PREDICTING ORAL READING MISCUES

Jack Mostow, Joseph Beck, S. Vanessa Winter, Shaojun Wang, and Brian Tobin

Project LISTEN (http://www.cs.cmu.edu/~listen)
Robotics Institute, Carnegie Mellon University
RI-NSH 4213, 5000 Forbes Avenue, Pittsburgh, PA 15213-3890

ABSTRACT

This paper explores the problem of predicting specific reading mistakes, called *miscues*, on a given word. Characterizing likely miscues tells an automated reading tutor what to anticipate, detect, and remediate. As training and test data, we use a database of over 100,000 miscues transcribed by University of Colorado researchers. We explore approaches that exploit different sources of predictive power: the uneven distribution of words in text, and the fact that most miscues are real words. We compare the approaches' ability to predict miscues of other readers on other text. A simple rote method does best on the most frequent 100 words of English, while an extrapolative method for predicting real-word miscues performs well on less frequent words, including words not in the training data.

1. INTRODUCTION

This paper addresses the problem of how to predict oral reading mistakes, called *miscues*. The ability to predict likely student mistakes is valuable in understanding, detecting, and remediating student difficulties [1]. Our objective is to characterize oral reading behavior statistically, and to generate models to help Project LISTEN's computer Reading Tutor [4] listen more accurately for miscues. This work may also be of interest to researchers and educators. For example, reading researchers and practitioners have used miscue analysis to infer children's reading strategies [3]. [2] discusses additional motivation for the problem of predicting miscues.

Our goal is to identify a small set of likely miscues to add to the Reading Tutor's language model. A tutor that listened for every possible phoneme sequence in place of a correct word would hallucinate too many miscues, given the limited accuracy of current speech recognition technology. In general, any computer tutor that tracks student behavior suffers from this problem of combinatorics: the more possible student paths the tutor has to consider, the more processing power required and the less certain the tutor can be.

Miscues include repetitions, insertions, substitutions, omissions, and hesitations. We are interested here in predicting insertions and substitutions – that is, sounds that a disfluent reader is likely to produce other than words in the text at hand. [6] manually identified children's miscues on specific words in a given text. [5] approximated miscues as concatenations of other words in the text, for example "elephant" as "and of that," and also predicting phonemic truncations of the correct word. For example, the word "reading" can be truncated to /r/, /r ee/, /r ee d/, /r ee d ih/, or /r ee d ih n/. This model predicts dropped word endings and some false starts, but not other miscues. When used in a speech recognizer, it detected only about half of the miscues serious enough to threaten comprehension, with false alarms on approximately 4% of correctly read words. The problem of false alarms precludes simply modeling miscues as arbitrary phoneme sequences.

2. MISCUE DATABASE

The Colorado database contains over 100,000 oral reading miscues recorded, transcribed, and annotated by Professor Richard Olson's team at the University of Colorado. The reading material consists of seven graded text passages from [7], ranging from second to eighth grade in difficulty and 296 to 461 words in length, with a vocabulary of 881 distinct words. The Colorado database encodes descriptive information about each text token, including its passage, location on a computer display, spelling, and pronunciation.

Most of the 868 students were between eight and twelve years old. Each student read one passage, selected to be challenging for that student's reading level. Human coders listened to the recorded oral reading. For each miscue, they coded the word on which the miscue took place, transcribed the miscue phonetically, and categorized its type and severity. The Colorado database represents miscues phonetically rather than orthographically. Translating the transcribed pronunciation to the spelling of the actual word is a surprisingly thorny problem, as we shall see later. But first we describe an approach that does not require such translation.

3. ROTE PREDICTION OF MISCUES

A baseline "rote" approach to training a miscue predictor is simply to enumerate specific actual miscues on particular words, and predict that they will recur. Given enough training data, this naive method will achieve high predictive coverage, without predicting any miscues that never occur. Of course even the Colorado database is not large enough to approach this asymptotic behavior for most of the 669 words in the training vocabulary – let alone for other words, for which it can make no predictions at all. However, the rote approach exploits the uneven distribution of words in natural text: high-frequency words have enough examples of miscues in the Colorado database to cover a surprisingly large amount of test data. Another reason to try the rote approach is as a baseline against which to compare more complex methods.

To apply and evaluate the rote approach, we split the Colorado data into training and test sets, just as in [2]: we held out the third and seventh grade passages as test data, and used the rest as training data. Because each subject read only a single text passage, there was no overlap in subjects or text passages between our training and test data.

We now quantify the overlap in words and miscues between training and test sets. For both words and miscues, we distinguish between types and tokens. A word token is an instance of a word type. For example, in the sentence "The dog hid in the shed," there are 6 word tokens but only 5 word types, because "the" occurs twice.

The five training passages consist of 1,849 word tokens, with a vocabulary of 669 distinct word types. Our training data consists of miscues with phonetic transcriptions, mostly substitution errors. All the words were misread at least once, with 49,848 transcribed miscues of 22,927 distinct types.

The two test passages consist of 875 word tokens and 364 word types. Although only 163 of the 364 word types in the test data occur in the training passages, they cover 594 of the 875 word tokens in the test passages, because high-frequency words comprise much of the text.

We expect a rote model to do well on well-trained miscues – that is, words with plenty of training examples. When we plotted coverage against word frequency, we found that the rote approach did best on the 100 most frequent words of English, 89 of which occurred in the training data. There were diminishing returns on the rest of English, only 581 words of which occurred in the training data.

One problem with the rote model is that it predicted an average of 34.3 possible distinct miscues for each word. Our experience with Project LISTEN's Reading Tutor suggested that listening for so many miscues would raise too many false alarms. We noticed that some miscues occurred more frequently than others. To improve precision without overly harming coverage, we decided to limit predictions to miscues produced by more than one student. We reasoned that such "popular" miscues would be much less likely to be idiosyncratic to a particular student, and hence much likelier to show up in a test set of miscues by different students.

The resulting method performed as follows. Overall coverage was 6.1% of miscue types and 22% of miscue tokens. The pruned rote model made fewer miscue predictions for each target word type – only 7.4, versus 34.3 for the unpruned model, thanks to ignoring idiosyncratic miscues in the training set. However, recall of the pruned model was similar to that of the unpruned model (22% vs. 26% for token recall).

4,640 of the miscue tokens in the test set occurred on the 100 most frequent words of English. For these words, miscue type coverage was 34% and miscue token coverage was 67%, averaging 11.5 predictions per word type. Thus the rote approach performs respectably on common words.

10,973 of the miscue tokens in the test set occurred on less frequent (word rank > 100) words. For these words, miscue type coverage was 2.2% and miscue token coverage was 8%, with an average of 6.2 predictions per word type.

Even though rarer words are known to account for many fewer text tokens, they account for the bulk of the miscue tokens because they are so much harder for students learning to read. How can we predict miscues better on those words?

4. EXTRAPOLATING MISCUES

Most substitution errors in oral reading are themselves real words: approximately 45% of the miscues in the Colorado database are real words, 30% have no transcription (for example, omissions), and only 25% are non-words. By focusing on real-word miscues, we can move miscue prediction from pronunciation space to word space, where we can exploit systematic regularities in the relation of miscues to target words. For example, we might use dictionary knowledge to predict miscues with a similar spelling as the target word, a similar pronunciation, the same root, or a related meaning.

4.1. Data preparation

The University of Colorado database provided only the phonetic transcription of each miscue. To identify the word, we had to find it in a pronunciation dictionary. The Carnegie Mellon Speech Group has a large pronunciation dictionary (http://www.speech.cs.cmu.edu/cgi-bin/cmudict) but it uses a different notation based on the phonemes used in the Sphinx speech recognizer. We first used the converter described in [2] to translate miscues from Colorado notation into Sphinx.

The next step was to find the corresponding word. This task was not as trivial as finding an exact match. Only 40% of the phonetically transcribed miscues matched the Sphinx dictionary pronunciation perfectly. The right word might be in the dictionary but its pronunciation might not match exactly, whether because of variations in defined pronunciations, or because the student pronounced the word differently. In the example above, the transcribed miscue /wut/, translated into the Sphinx phonemes /W AH T/, might still not match the reference pronunciation /HH W AH T/.

To solve this problem, we looked for the best phonemic match instead of a perfect match. To compute match distance, we used a modified version of the Levenshtein edit distance algorithm, with different weights to penalize or tolerate substitutions, insertions, and deletions. The algorithm assigned a 0- or 2-point penalty for substituting similar phonemes, and a 5-point penalty for non-similar phonemes. We then normalized the total penalty by dividing by the number of phonemes in the pronunciation we were trying to match.

Computing the Levenshtein distance between every miscue and every word in the dictionary took longer than we had. To speed up the search, we exploited the fact that most miscues start with the same letter as the target, and only considered such words as possible matches.

We needed to distinguish between good and bad matches, for several reasons. One reason was the same-first-letter heuristic, because it excluded proper matches for those real-word miscues that started with a different letter, finding bogus matches instead. Another reason was that our dictionary lacked some of the miscue words, especially inflections.

There were a total of 45,503 miscues labeled as real-word miscues, of which we used 33,491 for training data and 12,012 for test data. We found that a threshold of 1.0 on normalized match penalty excluded bogus matches fairly well. This threshold eliminated as training examples 40% of the miscues marked in the Colorado data as real words. Allowing poorer matches would have increased the percentage of mislabeled training data. For example, one miscue on the word "SUNSHINY" was transcribed as /shoo' shing/, which translates to "shushing" (asking someone to be quiet). Our dictionary did not have this word, and was forced to map it to "SCHWING." This match scored 2.0, too bad to include as a training example.

4.2. Predictive features of real-word miscues

How can we learn to predict real-word miscues from a database of miscues on only a few hundred words of text? That is, how can we generalize to predict miscues on words for which we have sparse data or none? To address this problem, we abstracted from specific miscues to features that might generalize. For brevity we omit here other features that we considered but did not try.

A real-word miscue involves a particular student misreading a target word as some other word. We therefore looked for features of the student, the target word, the miscue word, and relations among them. We had no explicit information about the students. However, passages were assigned based on student reading levels. We therefore used passage level as a proxy for student reading level.

We expected that the miscue would resemble the target word in one or more ways, which we encoded in terms of the following features. To quantify similarity in spelling, we computed their edit distance, the difference between their lengths, and the absolute value of that difference. To help check for dropped and added plurals, we added a feature that was true when one word ended in S and the other did not.

To quantify similarity of pronunciation, we computed the edit distance between phoneme sequences, using the same metric described above for matching transcribed miscues to dictionary entries. Miscues usually have the correct first phoneme and often have the correct last phoneme, so we encoded features for matches on the first and last phonemes.

We expected that students would have more trouble with rarer words, and would be likelier to know – and therefore guess – more frequent words. We therefore encoded the frequency rank of both target and miscue. (The Nth most frequent word has frequency rank N.) We used a word frequency table (generated by Project LISTEN member Greg Aist) of 25,000 words in a corpus of children's stories. This table covered all but 55 of the miscue word types.

4.3. Generalizing from the features

We tried various methods to predict miscues from the features. Our initial explorations plotted the distributions of feature values for the miscues in the training set. One finding was that "big kids make little mistakes, and vice versa." That is, normalized edit distance between target word and miscue was larger on lower passages.

We adopted a classifier learning approach to distinguish probable miscues. Given a target word, we would then use the classifier to predict which words in the dictionary were likely to occur as miscues for the target.

For this approach, we needed not only positive training examples of real-word miscues, but negative examples as well – words that were not produced as miscues. We chose the negative training data from a region containing most of the miscues, as follows. For each target word, we selected from the dictionary all words that started with the same first letter as the target word, were within edit distance of 3 or less, and normalized pronunciation distance of 3.75. These words, minus the actual miscues in the training data, comprised the negative training examples for each target word. We wanted to train the classifier to distinguish miscues from real words within this region, and to avoid swamping the learning procedure with negative training examples. As it was, we had ten times as many negative training examples as positive ones.

To train a classifier, we wanted to start simple and fast, so we used linear discriminant as implemented in SPSS. This method took less than a minute on a training set of 341,224 examples. Table 1 describes the standardized coefficients of the linear discriminant, listed from most to least predictive. The lower the output of the classifier, the more likely the word is to be a real student miscue.

EDITDIST	0.675
FMATCH	0.602
PHONAVER	0.447
PHONDIST	0.364
FTARGET	-0.303
GRADEGRO	-0.046
FIRSTPHO	0.045
GRADE	-0.024
LASTPHON	-0.007

Table 1. Coefficients for linear discriminant

The first feature is the edit distance between target and miscue: miscues tend to be spelled like the target. The second feature is the frequency rank of the miscue, confirming that students guess more frequent words. The negative coefficient on the frequency of the target word (FTARGET) suggests that students tend to make miscues on rarer words.

The next two features involve pronunciation – normalized and absolute distance between target and miscue. Matching the first and last phonemes (FIRSTPHO and LASTPHON) added little predictive power. One reason may be that the training data was restricted to miscues that started with the same first letter as the target word, and are therefore likely to start with the same phoneme.

Student features based on passage level (GRADE and GRADEGRO) were not very predictive. One reason may be that these features correlated with target word frequency rank, which tended to be lower in easier passages.

5. PERFORMANCE ON TEST DATA

Table 2 shows how well the extrapolative model performed compared to the rote model trained on the same data. Its overall coverage of miscue tokens was 38%, versus 22% for the rote model. However, its precision was somewhat worse, in that it predicted an average of 8.8 miscues for each target word type, versus 7.5 for the rote model.

Evaluating coverage for the rote method simply involved counting how many of the actual miscues occurred in the list of miscues predicted for their target word. Predicted and actual miscues were represented in the same phonetic notation, and could therefore be compared using a string equality test.

In contrast, evaluating coverage for the extrapolative method was more complex. We translated each phonetically transcribed miscue in the test set into the spelling of the corresponding real word and generated plausible negative examples by using the same method as for the training data.

To compare the rote and extrapolative methods more informatively, we analyzed performance separately on two parts of the test set, based on the frequency of the target word in English. We knew that the rote model achieved good coverage on the most frequent 100 words of English, and little or no coverage for less frequent words.

For more frequent words, the extrapolative model achieved much lower coverage than the rote model (39% vs. 67%), though with two fewer predictions per target word (9.1 vs. 11.5). However, for less frequent words, the extrapolative model achieved over four times the coverage of the rote model (38% vs. 8%), with only two more predictions per target word (8.7 vs. 6.2). Of course this difference reached an extreme for miscues on the 201 target words in the test data that did not occur in the training data, so that the rote method could not predict them at all. The advantage of the extrapolative model is precisely its ability to predict real-word miscues on words that – like most of English – were not in the training data.

		Rote	**Extrapolative**
Overall	Coverage	22%	38%
	Predictions per word	7.5	8.8
Word rank ≤ 100	Coverage	67%	39%
	Predictions per word	11.5	9.1
Word rank > 100	Coverage	8%	38%
	Predictions per word	6.2	8.7

Table 2. Coverage and predictions per word

6. CONCLUSIONS

Predicting oral reading miscues is important to detecting and remediating them in an intelligent tutor. Here we report, evaluate, and compare two approaches to this problem.

The "rote" approach simply predicts that students will produce the same miscues seen in the training set, especially "popular" miscues that more than one student produced. The rote approach performed surprisingly well, especially on the 100 most frequent words of English.

The "extrapolative" approach focuses on real-word miscues, where the student misreads the target word as some other word. This approach predicts that the relation of target to miscue will be approximately the same as in the training set. This relation is expressed in a feature representation based on the spelling, pronunciation, and frequency of the target and miscue words. The extrapolative approach generalizes to predict real-word errors on words not seen in the training data and it outperformed the rote approach on less frequent words.

It is natural to ask how these methods fare compared to [2], but a direct comparison is problematic. [2] addressed the closely related but slightly different problem of predicting the frequency of different phoneme-level decoding errors. Consequently they reported different evaluation criteria than those here. However, we can usefully compare the three approaches in terms of which miscues they predict. [2] trained phoneme-level malrules that predict miscues whether or not they are words, and whether or not the target word is in the training set – but only if the miscue differs from the target word by adding, dropping, or substituting the individual phonemes specified by the malrules. The rote approach also predicts miscues whether or not they are words, but only for target words that occur in the training data, especially high-frequency target words. The extrapolative approach predicts only real-word miscues, but generalizes to lower-frequency target words that do not appear in the training data.

One future direction is to integrate these methods so as to combine the different regularities they exploit in phonology, spelling, and uneven distribution of words and miscues. It could be especially fruitful to exploit knowledge about the reading skills we want students to learn, so as to characterize which manifested deficiencies in those skills are not only likeliest, but most important to remediate – or to ignore. Finally, we are working to identify which predicted miscues a speech recognizer can detect accurately enough to let Project LISTEN's Reading Tutor listen for them.

ACKNOWLEDGEMENTS

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). This work was supported in part by the National Science Foundation under Grant No. REC-9979894. Any opinions, findings, conclusions, or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the National Science Foundation or the official policies, either expressed or implied, of the sponsors or of the United States Government.

REFERENCES

1. Baffes, P. and Mooney, R., Refinement-based student modeling and automated bug library construction. *Journal of Artificial Intelligence in Education*, 1996. **7**(1): p. 75-116.
2. Fogarty, J., Dabbish, L., Steck, D. and Mostow, J., Mining a database of reading mistakes: For what should an automated Reading Tutor listen?, in *Artificial Intelligence in Education: AI-ED in the Wired and Wireless Future*, J.D. Moore, C.L. Redfield, and W.L. Johnson, Editors. 2001. p. 422-433.
3. Goodman, Y.M. and Burke, C.L., *Reading Miscue Inventory: Manual and procedures for diagnosis and evaluation*. 1972, New York: MacMillan.
4. Mostow, J. and Aist, G., Evaluating tutors that listen: An overview of Project LISTEN, in *Smart Machines in Education*, K. Forbus and P. Feltovich, Editors. 2001.
5. Mostow, J., Roth, S.F., Hauptmann, A.G. and Kane, M., Mostow, J., Roth, S.F., Hauptmann, A.G. and Kane, M. A prototype reading coach that listens. in *Proceeding of the Twelfth National Conference on Artificial Intelligence*. 1994
6. Nix, D., Fairweather, P. and Adams, B., Nix, D., Fairweather, P. and Adams, B. Speech Recognition, Children, and Reading. in *CHI 98*. 1998
7. Spache, G.D., *Diagnostic Reading Scales*. 1981, Monterey, CA: McGraw-Hill.