): 1-56
[8] Gu, Y & Johnson R (1996) Vocabulary learning strategies and language learning outcomes. Language Learning, 46 (4): 643-679.
[9] Hansen, L., Umeda, Y. & McKinney, M. (2002) Savings in the relearning of second language vocabulary: The effects of time and proficiency. Language Learning, 52 (4): 653-678.
[10] Joe, A (1998) What effects do text-based tasks promoting generation have on incidental vocabulary acquisition? Applied Linguistics, 19 (3): 357-377.
[11] Jones, M., Levin M., Levin, J. & Beitzel, B. (2000) Can vocabulary-learning strategies and pair-learning formats be profitably combined? Journal of Educational Psychology. 92 (2): 256-262.

[12] Joseph, S., Smith Lewis, A. & Joseph, M.H. (2004) Adaptive Vocabulary Instruction. IEEE International Conference on Advanced Learning Technologies, 141-145.
[13] Katsikopoulos, K.V., Fisher, D.L. (2001) Formal requirements of Markov state models for paired associate learning. Journal of Mathematical Psychology 45 (2): 324-333
[14] Nesbit, J.C. & Yamamoto N. (1991) "Sequencing Confusable Items in Paired-Associate Drill" Journal of Computer-Based Instruction, 18-1, 7-13.
[15] Raugh, M.R., Schupbach, R.D. & Atkinson, R.C. (1977) Teaching a large Russian language vocabulary by the mnemonic keyword method. Instructional Science 6:100-221.
[16] Rizzuto, D. & Kahana, M. (2001) An autoassociative neural network model of paired-associate learning. Neural Computation, 13 (9): 2075-2092.
[17] Schneider, V., Healy, A. & Bourne, L. (2002) What is learned under difficult conditions is hard to forget: Contextual interference effects in foreign vocabulary acquisition, retention, and transfer. Journal of Memory and Language, 46 (2): 419-440.
[18] Siegel, M.A. & Misselt A. L (1984) Adaptive feedback and review paradigm for computer-based drills Journal of Educational Psychology 76(2):310-317
[19] Van Bussel, F.J.J. (1994) Design rules for computer aided learning of vocabulary items in a second language. Computers in Human Behaviour 10:63-76

MAC: An adaptive, perception-based speech remediation s/w for mobile devices

Maria Uther [a,1], Pushpendra Singh [a], Iraide Zipitria [a] and James Uther [b]

[a] *Speech and Audio Interaction Lab (SAIL), Department of Psychology, University of Portsmouth, U.K.*

[b] *Department of Information Technology, University of Sydney, Australia*

Abstract. In this paper, we present a mobile adaptive computer assisted language learning (MAC) software aimed to help Japanese-English speakers in perceptually distinguishing the non-native /r/ vs. /l/ English phonemic contrast with a view to improving their own English pronunciation in this regard. The software is adaptive and more practice is given for the learner on contrasts that are most difficult for them, and the learners themselves choose their level of adaptation. MAC is implemented in Java (J2ME), allowing the software to be used on a wide range of mobile devices including most recent mobile phones. This allows the application to be used anywhere and anytime, on a device that the learner probably already owns. We first discuss the theoretical background underpinning this work, followed by a discussion of the software and some of the constraints for adaptive tutoring on mobile devices.

Keywords. Adaptation, Language learning, Mobile Phone, PDA,

1. Introduction

In the Automatic Speech Recognition (ASR) field, Second Language (L2) learning applications primarily focus on pronunciation improvement using voice recognition [1]. A problem with this approach is the difficulty of scoring learner responses with certainty. An alternative way of approaching pronunciation improvements is instead to tackle the underlying *perceptual* difficulties rather than focus on quantifying/scoring the learner's speech production. This approach derives from several psycholinguistic studies which have highlighted a link between the *perception* and *pronunciation* of phonetic contrasts [2,3,4]. Specifically, for the case of Japanese /r/-/l/ discrimination, work by Bradlow and colleagues have shown that training individuals to perceptually discriminate /r/ and /l/ contrasts results in improved pronunciation [5,6]. The approach of tackling underlying perceptual difficulties in Japanese /r/-/l/ discrimination has been shown to be highly effective [7,6] resulting in improvements in discrimination to around 82 percent accuracy from 65 percent and lasting 3 months [6] from a training period of 15-22 hours over 3 weeks.

[1]Correspondence to: Dr. Maria Uther, Department of Psychology, University of Portsmouth, PO1 2DY, U.K. Tel.: +44 23 9284 6330; Fax: +44 23 9284 6300; E-mail: maria.uther@port.ac.uk

1.1. Theoretical foundations

The approach taken in the previously mentioned /r/-/l/ training studies was to train on as wide a phonetic variability as possible. This involves providing the learner with speech samples across a full range of linguistic variability that s/he would be expected to encounter in everyday language. This 'high-variability' training provides presentations of different samples from different speakers and using different types of words. In terms of word-type variability, the target contrast can occur at the beginning or elsewhere in the word as in 'rake' vs. 'lake', 'blew' vs. 'brew', etc. This linguistic variability was used by Logan et al. [7] and later also in other studies [8,6] with effective and long-lasting outcomes. On the basis of the success of these results, word type variability would appear to be a highly important factor in successful training. The variation of word type has thought to be necessary as studies have suggested that /r/-/l/ discrimination across different word types is not equivalent. In general, identification performance is worst for /r/ and /l/ contrasts in initial positions and performance is best for /r/ and /l/ in final positions of the word [9]. Variability is thought to be a way of ensuring that the learner sees the perceptual constancy across a wide range of environments.

Furthermore, it has been argued that the failure of early attempts [10] to provide generalisable effects may (at least in part) be due to the fact that only a single phonetic context was trained [7]. Apart from word type variability, talker type variability has also been highlighted as an important factor by Lively et al. [8] who showed that the addition of talker variation was important in demonstrating robust and *generalised* improvements in /r/-/l/ discrimination. This result has also been validated by many subsequent studies demonstrating successful and long-lasting training effects [11,6]. Talker variability may be important because different talkers produce outputs that vary acoustically due to different vocal tract size and shapes, speaking rates and glottal source functions (see Logan et al. [7] for a review).

Our Mobile Adaptive CALL (MAC) software builds directly on the work done using a high-variability training approach. It also (as with the reviewed high-variability studies) tackles the problem of Japanese /r/-/l/ discrimination which is notoriously difficult for Japanese language learners of English [2]. We advance on current high-variability approaches in CALL for /r/ vs /l/ training by first exploiting the dimensions used as part of that approach to tailor the training for the learners. There are several justifications for providing an *adaptive* training based on the high-variability approach. Firstly, as has been argued in other research, a more personalised tutoring approach has been shown to be a particularly successful strategy to train learners [12]. Secondly, there is strong evidence to suggest that although Japanese speakers will at a *general* level tend to be worse at certain word positions, at an *individual* level there is a large degree of variation in the discrimination of /r/-/l/ for different word and talker types [13]. Developing a model of these individual differences would appear to be a way of more effectively delivering this training. This adaptation can tap into the learner's possible weaknesses with respect to a talker's voice, e.g. male or female talker, or a characteristic of the word, e.g. positioning of /r/ or /l/ within the word. Based on these premises, MAC aims to adapt to individual learners' needs in order to focus training in areas that are most difficult for the individual. MAC uses adaptive, personalized training whilst maintaining the use of natural speech samples and high phonetic and talker variation.

Recent work on /r/-/l/ training by McClelland and colleagues [14,15] used a novel alternative to the high variability approach. They demonstrated successful training in

Japanese speakers with an adaptive system which instead exaggerated /r/ and /l/ differences on a continuum using semi-synthesised speech. With their approach, they found improvements of up to 30 percent using relatively short periods of training. McClelland et al.'s results were quite dramatic. However, the extent to which these effects would generalise to other natural speech contexts has not yet been shown. Indeed, on the basis of previous work using synthesised stimuli [10], there is some reason to believe that effects on synthesised speech may not necessarily generalise to natural speech or to other phonetic contexts. This possible limitation was also acknowledged by McClelland et al. themselves, even though they demonstrated generalisation to another untrained semi-synthesised continuum.

Nonetheless, the success of the approach used by McClelland and colleagues could also be interpreted in a similar light to the theoretical foundation underpinning the high-variability training studies. The success of the high-variability approach is generally explained in terms of current 'attention to dimension' or 'A2D' models of speech perception. Within this view, the effective learning of a new phonetic category results in the 'stretching' of perceptual spaces that are the target of focussed attention and 'shrinks' the perceptual spaces for nonattended. The training is thought to affect the learner's perception so that their attention is directed towards dimensions that are relevant for classification and conversely away from dimensions that are irrelevant [16,17,18].

1.2. Mobile technology in learning

Another novel aspect of our approach is to use mobile technology to deliver the training. Previous studies using /r/-/l/ training (at least to our knowledge) have only been delivered on traditional PC-based platforms. This has significant drawbacks for the learner: s/he would need to attend training sessions at a fixed location (usually by means of attending a lab every day for a fixed period) and would not be able to practice in their own time. Even if implemented as an Internet application, this generally still makes the application less accessible and potentially costly. Learners with unlimited web access still have to boot up and use a PC. Those using PDAs/phones with Internet connectivity would still have to pay for Internet access. In contrast, a large percentage of the population see their phone as a personal, trusted device that they always carry. MAC provides the opportunity of downloading the application to a Java-enabled phone that the learner may already own. If learners were able to practice in their own time, whenever and wherever they wanted to (e.g waiting for or on public transport, sitting at home), then it would likely provide much quicker and more effective training.

Recent work has capitalised on these advantages producing several novel mobile learning applications [19,20,21]. Mobile learning has several potential advantages over class-room learning, but the most obvious is the availability of application (i.e. pervasiveness). Being so readily available, learners can therefore choose to engage in learning activities across a greater time span. Another consideration (particularly for mobile phones) is that there is a natural affordance for phones to speak into and listen from (i.e. audio interaction). In this way, applications that involve audio interaction may actually be better suited for mobile phones than for the PC environment. Modern handsets are often used for games and push-to-talk. For these types of mobile applications, the audio is often played using a speaker functionality. MAC also follows this interaction style (but also could be used in conjunction with a headset).

In this paper, we present the design of an adaptive speech remediation software aimed at training native Japanese speakers in distinguishing /r/ and /l/ contrast using a high-variability approach. The MAC software adapts according to the learner's responses and presents to the learner a contrast of the type on which they will most need further practice. The software is designed for J2ME-compliant mobile phones. Firstly, we give an overview of the software functionality in section 2, then we detail the learner adaptation strategy and discuss issues relating to learner control in section 3. In section 4, we discuss implementation issues including constraints in adaptive tutoring in mobile devices and in the final section 5, we conclude with a discussion of future directions for development.

2. Overview

2.1. Stimuli

As stated earlier, one dimension of the high-variability approach is the position of the contrast within the word. In linguistic terms, we could broadly divide /r/-/l/ minimal pair word types into five categories [1] as follows:

1. Final singleton: The contrast appears at the end of the word as a single consonant. For example: 'tire' vs. 'tile'.
2. Initial singleton: The contrast appears at the beginning of the word as single consonant. For example: 'lead' vs. 'read'.
3. Intervocalic: The contrast appears in the middle, in between two vowel sounds. For example 'miller' vs. 'mirror'.
4. Initial cluster: The contrast appears at the beginning of the word as part of a consonant cluster. For example: 'blue' vs. 'brew'.
5. Final cluster: The contrast appears at the end of the word as part of a consonant cluster. For example: 'wild' vs. 'wired'.

2.2. Talkers

In MAC, we use four talkers (two male, two female) were used to create the stimuli, and these form the second dimension upon which we can adapt the application for the learner.

2.3. MAC operations

MAC software operations can be summarized as follows:

1. Under Options, we have *play* and *exit*. On pressing *play*, the learner is presented with a random word articulated by a random talker. The learner's task is to identify which word they thought they heard by clicking one of two buttons (left and right navigation buttons) corresponding to either of the two written words apearing on the screen. A screenshot of this can be seen in Figure 1.

[1] Although there are five possible categories, our initial work has been based on 3 and then 4 categories omitting the final singleton and final cluster category for expediency, especially as the research shows that these are the easiest ones for Japanese speakers to discriminate.

Figure 1. Screen shot of our MAC software from a Nokia Series 60 SDK

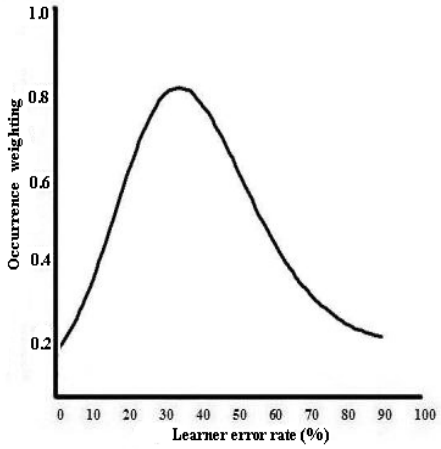

Figure 2. Occurrence weights for next trial as a function of past learner error rate (default critical error rate set as 40%)

2. The learner is give immediate feedback as to whether they made a correct or incorrect choice with either a chime or a buzz.
3. The learner can increase or decrease the volume by using the up or down arrows respectively and can repeat the sound by pressing *play*.
4. The learner can also select different degrees of difficulty at the outset and can change it at anytime during the test.
5. The learner's responses are monitored by the software and this forms the basis of the student model as described in detail in the following section.
6. The choice of the next presentation is based upon a tutor algorithm taking into account the learner's error rate for each token type and the learner's preferred difficulty level.
7. All data used by MAC is stored on the device itself, using the persistent memory storage.

3. Student modelling and learner adaptation

One of the main goals in student modelling is to adapt the system's behaviour to individual needs and preferences [22,23]. Within MAC, we have designed the system so that it presents the learner with more trials in the area where they make most mistakes, but only up to a point at which their motivation is not affected. Within classical approaches to designing ITSs, one can separate the student model (i.e. the system's beliefs about the student), the pedagogical model (i.e. the tutoring strategy) and the domain/expert knowledge (i.e. all possible knowledge about the domain). In MAC, the domain model is fairly simple - it is essentially expressed by the database of correct answers, which is not unusual for a mainly procedural rather than declarative knowledge base. Similarly, the student model is also fairly simple: it reduces to knowledge of the student's profi-

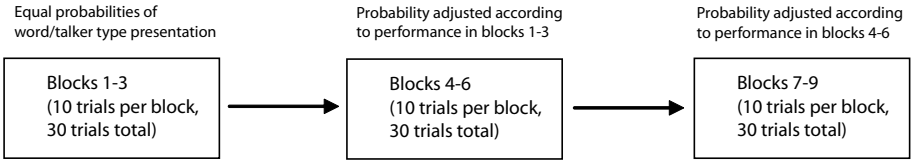

Figure 3. Adaptation as a function of blocks of trials

ciency on all word/talker type combinations. This student model bases itself on data from the most recent 30 trials (3 blocks of 10). This is done mostly for expediency as it was felt necessary to (a) have as current data as possible and (b) to have a statistically large enough number of trials on which to base the model. Admittedly, there were other possible approaches, such as to weigh old and new information, but we wished to start with an initial base version from which we could further develop, particularly given the limited computational power of a small device. Furthermore, the speech perception literature would suggest that having fine-grained information about the student's old performance is actually of very little value.

Figure 2 shows adjustment curve that supports the pedagogical approach in MAC and determines the selection of which stimuli to train on, based on information in the student model. As shown in Figure 2, the probability for a word type or talker being chosen should initially increase if the learner makes more errors in identifying tokens of that word or talker type, until it reaches an error rate of 40% (by default). If a learner's error rate rises above 40%, then the probability of that particular category or talker being chosen should start to decrease. This is done in order to ensure that the learner's motivation is not affected, as too many errors by the learners would undoubtedly be highly demotivating. Moreover, the peak critical error rate can be changed by the learner, between values of 20% to 50%, for reasons discussed later in this section.

The adaptation algorithm monitors learner responses and based on this, adjusts the probabilities of each token type being chosen for the next block of trials (shown in Figure 3). The student model is updated and checkpointed after each 10 trial block, and never within these blocks. This is done so that the learner is able to exit the application at any time without loss of too much data. Ten trials was chosen as a reasonable tradeoff with more complex (and therefore computationally intensive) checkpointing and exit code.

Two dimensions of learner response are taken for the adaptation:

- *Talker*: Out of four talkers, any can be randomly chosen. Learner responses are monitored to determine which talkers the learners find difficult to make the relevant discrimination.
- *Word type*, outlined in section 2.1: In a similar way to the talker dimensions, words can be chosen from any of the word types and learner responses are monitored to determine which type is more difficult for the learner to make the relevant discrimination.

As can be seen in Table 1, this depicts an example where there are $4 * 4$ possible /r/ and /l/ tokens types, yielding a $1/16$ probability for any speech token type combination $T_i W_j$ being presented if all are equally weighted. Within each of these token type combinations, there have been (within our early trials) 30 actual word sample pairs under

Table 1. Possible /r/-/l/ token type combinations for each word and talker types where T=talker type and W=word type for an example where there are 4 talkers and 4 word types

	W_1	W_2	W_3	W_4
T_1	T_1W_1	T_1W_2	T_1W_3	T_1W_4
T_2	T_2W_1	T_2W_2	T_2W_3	T_2W_4
T_3	T_3W_1	T_3W_2	T_3W_3	T_3W_4
T_4	T_4W_1	T_4W_2	T_4W_3	T_4W_4

each word type, forming several hundred possible speech samples in the database. Even though only 30 trials are taken into consideration for the student model, since the talker and word type error rates are assumed to be independent, then a given trial gives data for both the talker and word type independently. In this way, any specific combination T_iW_j need not actually be presented to make the decision that it would be problematic for that student if there were data suggesting that other T_i samples and other W_j samples were difficult.

On the first three blocks of trials, no adaptivity is applied. This means that the presentation of all categories and all talkers are equiprobable. The error rates from the student model are scaled according to the polynomial function below, which is also shown in Figure 2:

$$l_n(P) = a_0 + a_1 * x + a_2 * x^2 \tag{1}$$

Here x is the error rate from the student model and a_0, a_1 and a_2 are constants chosen to give a curve which adjusts the degree of difficulty of the training while maintaining optimum motivation. Although presumably, the same ends may have been achieved with other functions (e.g. stepwise), a curve seemed best suited for a more gradual adaptation. For the default difficulty level (peaking at 40%) the constants are given values 2.5357, 0.0902, and -0.0011 respectively. P is then the weighted value for that category. There is also an additional constraint that any word or talker type should be given a minimum weight of 10 and the rest of the probabilities are scaled accordingly.

Ideally, adaptive tutoring systems should give the learner as much autonomy as possible while s/he is making good progress but at the same time offer control when things are not working well for them [24]. Giving students control over how their model is used has already proved a successful approach [25,26,27]. Although MAC does not allow the learner to fully inspect all data in their own student model, it still gives the learner some degree of control by letting them choose the degree of difficulty they consider to be adequate. In MAC, learners have the possibility to choose the degree of difficulty moving the 40% threshold peak of the model on a range that goes from 20% to 50%, an advance also on earlier prototypes of the software [28].

4. Implementation issues

Mobile devices have not yet converged on a small number of dominant software platforms as have larger computing devices. For this reason it is often difficult to justify programming a specific device in ‘C’, as this severely limits the number of devices the software supports. It is also the case that native programming environments for mobile devices are not as polished as for the PC, with Symbian in particular noted for unusual

APIs. On the other hand, the most widely available cross-platform programming environment for mobile devices, Java J2ME has its limitations. It is much slower, uses more battery, has a much reduced class library, missing language features such as floats, and has very limited access to device cababilities. However, since we had limited time for this initial implementation, and wished to test the software on a number of devices, J2ME (MIDP 2.0) was chosen. We encoded the speech samples using the AMR codec, although we plan to migrate to the WB-AMR codec for better sound quality.

We have run the software on the Sun WTK emulator and Nokia emulators. We have also run the software on a Nokia 9500 communicator handset, and plan also to run it on a Nokia 6630. Our application can also be theoretically run on any mobile phone which is J2ME (MIDP 2.0) compliant as long as it supports the correct codecs. Furthermore, initial tests of the MAC algorithm with a control group of users (native English speakers trialing the application) have found a good fit between the observed and predicted behaviour of the MAC adaptation.

5. Conclusion & future work

This paper suggests an approach to speech remediation which is founded on several years of psycholinguistic research. The implementation of an adaptable mobile device software is a novel contribution to the field of CALL by both allowing the learning to be personalized to learner needs and also allowing portability. Our adaptive software (MAC) was developed with a view to help Japanese speakers of English distinguish /r/ and /l/ sounds, adapting so it changes itself according to the learner's needs. For future work, we have planned several studies to fully evaluate MAC's usability and efficacy. We are currently conducting user tests on the interface and hardware and also plan to conduct further studies using a traditional pre- and post-test repeated measures experimental design. In other future work, the system could also be easily adapted to train learners in the acquisition of similarly difficult phonemic contrasts (e.g. Chinese speakers' difficulties in discriminating English /t/-/d/ [29] or Korean speakers' perception of the /v/ and /f/ contrast in English [30], and so on). There would be little change required to adapt MAC for these applications beyond changing the database of speech samples used and any internationalization issues that would arise for a non-English language contrast. Nonetheless, our work so far represents a useful and novel starting point in the field of adaptable systems for speech remediation.

Acknowledgments

This research was supported by the Engineering and Physical Sciences Research Council, U.K. (grant number GR/S55095/01). We thank Drs. Paul Iverson and Valerie Hazan from the Phonetics Department at UCL for the use of their speech sample database. We also thank Prof. Reiko Akahane-Yamada for valuable discussions and suggestions during our visit to ATR labs.

References

[1] G. Aist. *Call Media, design and applications*, chapter Speech recognition in computer-assisted language learning, pages 165–182. Swets & Zeitlinger, 1999.

[2] J.C. Ingram and S-G. Park. Cross-language vowel perception and production by japanese and korean learners of english. *Journal of Phonetics*, 25:343–370, 1997.

[3] J.E. Flege, O.S. Bohn, and S. Jang. Effects of experience on non-native speakers' production and perception of english vowels. *Journal of Phonetics*, 25:437–470, 1997.

[4] J.E. Flege, I.R.A. MacKay, and D. Meador. Native italian speakers' perception and production of english vowels. *Journal of Acoustic Society of America*, 5(106):2973–2987, 1999.

[5] A. R. Bradlow, D. B. Pisoni, R. Akahane-Yamada, and Y. Tokhura. Training japanese listeners to identify english /r/ and /l/: Iv. some effects of perceptual learning on speech production. *Journal of the Acoustical Society of America*, 101(10):2299–2310, April 1997.

[6] A. R. Bradlow, D. B. Pisoni, R. Akahane-Yamada, and Y. Tokhura. Training japanese listeners to identify english /r/ and /l/: Long term retention of learning in perception and production. *Percept Psychophsy.*, 61(5):977–985, July 1999.

[7] J. S. Logan, S. E. Lively, and D. B. Pisoni. Training japanese listeners to identify /r/ and /l/: A first report. *Journal of the Acoustical society of America*, 89(2):874–886, Feb. 1991.

[8] S. E. Lively, J.S. Logan, and D.B. Pisoni. Training japanese listeners to identify english /r/ and /l/: Ii. the role of phonetic environment and talker variability in learning new perceptual categories. *Journal of the Acoustical Society of America*, 94(3):1242–1255, September 1993.

[9] A. Sheldon and W. Strange. The acquisition of /r/ and /l/ by japanese learners of english: Evidence that speech production can precede speech perception. *Applied Psycholinguistics*, 3.

[10] W. Strange and S. Dittman. Effects of discrimination training on the perception of /r-l/ by japanese adults learning english. *Perception and Psychophysics*, 1984.

[11] A. R. Bradlow, D. B. Pisoni, R. Akahane-Yamada, Y. Tokhura, E. Lively, and T. Yamada. Training japanese listeners to identify english /r/ and /l/ iii. long-term retention of new phonetic categories. *Journal of the Acoustical Society of America*, 96(4):2076–2087, October 1994.

[12] B. S. Bloom. The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring. *Educational Researcher*, 13:3–16, 1984.

[13] N. Takagi. The limits of training japanese listeners to identify english /r/ and /l/: Eight case studies. *Journal of Acoustic Society of America*, 111(6):2887–2896, 2002.

[14] J.L. McClelland, J.A. Fiez, and B.D. McCandliss. Teaching the /r/ -/l/ discrimination to japanese adults: behavioral and neural aspects. *Physiology and Behavior*, 77:657–662, 2002.

[15] B.D. McCandliss, J.A. Fiez, A. Protpapas, M. Conwayand, and J.L. McClelland. Success and failure in teaching the /r/ -/l/ contrast to japanese adults: Tests of a hebbian model of plasticity and stabilization in spoken language perception. *Cognitive, Affective and Behavioral Neuroscience*, pages 89–108, 2002.

[16] P. Jusczyk. Developing phonological categories from the speech signal. In *Proceedings of the International conference on Phonological development*, Stanford University, 1989.

[17] P. Kuhl and P. Iverson. Linguistic experience and the 'perceptual magnet effect'. In *Speech Perception and linguistic experience*, pages 121–154.

[18] A.L. Francis and H.C. Nusbaum. Selective attention and the acquisition of new phonetic categories. *Journal of Experimental Psychology: Human Perception and Performance*, 28:349–366, 2002.

[19] Y.S. Chen, T.C. Kao, and J.P. Sheu. A mobile learning system for scaffolding bird watching learning. *Journal of Computer Assisted Learning*, 19(3):347–359, 2003.

[20] R. Oppermann and M. Specht. Adaptive mobile museum guide for information and learning on demand. *HCI (2)*, 1999.

[21] S. Davis. Research to industry. four years of observations in classrooms. *Proceedings of*

IEEE International Workshop on Wireless and Mobile Technologies in Education (WMTE), 99:31–38, August 2002.

[22] A. J. Kok. A review and synthesis of user modelling in intelligent systmes. *The Knowledge Ingineering Review*, 6(1):21–47, 1991.

[23] J. Self. The defining characteristics of intelligent tutoring systems research: Itss care, precisely. *International Journal of Artificial Intelligence in Education*, 10:350–364, 1999.

[24] B. du Boulay and R. Luckin. Introduction to the special issue on modelling human teaching tactics and strategies. *International Journal of Artificial Intelligence in Education*, 12:232–234, 2001.

[25] A. Mabbott and S. Bull. Alternative views on knowledge: presentation of open learner models. In In J. C. Lester and R. M. Vicary, editors, *Proceedings of the 7th International Conference in Intelligent Tutoring Systems, ITS2004*, pages 646–655. Springer-Verlag, Berlin, 2004.

[26] S. Bull. Supporting learning with open learner models. In *Proceedings of the 4th Hellenic Conference with International Participation: Information and Communication Technologies in Education*, Athens, 2004.

[27] J. Kay. Learner know thyself: student models to give learner control and responsibility. In T. Ottomann Z. Halim and Z. Razak, editors, *ICCE'97, International Conference on computers in education*, pages 17–24, 1997.

[28] M. Uther, P. Singh, and J. Uther. Mobile adaptive call (mac): A software for speech remediation. In *Proceedings of IEEE Pervasive services in computing*, July 2005. To appear.

[29] J. E. Flege. Chinese subjects' perception of the word-final english /t/-/d/ contrast: Performance before and after training. *Journal of the Acoustical Society of America*, 86(5):1684–1697, July 1989.

[30] A. Elreyes F. R. Eckman and G. K. Iverson. Some principles of second language phonology. *Second Language Research*, 19(3):169–208, July 2003.