Why and How Our Automated Reading Tutor Listens

Jack Mostow
Project LISTEN (www.cs.cmu.edu/~listen), School of Computer Science
Carnegie Mellon University
Pittsburgh, PA, USA
mostow@cs.cmu.edu

Abstract— **Project LISTEN's Reading Tutor listens to children read aloud, and helps them learn to read. This paper outlines how it gives feedback, how it uses ASR, and how we measure its accuracy. It describes how we model various aspects of oral reading, some ideas we tried, and lessons we have learned about acoustic models, lexical models, confidence scores, language models, alignment methods, and prosodic models.**

Keywords: speech recognition; reading tutor; oral reading

I. INTRODUCTION

Automated reading tutors [1-4] use automatic speech recognition (ASR) to listen to students read aloud. American children typically read aloud in grades 1-2 (ages 6-7) and are expected to be fluent silent readers by grade 4, often called the transition from "learning to read" to "reading to learn."

Reading is more than turning text into speech; its goal is to making meaning from print. Thus reading requires the ability to map graphemes to phonemes; decode new words; identify familiar words quickly; read connected text quickly, accurately, effortlessly, and expressively; retrieve context-appropriate word senses; comprehend the meaning of text; and stay motivated enough to practice reading and build fluency.

From the viewpoint of speech recognition, children's oral reading is often marked by hesitations, false starts, miscues (reading mistakes), regressions (rereading one or more words), list-like prosody, and off-task speech. Deviations from a dictionary pronunciation of a text word include mistakes in decoding, identifying, or pronouncing the word, dialect phenomena, and individual speech defects, such as the inability to produce or distinguish certain phonemes. As readers gain fluency, they hesitate less often, regress less, make fewer miscues, and read faster and more expressively.

A note about terminology: to reduce the potential for confusion, the word "mistake" refers in this paper to incorrect reading or pronunciation by the child; the word "error" refers to incorrect listening by the computer.

This paper is organized as follows. Section II describes why the Reading Tutor listens. Section III defines our measures of how well it listens. Section IV discusses how we represent and train the models it uses to listen. Along the way, we discuss some of the approaches we tried over the past 20+ years, and lessons we learned. Finally, Section V concludes.

II. PURPOSES OF LISTENING IN A READING TUTOR

The Reading Tutor listens for several purposes. By detecting speech and silence and using timing information, it decides when and how to respond. By aligning the ASR output with the text, it tracks the reader's position in the text. By comparing each text word with the hypothesized word aligned against it, it detects oral reading miscues. By analyzing the time alignment of the ASR output, it computes how long the student takes to identify each word and read it aloud. By extracting the pitch, amplitude, and duration of read words, it computes their prosodic contour. We mine these various sorts of information off-line to assess students and evaluate tutor actions, but this paper is about the speech information the Reading Tutor uses at runtime.

A session with the Reading Tutor starts when the child clicks *Hello* and uses a talking menu interface to log in by clicking on his or her name and (as a light-weight but easy-to-remember password) birth month. The Reading Tutor then takes turns with the child at picking a text to read or other activity to do, such as jointly composing a story. The session ends when the child logs out by clicking an on-screen *Stop* sign, or times out by not speaking or clicking for 30 seconds, or if the Reading Tutor crashes or hangs.

Reading Tutor activities are built out of several types of steps, each with its own screen interface: assisted oral reading (and narrating); tutor instruction; multiple choice questions; keyboard input; using on-screen letter tiles to build words and sound them out; and free spoken responses.

Project LISTEN has focused primarily on assisted oral reading, and so does this paper. Some other types of steps also involve listening. In word-building steps, the Reading Tutor prompts the child to sound out the word, and tries to

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grants R305B070458 and R305A080628, and 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 author and do not necessarily reflect the views or official policies, either expressed or implied of the Institute, the U.S. Department of Education, the National Science Foundation, or the United States Government. I thank the educators and students who helped generate our data, and the many LISTENers over the years who co-authored the work summarized in this paper and cited in the References.

follow along, but does not attempt to detect mistakes. In free-response steps, the Reading Tutor graphically indicates the approximate amount of speech, but records it without trying to recognize it at runtime. However, we've worked on recognizing some types of speech off-line [5-7].

The Reading Tutor reacts to speech, mouse clicks, and delays by responding with graphical and spoken feedback, described respectively in Sections II.A and II.B.

A. *Graphical interface*

Assisted oral reading uses a graphical interface. As the screenshot in Figure 1 shows, a robot persona provides a visible audience by blinking sporadically to appear animate, gazing at the current word to appear attentive, and displaying a volume meter to show that it's listening. Up and down buttons adjust its output volume. (We hide the input level control to protect it from misadjustment.) Navigation buttons at the top of the screen consist of *Stop* (to quit the story) and *Go* (to advance to the next sentence). The Reading Tutor displays text on a book-like background by adding one sentence at a time and graying out the previous sentences, unlike educational software that displays text page by page like a book.

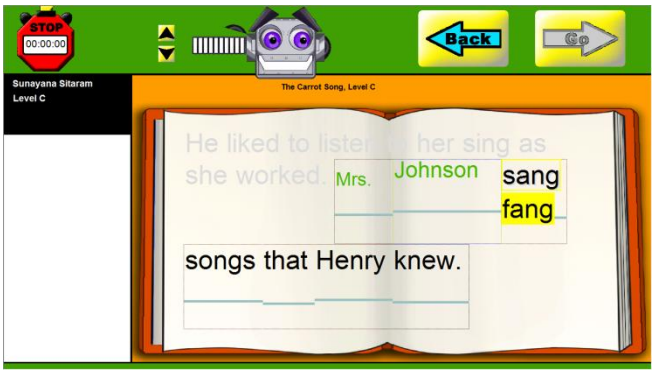

Figure 1: Reading Tutor screenshot (2012)

The Reading Tutor displays text sentence by sentence for three reasons. One reason is ASR accuracy. Controlling which sentence is displayed imposes a strong constraint on what the student can read aloud. A second reason is usability of the spoken dialogue. Before displaying the next sentence, the tutor has an opportunity to intervene without risk of interrupting the student. A third reason is pedagogical. Postponing the display of the next sentence frees the tutor to decide on the fly what to display next, e.g. to help decode a hard word, explain unfamiliar vocabulary, or give comprehension assistance.

The Reading Tutor maps various components of its internal state to graphical properties, such as word color, background color, shadowing, and underlining. It displays earlier sentences in gray, words to read in black, credited words as green, and future sentences in white, i.e. invisible. It shadows the word it thinks the child will read next, underlines a word to prompt the child to read it, boxes the word the cursor is on, and highlights the background of a word in yellow while reading it aloud or hinting how to do so, which may involve temporarily showing a rhyming or other related word below it. Figure 1 shows the Reading Tutor saying "rhymes with *fang*" as a hint to decode *sang*.

The Reading Tutor can display assisted reading rate on a "readometer" while the child is reading a story, and in an on-screen certificate after the story as a reward for finishing it. In addition, it can provide real-time graphical feedback on the child's oral reading prosody [8], for example by mapping the loudness of a word read by the student to its size, its pitch to its vertical position, and the narrator's pitch contour to a staircase-like sequence of lines. Thus in Figure 1, *Mrs.* is smaller because it was spoken softly, *Johnson* is higher because it was spoken with rising inflection characteristic of a question or guess, both words are green because the Reading Tutor has credited them as read, and the color, size, and position of the remaining words are unchanged because the child hasn't read them yet.

B. *Multimodal dialogue*

The Reading Tutor's dialogue architecture [9, 10] is driven by speech, silence, time, and mouse clicks. It represents turn-taking state in terms of four binary variables:

- Is the student speaking?
- Is the Reading Tutor speaking?
- Does the student have the floor?
- Does the Reading Tutor have the floor?

Each variable has a timer that records when it last changed.

At any point in time the student, the Reading Tutor, both, or neither may have the floor. Transitions between states occur when the student or Reading Tutor starts or stops speaking, or when one of the timers reaches a specified threshold value. The states and timers govern whether, when, and how the Reading Tutor speaks.

For example, after a 2-second silence, the Reading Tutor may backchannel to encourage the student to continue reading. However, if the student remains silent for 2 additional seconds, the Reading Tutor takes the floor to verbally prompt the student to click for help.

Backchanneling does not take the floor away from the student. For instance, if the Reading Tutor detects a skipped word, it underlines the word and coughs to draw attention to it, but it's still the student's turn.

Usually the Reading Tutor does not take the floor when the student has it. An exception is choral reading, when the Reading Tutor prompts the student to "read with me."

The Reading Tutor takes the floor when the child clicks the mouse, whether on *Stop* (to quit the story), *Go* (to advance to the next sentence), a word (to get help reading it), below the sentence (to hear the Reading Tutor read it), or elsewhere on the screen (by mistake).

By design, the Reading Tutor responds to all clicks rather than ignore the child, even if only to explain why it can't perform the requested action. For instance, if the child clicks on the *Go* button without reading at least half the sentence, the Reading Tutor says "Sorry, can't go on right now." To make its behavior easier to understand, the Reading Tutor responds to mouse clicks immediately to

make clear what it is responding to. Thus it interrupts itself if it is speaking, rather than wait to finish what it is saying. Waiting would complicate how it represents and displays the dialogue state.

The Reading Tutor also takes control when it hears the student read the end of the sentence. If any content words remain uncredited, it waits for the student to read. If it heard the student read the entire sentence fluently, it advances to the next sentence without intervening. If not, it reads the sentence aloud first, so as to scaffold comprehension, because failure to read the sentence fluently indicates that the child may not have understood it.

The Reading Tutor uses the overall distribution of inter-word latencies [11] to estimate the child's reading level [12], which it uses in deciding which stories to pick from.

In short, listening to a child read aloud enables the Reading Tutor to decide when and how to give feedback, track the child's position in the text, compute the latency before a word and the time to read it, detect miscues, assess children's oral reading fluency, and mirror their oral reading prosody. We next discuss ways to define and measure the accuracy of its listening for these purposes.

III. Listening Accuracy Metrics for Oral Reading

Over the years, we have evaluated the Reading Tutor's listening accuracy in several different ways.

A. Miscue detection accuracy

At first we focused on measuring accuracy in detecting oral reading miscues. Conventional word error rate can measure accuracy in *recognizing* miscues – i.e. in transcribing them. However, word error rate does not measure accuracy in *detecting* miscues [13].

The Reading Tutor detects miscues by aligning the hypothesis output by the ASR against the sentence displayed. Thus detecting a miscue does not require the ASR to recognize it correctly, merely to recognize it as anything other than the word the child was supposed to read.

We have measured miscue detection at different levels. At the highest level, which we call "text space," we treat miscue detection as a classification problem: classify each text word as read correctly, misread, or omitted, or simply as read correctly or not [14]. At this level, we define a miscue as a text word the child failed to read in the course of reading the sentence. This criterion treats false starts, sounding out, incorrect attempts, and other insertions as steps toward the goal of reading the text word, not as mistakes to remediate if they culminate in reading it correctly. This pedagogical policy means that ASR insertion errors don't matter except if hallucinating a word causes the Reading Tutor to misclassify it as read correctly, or leads the ASR astray, causing it to make deletion or substitution errors.

At the next level, which we call "speech space," we classify each *transcribed* word instead. Thus at this level each failed attempt to read a word counts as a miscue whether or not the child subsequently read the word correctly. These misreadings provide a welcome source of training and test data, since text-space miscues are scarce.

At the most detailed level, which we call the "time domain," we classify *time-aligned* transcript words. Time domain accuracy is more stringent. For instance, scoring an accepted word as true requires that it occur in approximately the same time interval in the time-aligned ASR output as in the time-aligned transcript.

B. Tracking accuracy

More recently we have measured tracking accuracy as well. These measures evaluate the Reading Tutor's estimate of the child's position in the current sentence. By aligning ASR output or a reference transcript of the child's oral reading against the sentence, we obtain a *trace*: a sequence of integer positions in the sentence, where position *i* represents the i^{th} word of the sentence. The sign of the integer encodes whether the aligned word matches the text word: + if yes, – if no.

The minimal edit distance between traces based on the reference transcript and the ASR output is a "speech space" measure of tracking error, defined as the number of insertions in, substitutions for, and deletions from the trace based on the reference transcript to turn it into the trace based on the ASR hypothesis.

For example, consider these alignments of transcribed and recognized readings to the text "Once upon a time, the dog":

```
Ref.: once up upon the time      dog
      +    –  +    –   +         +
Text: Once1 upon2   a3  time4,the5 dog6
      +     –      +   +    +    +
Hyp.: ONCE /AH_P/  A   TIME THE  DOG
```

Here "+" and "–" show if the aligned word matches the text. The transcript-based trace is +1, −2, +2, −3, +4, +6, where −2 comes from misreading "upon" as "up," and −3 comes from misreading "a" as "the." The hypothesis-based trace is +1, −2, +3, +4, +5, +6, where −2 comes from recognizing "up" as a truncation of "upon."

Aligning the two traces to minimize edit distance yields the sequence +1/+1, −2/−2, +2/, −3/+3, +4/+4, /+5, +6/+6 where:

+I/+I = accepted reading
–I/–I = detected miscue at correctly tracked position
+I/–I = false alarm at correctly tracked position
–I/+I = undetected miscue at correctly tracked position
/+I = inserted text word
/–I = inserted miscue or garbage
+I/ = deleted text word
–I/ = deleted miscue or garbage
+I/+J = mistracked accepted word
–I/–J = mistracked miscue
+I/–J = mistracked false alarm
–I/+J = mistracked undetected miscue

Thus +1/+1, −2/−2, +2/, −3/+3, +4/+4, /+5, +6/+6 shows that the transcript and hypothesis agree that the reader read $Once_1$ and misread $upon_2$. Then the ASR omits the correct

rereading of $upon_2$, and accepts the word a_3 rejected by the transcript. They agree that $time_4$ and dog_6 were read correctly, but the ASR hallucinates the word a_5.

"Time domain" measures of tracking accuracy take into account whether transcribed and hypothesized words occur at the same time in the speech signal [15]. Such measures compare time-aligned traces to determine the relationship between the transcript and hypothesis [16]. Each segment of a time-aligned trace is either a silence (#), a word that is aligned against a matching text word (+), or a word that is not (–).

A transcript segment and hypothesis segment that overlap in time have one of three temporal relations (labeled as shown). The midpoint of each one can fall within the other (=). The midpoint of the transcript segment can fall within the hypothesis segment, but not vice versa (<). Otherwise, the transcript segment contains the midpoint of the hypothesis segment, but not vice versa (>).

Finally, if transcribed and hypothesized segments that overlap in time are not silences, they may or may not be aligned to the same text word. We mark the latter case "J", short for the I/J notation used above for speech space.

To illustrate this notation, here it is for the fragment above:

```
1 Once  (+=+)   ONCE
2       (#=#)
3 up    (-=-)   /AH_P/
4       (#<#)
5 upon  (+<#)
6 the   (-=+)   A
7       (#=+)   TIME
8 time  (+=#)
9 dog   (+=+)j  THE
10      (#=+)   DOG
```

Transcript segments 1-3 match the times and text positions of the first three hypothesis segments. Segment 5 has an ASR deletion error: where the transcript contains "upon", the hypothesis contains a continuation of the preceding silence. In segment 6, the transcribed and hypothesized words have matching times and text positions, but the hypothesized word matches the text while the transcribed word does not. In segments 7 and 8, the transcript and hypothesis have the same word, but at different times. In segment 9, the transcribed and hypothesized words match different text words. Segment 10 has an ASR insertion error: where the transcript has silence, the hypothesis has the word DOG.

One time domain measure of tracking accuracy is how often (as a percentage of time) the position computed by the Reading Tutor based on the ASR output agrees with the child's position at the same point in time according to the transcript. Another measure is the average absolute distance (in words) between the two positions.

Off-line measures of tracking accuracy [16] are based on the final hypothesis output by the ASR at the end of the utterance. In contrast, real-time measures of tracking accuracy are based on the partial hypotheses output by the ASR as the child reads, and can therefore be considerably lower than off-line measures. The accuracy of real-time tracking trades off against its timeliness. Waiting as little as 0.2 seconds to estimate the reader's position yields a substantial increase in its accuracy [17].

To understand accuracy better, we wanted to distinguish regions of oral reading from regions of off-task speech. To identify off-task speech automatically in a transcript, we defined *deviation length* as the number of consecutive transcribed words without two successive matches to the text words aligned against them. By inspecting deviations of different lengths, we determined that deviations longer than 2 were nearly always off-task speech. Not surprisingly, tracking accuracy is much higher during on-task than off-task speech.

C. Accuracy of confidence metrics

A confidence metric estimates the probability or other score of whether an ASR word hypothesis is correct, or of whether the child read a word correctly. We measure the accuracy of the confidence metric by binning it, say into percentiles, and correlating the percentile for each bin against the actual percentage of words in each bin correct according to the reference transcript.

D. Indirect measures of accuracy

Besides the direct measures of listening accuracy discussed above, we have tested the Reading Tutor's listening indirectly by its ability to predict other measures, such as children's help requests [18], performance on cloze questions [19], and scores on paper tests of oral reading fluency [12], word identification [20], and comprehension [21, 22]. We measure predictive accuracy as correlation of predicted to actual scores, reaching 0.9 in some cases.

E. Micro-efficacy

A fine-grained test of the Reading Tutor is the impact of its instruction and practice on children's fluency in reading the taught or practiced word. We have used inter-word latency [11, 23] and word reading time as micro-measures of oral reading fluency at the level of individual words. We have used two types of methodology to test micro-efficacy.

An "invisible experiments" methodology inserts within-subject randomized controlled trials of alternative tutor actions in the Reading Tutor, with latency or reading time as the outcome of each trial, and aggregates the outcomes of such trials over many students and words [24]. One such experiment used thousands of trials to compare the impact of different ways to preview new words before a story on accuracy in reading them after the story [25]. Another experiment used over 180,000 trials to compare different forms of help on words based on how often the child read the word fluently at the next encounter [26, 27].

A model-fitting methodology uses data logged by the Reading Tutor to compare different types of practice. We have mostly used two classes of model.

A Dynamic Bayes Net model of knowledge tracing [28] estimates the value of a practice type as the probability of the student learning a word from an encounter of that type.

For instance, Beck [29] used this method to tease apart the immediate scaffolding effects of tutor assistance on performance from its subsequent effects, finding a small but statistically significant contribution of help to student learning.

The learning decomposition method [30, 31] uses an exponential decay model of word reading time to estimate the relative value of each type of practice as a coefficient on the number of encounters of that type. For instance, a learning decomposition comparison of rereading vs. new reading found that seeing a word again in a sentence seen before was worth only about half as much as seeing it in a new sentence. Other learning decomposition analyses compared massed vs. distributed encounters of a word, reading text chosen by the child vs. by the reading tutor [32], and transfer to similar words [33, 34]. More recently, we have used linear mixed effects models to predict the log of reading time so as to account properly for statistical dependencies on students, words, and stories by modeling them as random effects.

F. Macro-efficacy

The ultimate test of the Reading Tutor is its impact on the reading proficiency of the children who use it. To measure this impact, we have performed controlled studies to compare children's pre- to post-test gains on tests of various reading skills from using the Reading Tutor compared to gains from other treatments, including classroom instruction, other software, independent reading practice, and individual human tutoring [35-37]. So have other researchers [38-42]. We measure the Reading Tutor's impact on a tested skill as an effect size: the difference between the mean test score gains for two treatments, divided by the within-treatment standard deviation. Effect scores of 0.3 are considered small, 0.5 medium, and 0.8 large [43]. The Reading Tutor's effect sizes for fluency gains reached as high as 1.3 standard deviations in some studies, varying by comparison condition and student population, with English language learners apparently benefitting the most.

It is natural to ask how lower-level listening accuracy in tracking the reader and detecting miscues affects the Reading Tutor's educational efficacy. Unfortunately, this question is more readily asked than answered. Comparing the macro-efficacy of Reading Tutor versions that differ in the accuracy of their listening would be costly in time, money, and sample size. Analyzing the effects of deliberate listening errors on micro-efficacy might be more feasible, but it is far from clear that such effects are local. For instance, frustration caused by listening errors might be cumulative. How much low-level listening accuracy affects educational efficacy remains a question for future research.

IV. Automated Analysis of Oral Reading

The Reading Tutor uses Sphinx2 [44] to recognize read words, signal processing to extract their pitch and amplitude, and post-processing to support feedback and assessment.

The Reading Tutor's key ASR components include its acoustic-phonetic models, pronunciation lexicon, acoustic confidence scores, language models, and alignment methods. We now describe how we have represented and trained each of these models over the years, sometimes in a series of different ways.

A. Acoustic models

Project LISTEN originally used semi-continuous HMM acoustic models trained on adult female speech [14, 45]. After recording a small corpus of children's oral reading in a Wizard of Oz experiment, we used it to adapt the codebook means of our models [46]. Once we had a larger transcribed corpus of children's oral reading in the Reading Tutor, we trained HMMs on it from scratch. We trained continuous models once computers became fast enough to use them to recognize oral reading in real-time.

We had much less oral reading manually transcribed than not, so we tried training on untranscribed speech. We knew the text sentence that each utterance was an attempt to read, and we used cherry-picking heuristics to select the utterances likely to be or contain correct readings [47]. The resulting models performed better on a test set of children's oral reading recorded under similar conditions than training on the manually transcribed KIDS corpus [48, 49] of comparable size (approximately 5,000 utterances), collected under more controlled conditions in a quieter environment.

Despite this promising result, once we had accumulated a manually transcribed corpus of tens of thousands of oral reading utterances recorded by the Reading Tutor during normal use, automatically labeled data did not help; in fact, it actually hurt ASR accuracy when used to augment the manually transcribed training data. That is, ***quality trumps quantity*.**

B. Lexical models

The Reading Tutor's active lexicon changes from sentence to sentence, taking advantage of knowing which sentence is currently displayed. The lexicon contains the words in the sentence. Their pronunciations come from CMUDICT [50] if it contains them, otherwise from the pronunciation component of a speech synthesizer.

The lexicon also contains distracters to model misreading and false starts. Over the years we have experimented with several types of distracters.

The only distracters we still use are the first kind we tried, namely phonetic truncations of the sentence words. For a word *w* whose pronunciation is *n* phonemes long, we add a distracter START_*w* with multiple pronunciations. They consist of initial subsequences of the *n* phonemes, containing at least the first 2 phonemes and at most *n*-2. Adding the truncation distracters increased miscue detection without increasing the false alarm rate (correctly read words misclassified as miscues). The resulting ASR detected about half the miscues rated by a human judge as serious enough to threaten comprehension, which in turn constituted only about half of the words whose transcription differed from the text – the more stringent criterion we used in our

later evaluations. The ASR rejected about 4% of correctly read words [46]. We prioritize accepting correct reading over detecting miscues, because children read 90% of words correctly unless the text exceeds their frustration level [51]. Also, rejecting a correctly read word frustrates the child, whereas accepting a miscue at worst confuses the child, though it may reinforce mislearning.

We initially included pronunciations with the first *n*-1 phonemes as well, but they reduced ASR accuracy by getting recognized too often in place of a correctly read word. One reason was a dialect phenomenon common among the children in our sample, namely dropping final consonants, such as /S/ at the end of a plural noun like *cats* or present tense verb like *sits.* Such a truncation may be a pronunciation mistake, but it does not constitute an oral reading miscue if it's the reader's normal pronunciation of the word. We therefore tried adding such truncations as alternate pronunciations for the correct word, but they reduced ASR accuracy by making it too easy to hallucinate. Accordingly, we do not include the first *n*-1 phonemes as a pronunciation, either of the correct word or as a distracter. This change remains our only accommodation to dialect phenomena.

The ASR often accepted misread short sentences as read correctly. The reason is that the ASR maps oral reading to the sequence of sentence words and distracters that it most resembles. Consequently, it typically does not detect a miscue unless the miscue resembles either a distracter for the correct word, or another word in the sentence, more than it resembles the correct word. A short sentence has fewer words for a miscue to resemble.

In an attempt to compensate for this limitation, we needed some additional distracters to help model miscues. We didn't want them to be too easy to hallucinate, so we refrained from adding individual phones as distracters. Instead, for short sentences we added as distracters a few two-syllable words used to spell out words over noisy radio connections: *alpha*, *bravo*, etc. Unfortunately, although they helped detect more miscues, they also hallucinated more miscues, so we wound up taking them back out.

Next we took a more systematic approach to miscue detection. By predicting likely miscues, we hoped to increase miscue detection without increasing false alarms. We explored three methods for predicting likely miscues.

The first method worked at the level of individual letter sounds, or more precisely graphophonemic mappings. For years, renowned reading researcher Richard Olson and his University of Colorado colleagues had been comparing the reading difficulties of identical and fraternal twins in order to quantify their genetic component. In the process they had recorded, phonetically transcribed, and annotated hundreds of twins' oral readings, in the process accumulating a database of tens of thousands of oral reading miscues. As a group project in a graduate course in machine learning, Fogarty *et al.* [52] mined this corpus to discover "malrules" that predict decoding mistakes at the level of individual graphophonemic mappings. Each malrule predicted that a grapheme *G* that should be decoded as some phoneme *P* would instead be decoded as some other phoneme *P'*. The 10 most frequent malrules turned out to be insertions and deletions. For instance, the two most frequent *G→P→P'* rules were *s* → /S/ → _ and *s* → /Z/ → _, where _ denotes the empty string. These malrules predict deletion of the plural endings of *plant*<u>s</u> and *arm*<u>s</u>, respectively.

The other two methods [53] exploited the fact that most miscues consist of misreading one word as another. The "rote" method simply identified misreadings made by two or more readers on the 100 most frequent words in the corpus, and predicted that those misreadings would continue to occur. The "extrapolative" method generalized the relation between words and real-word misreadings of them, and predicted analogous misreadings of other target words.

Unfortunately, adding distracters other than the truncations targeted just the specific predicted miscues. They might detect a few more miscues with slightly fewer false alarms, but they increased the miscue detection rate significantly only by also increasing the false alarm rate [54]. In short, ***distracters detract.*** We therefore gave up on distracters to look for a more generic way to detect miscues.

C. Confidence scores

To detect miscues without specifically predicting them in advance, we tried a confidence metric approach. One metric [55] trained decision trees using three types of features.

Decoder-based features used word-level information from the ASR output, namely "log energy normalized by number of frames, acoustic score normalized by number of frames, language model score, lattice density, averaged phone perplexity, and duration."

Alignment-based features used contextual information about the target text word from the alignment of the ASR output against the sentence, such as whether the ASR accepted the word, the latency preceding the word, the number of previous or subsequent text words hypothesized in a row, and the average distance between hypothesis words aligned against the target word.

History-based features used information logged by the Reading Tutor about the student. Word-level features included how many times the student had encountered the target word in the past, how many of them were accepted, and the student's average latency before words in general. Utterance-level features of the current sentence included the number of utterances so far, and averaged over them, the number of words attempted, the number accepted, the number of jumps, and the number of regressions to the start of the sentence.

This method trained two decision trees that operated in "text space." The first decision tree estimated the probability that an accepted word was actually misread, based primarily on (i.e. using in the top two levels of the decision tree) phone perplexity, log energy, and acceptance by the ASR. To undo ASR deletion errors, the second decision tree estimated the probability that a rejected word was actually read correctly, based primarily on the number of successive text words preceding and following it.

With a training set of 3714 utterances and a test set of 1883 utterances by different children, and a baseline of 56% miscue detection and 4% false alarm rate, the method could either increase miscue detection to 59% or reduce the false alarm rate to 3%. These miscue detection rates are inflated due to treating unattempted words as deletions, so their actual values aren't meaningful, but the changes to them still show improvement.

By 2007, we had reduced the false alarm rate below 1%, with 23% detection of substitution miscues defined as a mismatch between the spoken word and the text according to the manual transcript. Since "text space" miscues are much rarer than correctly read words, we decided to evaluate "speech space" accuracy so as to measure ASR performance more sensitively. Tracking error, defined as the combined substitution and deletion rate in speech space, was below 2%, but the insertion rate was almost 17%.

We tried using a more conventional (i.e. speech space) acoustic confidence metric [56, 57] to filter ASR output. The confidence threshold ROC curve for the tradeoff between false positives and true positives exceeded 0.83 AUC (Area Under Curve). We expected a confidence metric to be a good way to decide whether a recognized word was in fact read correctly, because in principle, it should be able to detect miscues without relying on the language model and lexicon to predict them in advance. However, in practice, using a confidence metric to reject misread words is limited by tracking accuracy, because when the ASR goes off-track and recognizes a different word than the one the reader was trying to read, its confidence score is irrelevant – akin to closing the barn door after the cows have escaped. This inconvenient truth defeated our grand scheme to estimate the probability that a word was read correctly by combining acoustic confidence with other information such as a model of the student. That is, ***tracking trips up scoring*****.**

D. Language models

The Reading Tutor uses a simple probabilistic finite state model of oral reading, which it generates on the fly for each sentence before displaying it [46]. In state *i*, it expects word *i* of the *n*-word sentence (with PrCorrect), a truncation of word *i* (with PrTruncate), a premature end of the utterance (with PrEndEarly), or a jump to state *j*, with different probabilities depending on *i* and *j*. A file specifies probabilities for the parameters PrCorrect, PrTruncate, PrEndEarly, PrRepeat, PrSkip, PrRestart, PrJumpBack, PrJumpForward, etc.

Initially this model was approximated as a bigram model. ASR accuracy improved when Ravi Mosur extended Sphinx2 to input finite state models and use them top-down. In contrast to bottom-up recognition, which relied on a lexicon-driven recognizer to hypothesize words, the top-down recognizer enabled high language model probabilities to overcome poor acoustic scores of words that the bottom-up method would have failed to recognize in the first place.

A classifier learning approach [58] reduced the speech space tracking error by adjusting language model probabilities iteratively. At each iteration, it used the language model from the previous iteration to recognize a training set of oral reading utterances, aligned the ASR output for each utterance against the target sentence to compute a trace, and scored it against the trace based on the transcript. It applied a simple credit assignment heuristic [59] to transitions between successive words in the recognized trace, classifying transitions that stayed on track as positive, and transitions that led off-track as negative. After using LogitBoost to learn a classifier from the labeled transitions, it increased the probability on transitions classified as positive, decreased the probability on transitions classified as negative, and used the adjusted language models to re-recognize the utterances. It repeated this cycle until tracking error started to rise. This method reduced tracking error from 9% to 7%, but was impractical to incorporate in the Reading Tutor because it involved applying the entire sequence of learned classifiers to the initial language model.

We explored various alternatives to simple *n*-state models. A key question was which additional states to include. For instance, the "watermark" model used $O(n^2)$ states of the form (i, j) to represent the reader being at word *i* and having previously read as far as word *j*. After attempts to design better finite-state models by hand, we extended Sphinx2 to allow non-finite-state models, and let a user-defined function directly compute the probability $\Pr(w \mid h)$ of word *w* following the preceding sequence *h* of recognized words. A SVM trained on such features as the frequencies of different transition types in *h* yielded a language model that reduced perplexity by a factor of 4 relative to the baseline. However, it merely slowed down the ASR by orders of magnitude without improving its accuracy.

E. Alignment methods

A key step in scoring oral reading is aligning the ASR output and manual transcript against the text to compute traces. The standard NIST align procedure is ill-suited to this purpose because it treats regressions (rereading one or more words) as insertions instead of as normal reading. Instead, we developed the MultiMatch alignment procedure to take regression into account.

MultiMatch uses dynamic programming to find the lowest-cost mapping from a sequence of recognized or transcribed words to positions in a text sentence. It imposes a mismatch penalty for aligning a word against a text word it does not match. This penalty reflects the orthographic and phonemic distance between them. MultiMatch imposes a jump penalty for a transition from position *i* to any position except *i* or *i*+1.

The penalties are set to prefer an isolated mismatch to jumping to a word and back. For instance, in aligning the reading *once upon the time* ... to the text "Once upon a time the beautiful princess …," MultiMatch aligns *the* against the text word "a" rather than jump forward in the sentence to match the word "the" and back to match the word "time."

MultiMatch outputs alignments in both text space and speech space. The text space alignment associates each

word of text with at most one spoken word. The speech space alignment associates each recognized or transcribed word with the text word it is aligned against.

Having recognized the crucial importance of tracking accuracy and spending years trying to improve it, with scant success, we decided to address the problem of tracking by redefining it. We had framed this problem as "chasing the kid" – that is, finding whichever word the child was trying to read. We decided to reduce the problem to "blaming the kid" – that is, deciding whether the child was reading whichever word the tutor determined should come next, namely the earliest uncredited word in the sentence. The tutor could then simply wait to hear this word. To avoid getting stuck at false alarms, the tutor could skip over at most one text word to accept the next word. To make its behavior more understandable, the tutor could highlight the word it is waiting to hear (though it does not yet do so).

Sure enough, tracking accuracy was substantially higher with this redefined criterion [16]. However, when we modified the language model to use the same criterion, tracking accuracy suffered. We concluded that "chase the kid" was more accurate at tracking the child's actual position, even if we used "blame the kid" to indicate which word to read next. We believe the reason is that "chase the kid" is a more accurate model of actual reading behavior, and therefore tracks the reader's actual position more accurately. In contrast, the monotonic left-to-right "blame the kid" language model is apt to get lost when it fails to follow the reader. The lesson is to use a faithful model of reading to track the reader's actual position, even if the tutor refrains from displaying it externally. I.e., ***rely on realism*** but ***mask mistracking.***

F. Prosodic models

Expressiveness is an important aspect of oral reading fluency. To assess children's oral reading fluency, we built on work [60-65] by Schwanenflugel and her colleagues, who analyzed the development of children's oral reading prosody and related it to their gains in fluency and comprehension. Given a child's oral reading of a sentence, they measured its expressiveness by correlating its prosodic contour – that is, the word-by-word sequence of pitch, duration, and intensity – against adult prosodic contours for the same sentence.

First we scaled up from Schwanenflugel *et al.*'s painstakingly hand-measured prosodic features of a few utterances to comprehensive automated assessments of children's prosodic contours by correlating them against the contours of the Reading Tutor's recorded fluent adult narrations of the same sentences [66]. We analyzed the sensitivity of this template-based measure to prosodic improvements in a child's successive readings of a sentence on the same or different days [67].

Then we generalized this approach by using the adult narrations to train a normative model of oral reading prosody, and using the trained model to score children's oral reading prosody [68]. The generalized model outperformed the template-based measure in predicting children's end-of-year scores and gains in fluency and comprehension [21]. It used only duration information, but latencies are very informative. That is, ***silences are golden.***

Next we used the generalized model to mine a corpus of children's oral reading in order to identify the specific common syntactic and lexical features of text on which children scored best and worst. These features predicted their fluency and comprehension test scores and gains better than the previous models.

Meanwhile, to explore how to give children real-time feedback on their oral reading prosody, we developed a flexible prosody visualization tool for mapping each word's prosodic features to graphical features, in order to user-test experimenter-specified mappings [8]. For instance, this tool can map a word's pitch to its vertical position, loudness to font size, and temporal features to the timing of the dynamic display. It can map multiple features to different dimensions of color, such as hue, saturation, and intensity. Mapping "adult-likeness" to hue provides visual feedback on the proximity of the child's pitch, duration, and/or intensity to the narrator's. Mapping latency to intensity makes higher-latency words pale so as to reflect tentative, hesitant reading. Mapping ASR confidence to saturation makes lower-confidence words look more like unread words, to reflect uncertainty that they were read correctly.

V. CONCLUSIONS

In over two decades of applying speech recognition to children's oral reading, Project LISTEN has learned a number of lessons about what worked – and more often, what didn't, at least for us – and found some hard questions:

Acoustic models: ***Quality trumps quantity.*** Augmenting a large corpus of manually transcribed oral reading with ASR output filtered to serve as automated transcripts hurt accuracy. Is there a way to make it help?

Lexical models: ***Distracters detract.*** Except for phonetic truncations of sentence words, predicting likely miscues detected more of them only by hallucinating them as well. What if any distracters are worth listening for?

Confidence scores: ***Tracking trips up scoring.*** Confidence scores of mistracked words are useless. How can confidence scores be made robust to mistracking?

Language models: ***Rely on realism.*** The better we model children's oral reading, the better we can track it. What if any models boost tracking accuracy dramatically?

Alignment models: ***Mask mistracking.*** It's easier to tell if children are at the right spot than where they are instead, and even easier to prompt them to click but not say where. Can alignment plus interface redesign hide tracking errors?

Prosodic models: ***Silences are golden.*** Duration of latency between words is a good gauge of reading fluency. How much can tracking better make latency measure better?

ASR is notoriously empirical, so what failed for us may work for others, and possibly vice versa. Thus these lessons come without guarantees of generality. However, if they steer readers towards fruitful approaches and away from fruitless ones, they will have served a useful purpose.

REFERENCES (many at www.cs.cmu.edu/~listen)

[1] M. J. Adams, "The promise of automatic speech recognition for fostering literacy growth in children and adults," in *International Handbook of Literacy and Technology*. vol. 2, M. McKenna, L. Labbo, R. Kieffer, and D. Reinking, Eds. Hillsdale, NJ: Lawrence Erlbaum Associates, 2006, pp. 109-128.

[2] J. Mostow and G. S. Aist, "Giving help and praise in a reading tutor with imperfect listening -- because automated speech recognition means never being able to say you're certain," *CALICO Journal*, vol. 16, pp. 407-424, 1999.

[3] A. Hagen, B. Pellom, and R. Cole, "Highly accurate children's speech recognition for interactive reading tutors using subword units," *Speech Communication*, vol. 49, pp. 861-873, December 2007.

[4] V. L. Beattie, "Scientific Learning Reading Assistant™: CMU Sphinx technology in a commercial educational software application," in *CMU Sphinx Users and Developers Workshop*, Dallas, TX, 2010.

[5] W. Chen, J. Mostow, and G. Aist, "Using Automatic Question Generation to Evaluate Questions Generated by Children," in *Proceedings of the AAAI Symposium on Question Generation*, Arlington, VA, 2011.

[6] W. Chen and J. Mostow, "A Tale of Two Tasks: Detecting Children's Off-Task Speech in a Reading Tutor," in *Interspeech: Proceedings of the 12th Annual Conference of the International Speech Communication Association*, Florence, Italy, 2011.

[7] X. Zhang, J. Mostow, N. K. Duke, C. Trotochaud, J. Valeri, and A. Corbett, "Mining Free-form Spoken Responses to Tutor Prompts," in *Proceedings of the First International Conference on Educational Data Mining*, Montreal, 2008, pp. 234--241.

[8] S. Sitaram, J. Mostow, Y. Li, A. Weinstein, D. Yen, and J. Valeri, "What visual feedback should a reading tutor give children on their oral reading prosody?," in *Proceedings of the Third ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, Venice, Italy, 2011.

[9] G. Aist and J. Mostow, "A time to be silent and a time to speak: Time-sensitive communicative actions in a reading tutor that listens," in *AAAI Fall Symposium on Communicative Actions in Humans and Machines*, Boston, MA, 1997.

[10] G. Aist, "Expanding a time-sensitive conversational architecture for turn-taking to handle content-driven interruption," in *Proceedings of the International Conference on Speech and Language Processing (ICSLP98)*, Sydney, Australia, 1998, p. #928.

[11] J. Mostow 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)*, Providence, RI, 1997, pp. 355-361.

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

[13] J. Mostow, "Is ASR accurate enough for automated reading tutors, and how can we tell?," in *Proceedings of the Ninth International Conference on Spoken Language Processing (Interspeech 2006 — ICSLP), Special Session on Speech and Language in Education*, Pittsburgh, PA, 2006, pp. 837-840.

[14] J. Mostow, A. G. Hauptmann, L. L. Chase, and S. Roth, "Towards a reading coach that listens: automated detection of oral reading errors," in *Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI93)*, Washington, DC, 1993, pp. 392-397.

[15] G. Doddington, "Word Alignment Issues in ASR Scoring," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU '03)*, St. Thomas, U.S. Virgin. Islands, 2003, pp. 630- 633.

[16] M. H. Rasmussen, J. Mostow, Z.-H. Tan, B. Lindberg, and Y. Li, "Evaluating Tracking Accuracy of an Automatic Reading Tutor," in *Proceedings of the Third ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, Venice, Italy, 2011.

[17] Y. Li and J. Mostow, "Evaluating and improving real-time tracking of children's oral reading," in *Proceedings of the 25th Florida Artificial Intelligence Research Society Conference (FLAIRS-25)*, Marco Island, Florida, 2012.

[18] J. E. Beck, P. Jia, J. Sison, and J. Mostow, "Predicting student help-request behavior in an intelligent tutor for reading," in *Proceedings of the 9th International Conference on User Modeling*, Johnstown, PA, 2003, pp. 303-312.

[19] X. Zhang, J. Mostow, and J. E. Beck, "Can a computer listen for fluctuations in reading comprehension?," in *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, Marina del Rey, CA, 2007, pp. 495-502.

[20] J. E. Beck and J. Sison, "Using knowledge tracing in a noisy environment to measure student reading proficiencies," *International Journal of Artificial Intelligence in Education (Special Issue "Best of ITS 2004")*, vol. 16, pp. 129-143, 2006.

[21] M. Duong, J. Mostow, and S. Sitaram, "Two Methods for Assessing Oral Reading Prosody " *ACM Transactions on Speech and Language Processing (Special Issue on Speech and Language Processing of Children's Speech for Child-machine Interaction Applications)*, vol. 7, pp. 14:1-22, August 2011.

[22] S. Sitaram and J. Mostow, "Mining Data from Project LISTEN's Reading Tutor to Analyze Development of Children's Oral Reading Prosody," in *Proceedings of the 25th Florida Artificial Intelligence Research Society Conference (FLAIRS-25)*, Marco Island, Florida, 2012.

[23] P. Jia, J. E. Beck, and J. Mostow, "Can a Reading Tutor that Listens use Inter-word Latency to Assess a Student's Reading Ability?," in *Proceedings of the ITS 2002 Workshop on Creating Valid Diagnostic Assessments*, San Sebastian, Spain, 2002, pp. 23-32.

[24] G. Aist and J. Mostow, "Using Automated Within-Subject Invisible Experiments to Test the Effectiveness of Automated Vocabulary Assistance," in *Proceedings of ITS'2000 Workshop on Applying Machine Learning to ITS Design/Construction*, Montreal, Canada, 2000, pp. 4-8.

[25] J. Mostow, "Experience from a Reading Tutor that listens: Evaluation purposes, excuses, and methods," in *Interactive literacy education: facilitating literacy environments through technology*, C. K. Kinzer and L. Verhoeven, Eds. New York: Lawrence Erlbaum Associates, Taylor & Francis Group, 2008, pp. 117-148.

[26] C. Heiner, J. E. Beck, and J. Mostow, "Improving the help selection policy in a Reading Tutor that listens," presented at the Proceedings of the InSTIL/ICALL Symposium on Natural Language Processing and Speech Technologies in Advanced Language Learning Systems, Venice, Italy, 2004.

[27] C. Heiner, J. E. Beck, and J. Mostow, "When do students interrupt help? Effects of time, help type, and individual differences," in *Proceedings of the 12th International Conference on Artificial Intelligence in Education (AIED 2005)*, Amsterdam, 2005, pp. 819-826.

[28] K.-m. Chang, J. Beck, J. Mostow, and A. Corbett, "A Bayes Net Toolkit for Student Modeling in Intelligent Tutoring Systems," presented at the Proceedings of the 8th International Conference on Intelligent Tutoring Systems, Jhongli, Taiwan, 2006.

[29] J. E. Beck, K.-m. Chang, J. Mostow, and A. Corbett, "Does help help? Introducing the Bayesian Evaluation and Assessment methodology," in *9th International Conference on Intelligent Tutoring Systems*, Montreal, 2008, pp. 383-394. ITS2008 Best Paper Award.

[30] J. E. Beck and J. Mostow, "How who should practice: Using learning decomposition to evaluate the efficacy of different types of practice for different types of students [Best Paper Nominee]," in *9th International Conference on Intelligent Tutoring Systems*, Montreal, 2008, pp. 353-362.

[31] J. E. Beck, "Using learning decomposition to analyze student fluency development," in *ITS2006 Educational Data Mining Workshop*, Jhongli, Taiwan, 2006, pp. 21-28.

[32] J. E. Beck, "Does learner control affect learning?," in *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, Los Angeles, CA, 2007, pp. 135-142.

[33] X. Zhang, J. Mostow, and J. E. Beck, "All in the (word) family: Using learning decomposition to estimate transfer between skills in a

Reading Tutor that listens," in *AIED2007 Educational Data Mining Workshop*, Marina del Rey, CA, 2007.

[34] J. M. Leszczenski, "Learning Factors Analysis Learns to Read," Masters Thesis, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 2007.

[35] J. Mostow, G. Aist, C. Huang, B. Junker, R. Kennedy, H. Lan, D. Latimer, R. O'Connor, R. Tassone, B. Tobin, and A. Wierman, "4-Month evaluation of a learner-controlled Reading Tutor that listens," in *The Path of Speech Technologies in Computer Assisted Language Learning: From Research Toward Practice*, V. M. Holland and F. P. Fisher, Eds. New York: Routledge, 2008, pp. 201-219.

[36] J. Mostow, G. Aist, P. Burkhead, A. Corbett, A. Cuneo, S. Eitelman, C. Huang, B. Junker, M. B. Sklar, and B. Tobin, "Evaluation of an automated Reading Tutor that listens: Comparison to human tutoring and classroom instruction," *Journal of Educational Computing Research*, vol. 29, pp. 61-117, December 2003.

[37] J. Mostow, J. Nelson, and J. Beck, "Computer-Guided Oral Reading versus Independent Practice: Comparison of Sustained Silent Reading to an Automated Reading Tutor that Listens," *Journal of Educational Psychology*, under review.

[38] R. Poulsen, P. Wiemer-Hastings, and D. Allbritton, "Tutoring Bilingual Students with an Automated Reading Tutor That Listens," *Journal of Educational Computing Research*, vol. 36, pp. 191-221, 2007.

[39] G. A. Korsah, J. Mostow, M. B. Dias, T. M. Sweet, S. M. Belousov, M. F. Dias, and H. Gong, "Improving Child Literacy in Africa: Experiments with an Automated Reading Tutor," *Information Technologies and International Development*, vol. 6, pp. 1-19, 2010.

[40] F. Weber and K. Bali, "Enhancing ESL Education in India with a Reading Tutor that Listens," presented at the Proceedings of the First ACM Symposium on Computing for Development London, United Kingdom, 2010.

[41] K. Reeder, J. Shapiro, and J. Wakefield, "A computer based reading tutor for young English language learners: recent research on proficiency gains and affective response," in *16th European Conference on Reading and 1st Ibero-American Forum on Literacies*, University of Minho, Campus de Gualtar, Braga, Portugal, 2009.

[42] T. Cunningham, "The Effect of Reading Remediation Software on the Language and Literacy Skill Development of ESL Students," Master's thesis, Department of Human Development and Applied Psychology, University of Toronto, Toronto, Canada, 2006.

[43] J. Cohen, *Statistical power analysis for the behavioral sciences*, 2nd ed. Hillsdale, NJ: Lawrence Erlbaum Associates, 1988.

[44] CMU, "The CMU Sphinx Group open source speech recognition engines [software at http://cmusphinx.sourceforge.net]," ed, 2008.

[45] A. G. Hauptmann, L. L. Chase, and J. Mostow, "Speech Recognition Applied to Reading Assistance for Children: A Baseline Language Model," in *Proceedings of the 3rd European Conference on Speech Communication and Technology (EUROSPEECH93)*, Berlin, 1993, pp. 2255-2258.

[46] J. Mostow, S. F. Roth, A. G. Hauptmann, and M. Kane, "A prototype reading coach that listens [AAAI-94 Outstanding Paper]," in *Proceedings of the Twelfth National Conference on Artificial Intelligence*, Seattle, WA, 1994, pp. 785-792.

[47] G. Aist, P. Chan, X. D. Huang, L. Jiang, R. Kennedy, D. Latimer, J. Mostow, and C. Yeung, "How effective is unsupervised data collection for children's speech recognition?," in *Proceedings of the International Conference on Speech and Language Processing (ICSLP98)*, Sydney, Australia, 1998, p. #929.

[48] M. Eskenazi, "KIDS: A database of children's speech," *Journal of the Acoustic Society of America*, vol. 100, p. 2, December 1996 1996.

[49] M. Eskenazi and J. Mostow, "The CMU KIDS Speech Corpus (LDC97S63)," ed: Linguistic Data Consortium (http://www.ldc.upenn.edu), University of Pennsylvania, 1997.

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

[51] E. A. Betts, *Foundations of Reading Instruction*. New York: American Book Company, 1946.

[52] J. Fogarty, L. Dabbish, D. Steck, and J. Mostow, "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, Eds. San Antonio, Texas: Amsterdam: IOS Press, 2001, pp. 422-433.

[53] J. Mostow, J. Beck, S. V. Winter, S. Wang, and B. Tobin, "Predicting oral reading miscues," in *Proceedings of the Seventh International Conference on Spoken Language Processing (ICSLP-02)*, Denver, CO, 2002, pp. 1221-1224.

[54] S. Banerjee, J. E. Beck, and J. Mostow, "Evaluating the effect of predicting oral reading miscues," in *Proc. 8th European Conference on Speech Communication and Technology (Eurospeech 2003)*, Geneva, Switzerland, 2003, pp. 3165-3168.

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

[56] M. Ravishankar, R. Bisiani, and E. Thayer, "Sub-Vector Clustering to Improve Memory and Speed Performance of Acoustic Likelihood Computation," in *Proceedings of Eurospeech*, Rhodes, Greece, 1997, pp. 151-154.

[57] D. Bansal and M. Ravishankar, "New Features for Confidence Annotation," in *Proceedings of International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998.

[58] S. Banerjee, J. Mostow, J. E. Beck, and W. Tam, "Improving Language Models by Learning from Speech Recognition Errors in a Reading Tutor that Listens," in *Second International Conference on Applied Artificial Intelligence*, Fort Panhala, Kolhapur, India, 2003, pp. 187-193.

[59] T. M. Mitchell, P. E. Utgoff, and R. B. Banerji, "Learning by experimentation: acquiring and refining problem-solving heuristics," in *Machine Learning*, R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Eds. Palo Alto, CA: Tioga, 1983, pp. 163-190.

[60] P. J. Schwanenflugel, A. M. Hamilton, M. R. Kuhn, J. M. Wisenbaker, and S. A. Stahl, "Becoming a fluent reader: Reading skill and prosodic features in the oral reading of young readers," *Journal of Educational Psychology*, vol. 96, pp. 119-129, 2004.

[61] J. Miller and P. J. Schwanenflugel, "Prosody of syntactically complex sentences in the oral reading of young children," *Journal of Educational Psychology*, vol. 98, pp. 839-853, 2006.

[62] P. J. Schwanenflugel, M. R. Kuhn, R. D. Morris, and B. A. Bradley. (2006, November 8). *The Development of Fluent and Automatic Reading: Precursor to Learning from Text*. Available: http://drdc.uchicago.edu/community/project.phtml?projectID=60

[63] J. Miller and P. J. Schwanenflugel, "A longitudinal study of the development of reading prosody as a dimension of oral reading fluency in early elementary school children," *Reading Research Quarterly*, vol. 43, pp. 336–354, 2008.

[64] R. G. Benjamin and P. J. Schwanenflugel, "Text complexity and oral reading prosody in young readers," *Reading Research Quarterly*, vol. 45, pp. 388-404, October/November/December 2010.

[65] M. R. Kuhn, P. J. Schwanenflugel, and E. B. Meisinger, "Aligning Theory and Assessment of Reading Fluency: Automaticity, Prosody, and Definitions of Fluency. Invited review article," *Reading Research Quarterly*, vol. 45, pp. 230–251, Apr-Jun 2010.

[66] J. Mostow and M. Duong, "Automated Assessment of Oral Reading Prosody," in *Proceedings of the 14th International Conference on Artificial Intelligence in Education (AIED2009)*, Brighton, UK, 2009, pp. 189-196.

[67] M. Duong and J. Mostow, "Detecting prosody improvement in oral rereading," in *Online Proceedings of the Second ISCA Workshop on Speech and Language Technology in Education (SLaTE)*, Wroxall Abbey Estate, Warwickshire, England, 2009, p. at http://www.eee.bham.ac.uk/SLaTE2009/.

[68] M. Duong and J. Mostow, "Adapting a Duration Synthesis Model to Score Children's Oral Reading," in *Interspeech 2010*, Makuhari, Japan, 2010, pp. 769-772.