Mining a Database of Reading Mistakes: For What Should an Automated Reading Tutor Listen?

James Fogarty, Laura Dabbish, David Steck, Jack Mostow
School of Computer Science
Carnegie Mellon University, Pittsburgh, PA 15213
{jfogarty, dabbish, dsteck, mostow}@cs.cmu.edu

Abstract: Using a machine learning approach to mine a database of over 70,000 oral reading mistakes transcribed by University of Colorado researchers, we generated 225 rules based on graphophonemic context to predict the frequency of the 71 most common decoding errors in mapping graphemes to phonemes. To evaluate their generality, we tested how well they predicted the frequency of the same decoding errors for different readers on different text. We achieved .473 correlation between predicted and actual frequencies, compared to .350 correlation for context-independent versions of the same rules. These rules may help an automated reading tutor listen better to children reading aloud.

1. INTRODUCTION

This paper addresses the problem of how to predict students' specific oral reading mistakes, called *miscues*. This problem arose in the context of Project LISTEN's automated Reading Tutor [6], which listens to children read aloud, but may also be of interest to the reading research and educational communities. We report on work proposed and supervised by the last author and performed by the first three authors as part of the Fall 2000 offering of Sebastian Thrun's graduate course on Machine Learning.

One reason to predict specific miscues is to improve listening accuracy, which must strike a tradeoff between detecting miscues and falsely rejecting correctly read words, thereby frustrating the student. Previous work has approximated miscues as other words in the text being read, as concatenations of those words [7], or as phonemic truncations of the correct word [8]. This method accepted over 95% of correctly read words, but detected fewer than half the miscues serious enough to threaten comprehension. Moreover, even when miscues are too minor to remediate, modeling them inaccurately may lead the speech recognizer astray. Thus a better model of miscues may not only improve miscue detection, but also reduce false rejection. Manual identification of likely mispronunciations for individual words in a given text [9] is onerous and does not transfer to new text containing other words. Although other researchers have trained acoustic models for children's oral reading [11], we are not aware of previous work to predict specific miscues in oral reading of connected text, or their frequencies.

Another reason to analyze miscues is to identify useful *types* of mistakes to listen for — whether to remediate immediately, tolerate as unimportant, or keep track of over time. *Miscue analysis* attempts to use a child's omissions, substitutions, insertions, and self-corrections to infer information about the child's strategies for reading [3, 4, 12]. This information can be useful to researchers in understanding reading processes and to teachers in remediating them. The ability to detect recurring instances of a particular type of miscue could help the Reading Tutor remediate the miscue. For example, if the student frequently mispronounces the letter '*i*' as the short vowel /IH/ instead of the long vowel /AY/ in words like "bite," the Reading Tutor could explain how silent '*e*' affects preceding vowels.

To address this problem, we applied a machine learning approach to a database of over 70,000 oral reading miscues recorded, transcribed, and annotated by Professor Richard Olson, Helen Datta, and their colleagues as part of a separate multi-year study at the University of Colorado. This database contained miscues by 868 subjects, mostly between the ages of eight and twelve, reading one of seven graded texts, ranging from 296 to 461 words in length. Each text included from two to four of the 22 graded passages in Spache's *Diagnostic Reading Scales* [13], which are written at levels ranging from grade 1 to grade 7. Based on subject reading levels, University of Colorado researchers assigned each subject to passages that he or she would find challenging, made recordings while the subject read, and manually coded miscues from the recordings. For each miscue, they coded the word on which the miscue took place, a phonetic transcription of what the subject said, and information about the type and severity of the miscue.

We mined this corpus to develop a set of rules that predict specific mistakes readers are likely to make, and the frequencies of these mistakes. For example, one rule predicts that in a word where the grapheme sequence '…*re*…' should be pronounced as the phoneme sequence /R EH/, readers will substitute an /IY/ phoneme for the /EH/ phoneme with a frequency of 5.49%, under assumptions we will explain momentarily. To validate our results, we set aside part of our data for testing. We designated data from the grade 2 and grade 6 texts as our test data. There was no overlap in subjects or text between our training and test data.

2. A SUITABLE HYPOTHESIS SPACE

Children identify printed words in various ways. They may use a "decoding" strategy to read a word, i.e., map its successive graphemes to phonemes, perhaps imperfectly. Alternatively, they may try to guess the whole word from its first and last letters, its overall shape, the syntactic and semantic context, and/or even an accompanying illustration. They may guess words that are particularly common, words that have similar patterns of letters, or even just words they know. Furthermore, readers may frequently switch between strategies [12]. To attempt to learn in a hypothesis space that allows for all of these possibilities would probably require substantially more data than we had available. We needed to make assumptions that would reduce the size of our hypothesis space.

Given this data and the goal of predicting the mistakes made by readers, we focused on errors that appeared to be the result of using a decoding strategy. We were interested in when a subject would substitute, insert, or omit a phoneme. We attempted to predict these types of errors, but did not attempt to predict whole-word substitutions or other errors that take place above the level of individual phonemes.

By focusing on errors in decoding graphemes to form pronunciations, we excluded the problem of predicting the words which subjects would be likely to guess. By focusing on individual phonemes, we reduced the problem of trying to predict the misreading of an entire word to the problem of predicting an improper decoding of an individual grapheme. These reductions in the size of our hypothesis space made it feasible to learn some interesting rules from our data. Modeling other reading strategies may be worthwhile as well [1], but lies beyond the scope of this paper.

Table 1 – Basic G to P to P' mappings.

G	*p*	*r*	*e*	*s*	*e*	*n*	*t*	$	$
P	/P/	/R/	/EH/	/Z/	/EH/	/N/	/T/	_	_
P'	/P/	/R/	/IY/	/Z/	/IH/	/N/	/T/	/AX/	/D/

2.1 Data Classes and Features

When analyzing the phonemes spoken using a grapheme decoding approach, it makes sense to look at the mapping from a grapheme, to the correct phoneme, to the phoneme that was actually spoken. Throughout this paper, we refer to this mapping as a G to P to P' mapping. We also refer to portions of this mapping. A G to P mapping, therefore, is a mapping from the grapheme to the correct phoneme. Similarly, a P to P' mapping is a mapping from the correct phoneme to the phoneme that was actually spoken by the subject.

Table 1 shows an actual sequence of G to P to P' mappings for the word "present" as in "birthday present." Note that we use the caret symbol ('^') to represent the beginning of a word, the dollar symbol ('$') to represent the end of a word, and the underscore symbol ('_') to represent a null grapheme or phoneme. In this sequence, the reader substituted different vowel sounds for each occurrence of /EH/ and appended a pair of phonemes to the pronunciation.

The G to P to P' mappings where P does not equal P' represent decoding errors that we would like to be able to predict. These mappings, therefore, are considered positive examples of decoding errors. Consequently, negative examples have P equal to P'.

2.2 Assumption Validity

In order to focus on miscues resulting from a grapheme decoding process, we needed to disregard data that did not appear to be the result of a grapheme decoding process. For example, a subject who mispronounced the word "present" as the word "party" was obviously not using a grapheme decoding approach. We needed to be able to identify and disregard this type of data.

We approached this problem by defining a metric of how closely a subject's mispronunciation matched the correct pronunciation for a word. We expect that a mispronunciation that matches the correct pronunciation reasonably closely was likely produced by a grapheme decoding approach. We have assumed that correct pronunciations, which do not reveal the strategy being used, are the result of a grapheme decoding approach. While we defer the details of our metric until section 3.3, we mention it here because defining such a metric was vital to focusing on data that matched the assumptions underlying the hypothesis space we defined.

Table 2 – Colorado and Sphinx phonetic transcriptions of a 'Rosetta Stone' example.

English	A	whole	joy	was	reaping,	but	they've
Colorado	*	hOl	joy	wuz	rEping	but	THAv
Sphinx	AX	HH+OW+L	JH+OY	W+AA+Z	R+IY+P+IH+NG	B+AH+T	DH+EY+V

English	gone	south,	you	should	fetch	azure	Mike.
Colorado	gon	sowth	U	shood	fech	azher	MIk
Sphinx	G+AO+N	S+AW+TH	Y+UW	SH+UH+D	F+EH+CH	AE+ZH+ER	M+AY+K

3. DATA CONVERSION

To factor each word-level miscue into individual grapheme-to-phoneme mappings, we needed to know the correspondence between the graphemes in the word and the phonemes in the correct and incorrect pronunciations. The University of Colorado database did not provide this correspondence. Project LISTEN members Greg Aist and Becky Kennedy had already computed these correspondences for the correct pronunciations of all but 49 of the 881 distinct words in the Spache passages used in the Colorado database, so we added the rest by hand. However, these correspondences were expressed in the phoneme set of the Sphinx pronunciation dictionary [14], rather than in the phonemic notation used to transcribe the miscues in the database.

Converting our data to a set of G to P to P' mappings therefore required three steps. First we translated the miscues from University of Colorado phonetic notation to Sphinx notation. Then we aligned each translated miscue against the correct pronunciation of the word to compute its P to P' mappings. Finally, we used the graphophonemic correspondence for the correct pronunciation to factor each miscue into individual G to P to P' mappings, one for each grapheme in the word. Aligning the pronunciations also allowed us to compute a pronunciation similarity metric that we used to determine whether to consider a miscue to be the result of a grapheme decoding approach.

3.1 Miscue Conversion

We converted University of Colorado pronunciations into Sphinx pronunciations with a simple pattern matching process. Table 2 illustrates the notations. To handle conversions involving the flap[1] and schwa[2] phonemes, as well as minor differences in pronunciations, we defined ten Sphinx phoneme pairs to be approximately equal. Five of the ten pairs defined the schwa phoneme to be approximately equal to the vowel phonemes /AH/, /EH/, /IH/, /OW/, and /UH/. Two of the pairs defined the flap phoneme to be approximately equal to the /D/ and /T/ phonemes. The remaining three pairs defined approximate equality between /AXR/ and /ER/, between /AA/ and /AO/, and between /AO/ and /OW/.

3.2 Pronunciation Alignment

With the miscue in Sphinx notation, we needed to align it with the correct pronunciation of the word. This allowed us to determine what parts of the pronunciation were problematic

[1] The flap phoneme represents the sound associated with the '*d*' in "faded" and with the '*t*' in "fated."

[2] The schwa phoneme represents a reduced vowel. It is used when it is impossible to distinguish a particular vowel sound, as in the vowel sounds near the ends of the words "wagon", "engine", and "cradle."

for the subject. We used a dynamic programming algorithm, derived from a solution to the longest common substring problem [2]. The algorithm considered possible alignments of the phonemes sequences, awarded points based on the quality of the match in each phoneme pair of a potential alignment, and selected the alignment with the best score. The algorithm awarded five points if the phonemes of a pair were identical, three points if they were approximately equal, and a single point if they were both vowels or were both consonants. These point values were selected arbitrarily, but comparison of the resulting alignments to manual alignments indicated that these values worked well.

Table 3 – Example miscues on the word "present," and their alignment scores.

present
/P+R+EH+Z+EH+N+T/

#	Pronunciation	Score
(1)	/ P+AA+R+D+IY/	1.50
(2)	/ P+R+EH+S/	2.29
(3)	/ P+R+IY+S+EH+N+T+AX+D/	3.00
(4)	/ P+R+EH+Z+AX+D+IH+N+T/	3.67
(5)	/ P+R+IY+Z+EH+N+T/	4.43

3.3 Measuring Alignment Quality

Given the method we used for aligning pronunciations, we defined our metric of alignment quality to be the score of the alignment divided by the number of graphemes in the aligned pronunciation. Alignment quality scores, therefore, had values ranging from zero to five.

Defining our metric in this way gave us a reasonable measure of whether a subject was using a grapheme decoding approach. If the subject's pronunciation was very close to the correct pronunciation, the alignment yielded a score close to five. If the subject's pronunciation could not be reconciled with the correct pronunciation, the alignment yielded a score close to zero. This metric also allowed subjects to make more pronunciation mistakes in longer words. It seems reasonable that subjects using a grapheme decoding approach would have made more decoding errors as words became longer.

Table 3 contains actual miscues for the word "present," together with the scores that the pronunciation alignments for these miscues received. Example (1) appears to be the result of a subject guessing the word "party." This pronunciation does not align well with the pronunciation for "present", and so it received a low score. Example (3), with a score of three, seems to be the result of a grapheme decoding process. The child substituted a vowel and appended a pair of phonemes, but the pronunciation is a reasonably close match. We include Example (4) to illustrate a shortcoming of our metric. While this example did align well with the correct pronunciation, this subject may have guessed the word "president," and so may not have been using a grapheme decoding approach. Example (5), an exact match except for a single vowel substitution, received a score of (5+5+1+5+5+5+5) / 7 = 4.43.

For the remainder of this work, we filtered our positive examples to those G to P to P' mappings derived from miscues whose alignment score was at least 2.5. Figure 1 shows a histogram of miscue alignment scores, grouped in bins of size 0.5. By filtering out miscues with alignment scores less than 2.5, we removed about half the miscues. This was necessary, however, to avoid interference from data that could not be reconciled with a grapheme decoding process.

4. INITIAL RESULTS

With our data in the form of a set of G to P to P' mappings, we obtained some initial results. We computed the most common positive G to P to P' examples in our training data. These initial results gave us a sense of what mappings we should continue to investigate. We also conducted some preliminary tests to see if these results generalized to an independent test set. These initial tests indicated that it is likely we would be able to predict errors in an independent test set.

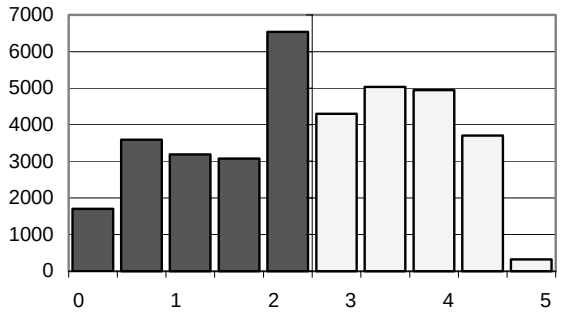

Figure 1 – Histogram of training data miscues, binned by alignment score. The bars on the right are above our threshold value of 2.5.

Table 4 – The ten most common training G to P to P', with example words from the training data.

G	*P*	*P'*	*#*	***Example***
s	/S/	_	1164	plants
s	/Z/	_	746	arms
_	_	/N/	735	ha_d
$	_	/Z/	724	car_
n	/N/	_	670	land
t	/T/	_	626	went
_	_	/R/	517	c_ook
r	/R/	_	476	trip
e	/EH/	_	475	quiet
_	_	/AX/	442	eng_lish

4.1 Common G to P to P' Mappings

After converting subject miscues, aligning them to the correct pronunciation, and filtering out the miscues with an alignment score below 2.5, our training data yielded 33,713 instances of 1807 distinct G to P to P' mappings. Luckily for this work, 1346 of those mappings occurred less than ten times each and accounted for a total of less than 12% of the training instances. We did not try to predict these errors, as any one of them occurred far too infrequently for the data to have provided us with enough information about its causes. We instead focused our attention on the seventy-one mappings that occurred at least one hundred times. This threshold is arbitrary, but the resulting mappings accounted for more than 53% of mapping instances.

Table 4 lists the ten most common mappings in our training data, all ten of which turned out to be insertions or deletions. Thus, the most common error was to ignore a letter '*s*,' thereby deleting the phoneme /S/ – such as the plural ending of the word "plants." The figure shows the number of times each mapping occurred.

4.2 Mappings in an Independent Test Set

Given the common G to P to P' mappings in our training data, it was appropriate to wonder if these mappings would also be common in an independent data set. Computing the G to P to P' mappings in our test data yielded 13,722 instances of 1085 distinct G to P to P' mappings. As should be expected, the mappings were not identical to those from the training set. Substantial overlap can be seen, however, by looking at the frequency in the test data of the seventy-one mappings in the training data on which we focused our work. This data is plotted in Figure 2. The chart contains 142 points. The seventy-one points that form a

smooth curve are the seventy-one most common mappings in the training data and are sorted by what percent of the total number of miscue mappings they represent. Each of these points is associated with a point on the same vertical line that represents the percent of miscue mappings the same error represents in the test data. The correlation coefficient is .582.

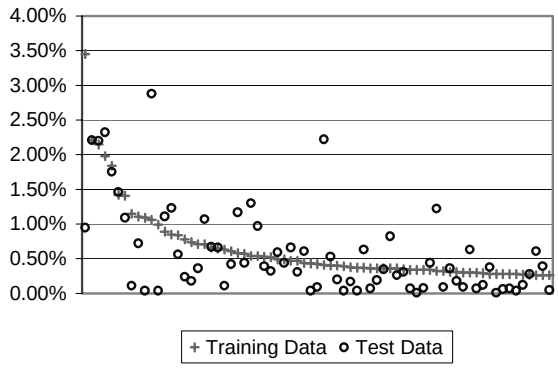

Figure 2 – The percent of total miscues in the training and test data of the seventy-one miscue mappings that are most common in the training data, ordered on the x-axis by frequency in the training data.

It seems reasonable, therefore, to use the frequency of mappings in our training data to predict the miscue mappings that will occur in an independent data set. It was important to verify this point, because the rest of our work required that the frequency of decoding errors in the training set generalize to other readers and other text.

5. MISCUE CONTEXT ANALYSIS

We decided to examine the contexts of decoding errors to see when particular decoding errors occur with a noticeably higher or lower frequency. By determining the contexts in which a particular miscue mapping is either more or less likely, we can improve the accuracy of our predictions for these contexts. It is important for an automated reading tutor to be given the best predictions possible, as the imperfections in speech recognizer technology assure that a speech recognizer will sometimes hallucinate the errors it is told are likely.

In order to find contexts that make a particular miscue mapping more or less likely, we applied a decision tree to a set of positive and negative mapping examples. The context of each mapping defined a set of features for that mapping. While our positive mapping examples were explicit in the University of Colorado data, our negative examples were implicit, and so we had to generate them. We then adjusted our data to compensate for the fact that we were using a classifier[3] to do density estimation[4]. Finally, we used a modified version of the C4.5 decision tree package [10] to discover which graphophonemic contexts had a substantial impact on the frequency of particular decoding errors.

5.1 Defining Our Feature Set

We defined the features of a G to P to P' mapping to be the graphemes, phonemes, and grapheme to phoneme mappings that appeared adjacent to the G to P to P' mapping in the word. Considering graphemes as context is a natural result of our assumption of a grapheme

[3] A classifier uses the features of objects to attempt to learn the classes to which data items belong. In our case, a classifier attempts to classify the value of P' spoken by the subject as correct or incorrect.

[4] Density estimation is the identification of locations in space where something occurs with high or low frequency. In our case, the context of G to P to P' mappings define the space and we are looking for high or low frequencies of positive examples.

decoding process. We also considered phonemes as context because it seems reasonable that we could find an influence by a variety of graphemes only when they map to a particular phoneme. Because we are making predictions of reading mistakes, we did not consider the value of P' in nearby mappings to be a feature. We do not have prior knowledge of these values in our planned application of decoding error prediction.

Ideally, we would like to be able to consider all of the graphemes, phonemes, and grapheme to phoneme mappings in a word. We might even want to look beyond the current word, to see if the previous word has an influence on subject miscues. Because our data set does not contain enough distinct words to allow this approach, we limited the feature set of a mapping to the values of G, of P, and of the G to P mapping in the locations directly on either side of the mapping. Considering even the two locations on either side of a mapping resulted in a number of rules that could be applied to only a single word in our training data.

5.2 Defining Our Training Data

The University of Colorado data set explicitly codes the miscues made by subjects, but not the words they read correctly. Using a decision tree requires that we have both positive and negative training examples. We derived our positive examples directly from the miscue data, but needed to infer our negative examples. We generated our negative examples by assuming that the absence of an error for a word indicated that the word was read correctly. For each subject, we examined each instance of a G to P mapping in the passage read by that subject. For each instance, we checked the errors made by the subject to see if an error occurred on that instance. If we did not find an error, we created a negative training example. If we did find an error, we examined the alignment score for the miscue containing the error. As discussed earlier, we ignored the error if the alignment score was less than 2.5. In this case, we created neither a positive example nor a negative example, as the subject was probably not using a grapheme decoding approach. If the alignment score was at least 2.5, we created a positive training example.

The process described gave us positive and negative examples that portrayed the contexts in which subjects using a grapheme decoding approach had trouble with a particular mapping. However, the resulting data was not yet appropriate for the application of a decision tree classifier, because we needed the classifier to do density estimation. To clarify the problem, consider the training examples that resulted from applying the described process to the miscue mapping from '*e*' to /EH/ to /IY/. The resulting training examples showed that, of the 23,514 instances of a mapping from '*e*' to /EH/ apparently produced by a grapheme decoding approach, /IY/ was substituted for /EH/ in 229 instances, a frequency of 0.97%. The entropy[5] of the data set, therefore, was low. If we considered the subset of this data where the '*e*' to /EH/ mapping is preceded by the grapheme '*r*' mapped to the phoneme /R/, we found that the substitution occurred in 85 of 1547 instances, a frequency of 5.49%. This was exactly the type of contextual influence that we wanted to identify. However, the entropy of the subset of the data where this rule applies was higher than that of the original data set. Decision tree learners try to minimize the entropy of a data set. If C4.5 had identified this subset of the data, it would have increased entropy.

---

[5] Entropy is a measure of the homogeneity of a set of data. A data set that is very homogenous will have low entropy. A data set that is very heterogeneous will have high entropy. For a discussion of entropy and decision trees, see [5].

We remedied this problem by duplicating our positive examples until they represented approximately half of our training examples. In the case of the '*e*' to /EH/ to /IY/ mapping, we used 103 instances of each of our 85 positive examples, for a total of 23,857 positive examples and 23,285 negative examples. The resulting data set had a positive example frequency of 50.32% and very high entropy. Because we duplicated only our positive examples, the subset of the mappings preceded by the grapheme '*r*' mapped to the phoneme /R/ consisted of 8755 positive instances and 1462 negative instances. This subset had a positive example frequency of 85.69%, and consequently much lower entropy. This change allowed C4.5 to work properly, and we recovered the actual miscue frequency by compensating for the ratio of positive example duplication after C4.5 identified meaningful contexts.

6. RESULTS AND EVALUATION

Application of the decision tree algorithm to the seventy-one mappings discussed in section 4.1 yielded 11,697 rules. We identified positive rules of interest and evaluated the performance of these rules on our independent test set. Similarly, we evaluated negative rules of interest.

6.1 Positive Rules

A positive rule, which indicates that a particular miscue mapping is more likely in a given context, was obtained from each positive node in the generated trees. In these cases, the frequency of the miscue mapping in the given context is greater than the frequency of the miscue mapping in the general case, which we shall refer to as the base frequency of the miscue mapping. Because we wanted our predictions to have a reasonable likelihood of generalizing to new words, we identified two other criteria for a rule to be considered significant. First, the rule needed to apply to at least three distinct words in the training data. Second, the rule needed to define a context in which a miscue occurred with frequency of at least one percent in the training examples. These criteria were met by 225 positive rules. Table 5 lists the fifty rules with the highest predicted miscue frequency, with words in the training data that were matched by the rules. For example, the first rule in the right column predicts substitution of the /IY/ in a mapping from '*e*' to /EH/ in contexts of the form '…*re*…' where '*re*' should be pronounced as /R EH/. The predicted miscue frequency for these examples ranges from 16.5% to 3.8%.

We evaluated each of the 225 rules by generating test examples in the same way that we generated training examples, except that we used the independent training data described in section 1. Of the 225 rules, 157 matched words in the test data. To determine the predicted number of errors, we multiplied the predicted miscue frequency for these 157 rules by the total number of test examples matched by the rules. We then compared this prediction to the actual number of miscues in the examples matched by the rules. The results, sorted by the predicted number of miscues, are given in Figure 3. The correlation coefficient is .473.

Table 5 – The top fifty positive rules and examples of words in the training data to which they apply. The rules are listed in descending order by predicted miscue frequency.

Left Context	*Error*	*Right Context*	*Example Word*
u /AH/	*n* /N to _/		unusual
p /P/	*o* /AA to AH/		poppies
c /K/	*o* /AA to AH/		competent
t /T/	*s* /S to _/		plants
m /M/	*i* /IH to AY/		mint
m /M/	*e* /EH to AE/		men
sh /SH/	*i* /IH to EH/		shillings
p /P/	*e* /EH to _/		competent
e /EH/	*s* /Z to _/	$	produces
t /T/	*e* /EH to _/	*d*	expected
c /S/	*e* /EH to _/		produces
a /AX/	*n* /N to _/	*d*	England
or	_ /_ to AH/	*m*	for m
n /AX/	*n* /N to _/		didn't
t	_ /_ to R/	/AH/	t usks
e /IH/	*t* /T to _/		pretended
m /M/	*a* /AX to AE/		material
n /N/	*i* /IH to _/		colonists
ee	_ /_ to Z/	$	tree
ure /AXR/	_ /_ to Z/	$	picture
/K/	*o* /AA to _/		colonists
or /ER/	_ /_ to R/		recor d
i /IH/	*s* /S to _/		colonists
	s /S to SH/	*o* /AX/	mason
c /K/	*r* /R to _/		crate

Left Context	*Error*	*Right Context*	*Example Word*
r /R/	*e* /EH to IY/		present
d /D/	*s* /Z to _/		backwoodsman
t /T/	*e* /EH to _/		eaten
m /M/	*s* /Z to _/		arms
^	*sh* /SH to S/	/IH/	shillings
d /D/	*s* /Z to _/	$	birds
d /DX/	*e* /EH to _/		proceeded
ll /L/	*y* /IY to _/		woolly
e /EH/	*d* /D to _/		visited
e /EH/	*s* /Z to _/		present
	sh /SH to S/	/IH/	shillings
w	*a* /AA to EH/		want
/P/	*o* /AA to OW/		poppies
	t /T to D/	*o* /AX/	mastodon
ure	_ /_ to Z/	$	picture
t /T/	*e* /EH to IH/	*d*	expected
n	*ed* /D to _/		cleaned
t	*o* /OW to AH/		tobacco
	c /K to _/	*o* /AX/	decoration
	l /L to _/	*u* /AH/	bluff
/AXR/	_ /_ to S/	*d*	northwar d
d /D/	*e* /EH to _/		pretended
ie /IY/	*s* /Z to S/		duties
s /S/	*t* /T to _/		mastodon
	c /K to _/	*r* /R/	democratic

To determine if it is worthwhile for an automated reading tutor to consider context, we compared these results to those obtained by considering only the base frequency of the miscue mappings. For the same test examples used to evaluate the context rules, the values had a correlation coefficient of .350 — indicating that rules based on graphophonemic context provide a better prediction than rules based only on base miscue frequency.

6.2 Negative Rules

A negative rule, which indicates that an error is *less* likely in a given context, was obtained from each negative node in the generated trees. Negative rules are of interest because they indicate situations in which it is very unlikely for a subject to make an otherwise common decoding error. An automated reading tutor could tell the speech recognizer not to listen for a mistake in a particular context, thus reducing the number of errors hallucinated by the speech recognizer.

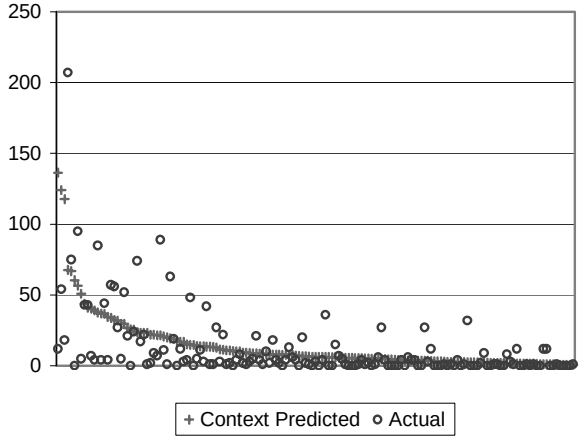

Figure 3 – Comparison for positive rules of the context prediction and the actual number of errors in test data, ordered on the x-axis by the predicted number.

Table 6 – The top twenty negative rules and examples of words in the training data to which they apply. The rules are listed in descending order by the base frequency of the miscue.

Left Context	*Error*	*Right Context*	*Example Word*	*Left Context*	*Error*	*Right Context*	*Example Word*
u /AH/	*d* /DX to _/		study	*i* /IH/	*s* /Z to _/	$	his
	o /AX to _/	/M/	from		*a* /AA to AE/	*n*	want
^	*s* /S to _/	*l* /L/	slow	*l*	*a* /AA to AE/	*r*	large
^	*s* /S to _/	*e* /EH/	settle	*wh* /W/	*e* /EH to _/		when
^	*s* /S to _/	*ee* /IY/	seed	*h* /HH/	*e* /EH to _/		helped
t /T/	*o* /AA to AH/		stop	*g* /G/	*e* /EH to _/		get
/P/	*o* /AA to _/		upon	*th* /DH/	*e* /EH to _/		there
/B/	*o* /AA to _/		body	*w* /W/	*e* /EH to _/		west
^	*o* /AA to _/		objects	*m* /M/	*e* /EH to _/		men
/T/	*o* /AA to _/		stop	*l* /L/	*e* /EH to _/	*ph*	elephants

We evaluated the 107 negative rules that apply to at least two words in the training data, predict the error will not be made, and define a context for a miscue mapping with a base frequency greater than one percent. Of these rules, 69 matched words in the test data. The average absolute difference between the predicted and actual number of errors in the test examples matched by the rules was 1.4 errors for the context rule predictions (standard deviation of 3.2), compared to 7.4 errors for the context-independent predictions (standard deviation of 13.7). We change criterion here due to zero variance in the context predictions.

7. CONCLUSION

Our results indicate that we can successfully predict graphophonemic contexts in which some oral reading miscues are particularly likely or particularly unlikely. Our results also indicate that the predictions that consider context are better than those that consider only base frequency of a miscue.

This work derived a new representation from the University of Colorado data. The new representation enabled us to conduct analyses that were not feasible with the data in its original form. Representing miscues as sets of G to P to P' mappings, filtering out the miscues that did not appear to be the result of a decoding approach, and examining the features of mappings allowed us to discover new information about the miscues. Further analysis of the data in this factored representation may lead to additional discoveries.

The work reported here did not modify the Reading Tutor itself. Now that we have learned rules to predict high-frequency errors, we have started off-line experiments to try listening for them. One goal is to increase the Reading Tutor's detection of mistakes serious enough to warrant intervention. Another goal is to reduce its false alarm rate by deliberately ignoring minor mistakes. A further step in this direction would be to split up the miscues based on whether they are coded in the Colorado data as minor, so as to train one set of rules for minor miscues to ignore, and a separate set of rules for more serious miscues.

Knowledge about which reading mistakes occur frequently is useful in another way too: for deciding on what to focus tutoring. Now that we have identified some frequent mistakes, we can design new instructional activities to remediate or prevent them. Moreover, given knowledge of which graphemes are misread in which contexts, we can identify *correct* context-specific mappings that apply frequently and reliably in those contexts. We can then

target those mappings as instructional objectives. Teaching such mappings does not necessarily mean teaching them as abstract rules, which young children are unlikely to understand. Instead, we are designing instructional activities to teach such mappings by presenting examples, and helping children induce the rules by calling attention to the context.

Future work will show whether the learned rules indeed enable Project LISTEN's Reading Tutor to listen more accurately and informatively to children's oral reading. The learned rules may be of interest in their own right to reading researchers, and to the AI and education community as an interesting example of educational data mining.

ACKNOWLEDGMENTS

We thank Richard Olson, Helen Datta, and Jacqueline Hulslander for the University of Colorado miscues database, whose development was supported in part by the National Institute of Health (grants HD-11681 and HD-27801) and the National Institute of Child Health and Development (grant ROI HD-22223).

We thank the reviewers for their comments. We thank the other members of Project LISTEN, especially Rachel Gockley for designing a readable notation for the learned rules. We thank Alan Black for the example sentence in Table 2. We thank John Langford for assistance with machine learning aspects of this work.

This work was supported in part by the National Science Foundation under Grants IERI REC-9979894, IIS-9800597, and IIS-9817527, and by the first author's National Science Foundation Graduate Research Fellowship. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government.

REFERENCES (see also http://www.cs.cmu.edu/~listen)

[1] Cassidy, S. (1990) Substitution errors in a computer model of early reading. *Proceedings of the First Conference of the Australasian Society for Cognitive Science*. Sydney, Australia.

[2] Cormen, T.H., Leiserson C.E., and Rivest, R.L. (1990). *Introduction to Algorithms*. NY: McGraw-Hill.

[3] Goodman, Y.M., and Burke, C.L. (1972). *Reading Miscue Inventory: Manual and procedures for diagnosis and evaluation*. New York: MacMillan.

[4] Hempenstall, K. (1999). Miscue analysis: A critique. *Effective School Practices*, 17(3), 85-93.

[5] Mitchell, T. (1997). *Machine Learning*. Boston: McGraw-Hill.

[6] Mostow, J. and Aist, G. (In Press, 2001). Evaluating tutors that listen: An overview of Project LISTEN. In K. Forbus and P. Feltovich (Eds.) *Smart Machines in Education: The coming revolution in educational technology*. MIT/AAAI.

[7] Mostow, J., Hauptmann, A., Chase, L.L., Roth, S. (1993). Towards a reading coach that listens: Automated detection of oral reading errors. *Proceedings of AAAI 93*, 392-399. Menlo Park, CA.: AAAI Press.

[8] Mostow, J., Roth, S., Hauptmann, A., Kane, M. (1994). A prototype reading coach that listens. *Proceedings of AAAI 94*, 785-792. Menlo Park, CA.: AAAI Press.

[9] Nix, D., Fairweather, P., and Adams, B. (1998). Speech recognition, children, and reading. *Proceedings of CHI 98*, 245-246.

[10] Quinlan, J.R. (1993). *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufmann.

[11] Russell, M., Brown, C., Skilling, A., Series, R., Wallace, J., Bohnam, B., and Barker, P. (1996). Applications of automatic speech recognition to speech and language development in young children. *Proceedings of ICSLP 96*, 176-179.

[12] Snow, C.E., Burns, M.S., and Griffin, P. (1998). *Preventing Reading Difficulties in Young Children*. Washington, D.C.: National Academy Press.

[13] Spache, G.D. (1981). *Diagnostic Reading Scales*. Monterey, CA: McGraw-Hill.

[14] The CMU Pronouncing Dictionary. http://www.speech.cs.cmu.edu/cgi-bin/cmudict