Training a Confidence Measure for a Reading Tutor that Listens

Yik-Cheung Tam, Jack Mostow, Joseph Beck, and Satanjeev Banerjee

Project LISTEN
Robotics Institute, Carnegie Mellon University
Pittsburgh, PA 15213-3890
{yct,mostow,joseph.beck,satanjeev.banerjee}@cs.cmu.edu

Abstract

One issue in a Reading Tutor that listens is to determine which words the student read correctly. We describe a confidence measure that uses a variety of features to estimate the probability that a word was read correctly. We trained two decision tree classifiers. The first classifier tries to fix insertion and substitution errors made by the speech decoder, while the second classifier tries to fix deletion errors. By applying the two classifiers together, we achieved a relative reduction in false alarm rate by 25.89% while holding the miscue detection rate constant.

1. Introduction

In this paper, we describe a confidence measure for Project LISTEN's automated Reading Tutor [1], which uses speech recognition to listen to children read aloud. The purpose of the confidence measure is to estimate the probability that a given text word in a sentence was read correctly by the student.

Confidence measures have been applied to many different speech recognition tasks, such as large vocabulary speech recognition [2], and spontaneous speech recognition [3]. One significant difference between these tasks and the Reading Tutor is that the Reading Tutor knows the text that the children are expected to read. In this domain, we can also exploit information about the student's past performance.

Related work on applying speech recognition in education domains includes an automated pronunciation learning system with confidence measures. Witt *et al.* [4] used forced alignment to obtain the phoneme boundaries of an input speech utterance. It computed a confidence score for each phoneme by normalizing its acoustic score obtained from the forced alignment by its corresponding acoustic score computed from a phone-loop decoder. However, using forced alignment with children's reading is inappropriate because children often jump back to the beginning of the phrase or sentence and reread, or skip hard words.

This paper is organized as follows. Section 2 describes how the Reading Tutor listens currently and how a confidence measures can be applied in the Reading Tutor. Section 3 describes the feature sets used for training the classifiers. Section 4 describes experimental settings of the confidence measure followed by evaluation results in Section 5. Discussion and conclusion are provided in Section 6 and Section 7 respectively.

This work was supported by NSF under IERI Grant REC-9979894. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

2. How does the Reading Tutor listen?

There are two issues that the Reading Tutor needs to address: (1) tracking the student's current position in the sentence and (2) deciding if a word was read correctly. To address the first issue, the Reading Tutor aligns the output hypothesis from the SPHINX [5] decoder against the target sentence. As shown in Figure 1, each word h_i in a hypothesis (H) is aligned to a word w_j in a target sentence (T). As shown in Figure 1, an utterance [1] can start at any word position of the target sentence and jump around in the sentence. Forced alignment against the target sentence ignores this problem. Instead, we use a constrained language model generated from the sentence [6]. Without confidence measures, the Reading Tutor determines if a word in the sentence is read correctly by using the alignment as shown in Figure 2. If some hypothesized word h_i that matches target word w_j (i.e. $h_i = w_j$) is aligned against w_j, the Reading Tutor classifies w_j as read correctly. Therefore the decision is currently a 0-1 hard decision. Instead, we propose to estimate Pr(Word was read correctly|Features) (or $Pr(W|F)$ for short) to give a soft decision between [0,1]. This scheme exploits additional information so as to decide more accurately which words to accept as correct, and provides flexibility to the Reading Tutor by varying a threshold t and accepting w_j as correct if $Pr(W|F) >$ t.

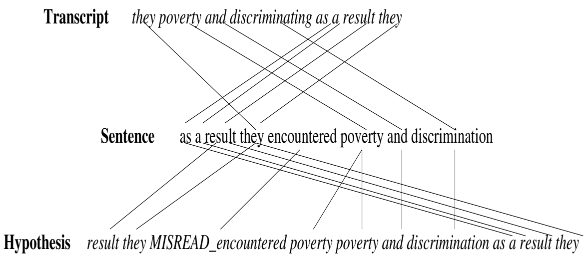

Figure 1: Alignments of a hypothesis (H) against a sentence (S), and a transcript (T) against S.

T:	c	c	c	c	o	c	c	s
S:	as	a	result	they	encountered	poverty	and	discrimination
H:	c	c	c	c	s	c	c	c

Figure 2: Classifications of words from sentence (S) based on alignments against hypothesis (H) and transcript (T) in Figure 1 (where c=correct, o=omission, s=substitution)

[1] A student can attempt all or part of a sentence more than once. Each attempt is recorded as an utterance.

3. Features used

We investigate three kinds of features to estimate the confidence probabilities. The first kind of features are obtained from a speech decoder (decoder-based features). The second kind of features are derived from an alignment as shown in Figures[1,2] (alignment-based features). The third kind of features are extracted from the student's previous reading (history-based features).

3.1. Decoder-based Features

Decoder-based features are computed at the word level from the output of the SPHINX decoder [5]. They consist of the log energy normalized by number of frames, acoustic score normalized by number of frames, language model score, lattice density, averaged phone perplexity [2] and duration.

3.2. Alignment-based Features

Alignment features are computed from the alignments of the hypothesis H against the target sentence S. To help describe the alignment features, we define the following symbols:

$\{h_1, h_2, ..., h_i, ..., h_H\}$ denotes the hypothesis H.

$\{w_1, w_2, ..., w_j, ..., w_T\}$ denotes the target sentence S.

i,j >0 denote word position in H and S respectively.

$\phi_c(w_j)$ denotes a set of h_i that are aligned to w_j s.t. $h_i = w_j$.
(In Figure 1, $\phi_c(w_6 = poverty)$ equals $\{h_4$=poverty, h_5=poverty$\}$)

$\phi_w(w_j)$ denotes a set of h_i that are aligned to w_j s.t. $h_i \neq w_j$.
(In Figure 1, $\phi_w(w_5 = encountered)$ equals $\{h_3 = MISREAD\_encountered\}$)

$S(h_i)$ denotes an aligned sentence position of h_i.
(In Figure 1, $S(h_7 = discrimination)$ = 8 because h_7 aligns against w_8).

$LC(h_i)$ is the left context of h_i, defined as the number of successive words read correctly [3] in a sentence S before the hypothesized word h_i.
(In Figure 1, $LC(h_7 = discrimination)$ = 2 because $h_6 = w_7 =$ "and" and $h_5 = w_6 =$ "poverty" but $h_4 \neq w_5$.)

$RC(h_i)$ is the right context of h_i, defined as the number of successive words read correctly in a sentence S after the hypothesized word h_i.
(In Figure 1, $RC(h_5 = poverty)$ = 2 because $h_6 = w_7 =$ "and" and $h_7 = w_8 =$ "discrimination".)

Alignment-based features of a target word w_j are:

- 0-1 indicator I(w_j) that w_j is read correctly, e.g. I(w_6="poverty") = 1 since $h_4 = w_6$.
- the number of hypothesized words h_i aligned against w_j where h_i is equal to w_j divided by the total number of h_i aligned against w_j:

$$\frac{|\phi_c(w_j)|}{|\phi_c(w_j)| + |\phi_w(w_j)|} \quad (1)$$

- averaged jump distance between successive hypothesized words in a sentence:

$$\frac{\sum_{h_i \in \phi_c(w_j)} \frac{(j - S(h_{i-1}))}{|T|}}{|\phi_c(w_j)|} \quad (2)$$

  This feature measures how much the student's reading jumps around a target sentence T.

- averaged difference between a hypothesized word position and a target word position:

$$\frac{\sum_{i, h_i \in \phi_c(w_j)} i - j}{|\phi_c(w_j)|} \quad (3)$$

- length of the left context of w_j in a sentence which is defined below. There are two possible cases:

  Case 1: $\phi_c(w_j)$ is non-empty:

$$max_{h_i \in \phi_c(w_j)}\{LC(h_i)\} \quad (4)$$

  Case 2: $\phi_c(w_j)$ is empty. This happens when the speech decoder does not recognize the word w_j.

$$max_{h_i \in \phi_c(w_{j-1})}\{LC(h_i) + 1\} \quad (5)$$

  If w_j is the first word of a sentence (i.e. w_1), or $\phi_c(w_{j-1})$ is empty, the feature value is zero.

- length of the right context of w_j in a sentence. The feature is computed analogously to the left-context feature described above.
- inter-word latency [7] (time between the end of (the hypothesis word h_i aligned against) the previous text word w_{j-1} and the start of (the hypothesis word h_i' aligned against) the current text word w_j (i.e., $S(h_i) + 1 = S(h_{i'})$).

3.3. History-based Features

These features are computed from the student's history in the Reading Tutor, including all of the student's recorded utterances, not just those that were transcribed.

The features are divided into word-level and utterance-level features. The historical word-level features for each w_j in a sentence include:

- The number of times the word was encountered in distinct sentences (i.e. do not count rereading a sentence)
- The number of times the word was accepted
- We compute the following features for all words, just for Dolch (high-frequency) words [8], and just for non-Dolch words:
  - The average of the lower-bound estimates of inter-word latency [7]
  - The average of the upper-bound estimates of inter-word latency

The utterance-level features for each w_j in a sentence include:

- The number of attempts the student has made to read this sentence
- Average number of sentence words the student attempts per utterance

[2] Averaged phone perplexity measures how well the acoustic observations from a word segment discriminate across different phonemes.

[3] For brevity, whenever "read correctly" refers to words in a hypothesis, we leave implicit the caveat "according to the recognizer."

- Average number of sentence words the student reads correctly per utterance
- Average number of words the student reads correctly (e.g. for the sentence "The cat in the hat," if the student says "the cat...the cat in the hat," then the total number of words would be 7)
- Average number of jumps (reading a word other than the next word or current word)
- Average number of times the student regresses to the first word in the sentence

4. Experimental Setup

4.1. Preparation of data sets

Children's speech data were collected from the field where the Reading Tutor has been deployed in elementary schools in Pittsburgh. Our speech data analyst transcribed speech data and we split it into training and test sets of 3714 and 1883 utterances respectively. Speakers in the training and test sets do not overlap. Alignments of a hypothesis and a transcript in Figure 2 create a 3x3 confusion matrix which describes the partition of data over the 9 possible cells as shown in Table 1. Table 2 and Table 3 show the partitions of training and test data. Each word is further categorized into content words and function words [6] (e.g. *a*, *an*, *the*). Function words do not carry much of a sentence's meaning; therefore, we focus on assessing a student's ability to read content words.

Table 1: 3x3 confusion matrix generated using text-space alignments of target sentence (S) against transcription (T) and hypothesis (H).

	H Correct	H Omission	H Substitution
T Correct	Cell 1	Cell 2	Cell 3
T Omission	Cell 4	Cell 5	Cell 6
T Substitution	Cell 7	Cell 8	Cell 9

Table 2: Training data distribution of content and function words (counts of function words are shown in parentheses).

	H Correct	H Omission	H Substitution
T Correct	9551 (4897)	392 (216)	290 (35)
T Omission	2117 (1289)	13402 (6421)	383 (45)
T Substitution	557 (243)	224 (136)	102 (21)

Table 3: Test data distribution of content, and function words (counts of function words are shown in parentheses).

	H Correct	H Omission	H Substitution
T Correct	5621 (3044)	195 (109)	165 (13)
T Omission	746 (549)	5844 (3042)	122 (19)
T Substitution	215 (85)	80 (56)	40 (6)

4.2. Classifier to estimate $Pr(W|F)$

We used WEKA [9] to train a decision tree to estimate $Pr(W|F)$. WEKA uses a maximum information gain criterion to grow the decision trees. At each leaf of a decision tree, $Pr(W|F)$ can be computed as the relative frequency of the labeled data assigned to the current leaf. During training, we used 10-fold cross-validation on the training data.

We trained two decision trees. The first classifier estimates confidence probabilities of words in the first and third columns of the confusion matrix in Table 1. The columns contain SPHINX's insertion errors (cells 4,6) of words omitted by the reader according to an alignment of a transcript against a sentence. Moreover, cell 7 (3) represents substitution errors in which the words are misread but SPHINX says the words are correct (or vice versa). The first classifier uses decoder-based and alignment-based features.

The second classifier is used to train and estimate confidence probabilities of words on the second column of the confusion matrix in Table 1. It tries to correct the deletion errors (cell 2) made by the speech decoder. There are no decoder-based features for deleted words, only history-based features plus the left and right context features of the alignment-based features.

4.3. Performance Metrices

To simplify analysis, we reduce a 3-class classification problem into a 2-class classification problem by defining the data labeled "omission" and "substitution" as "miscue". In this case, columns 2 and 3 are combined into a single column, and rows 2 and 3 are combined into a single row in Table 1.

Performance of a Reading Tutor is evaluated using the false alarm rate (FA) and miscue detection rate (MD) which are defined as follows:

$$MD = \frac{100 \cdot N_c(cells\{5,6,8,9\})}{N_c(rows\{2,3\}) + N_f(row\{2,3\})}$$
$$FA = \frac{100 \cdot N_c(cells\{2,3\})}{N_c(row1) + N_f(row1)}$$

where N_c and N_f represents the counts of content words and function words. The Reading Tutor ignores miscues on function words, so the numerators omit N_f. FA measures hearing correct student reading as incorrect; MD measures the ability to hear students' mistakes and omissions caused by skipping words or (more often) attempting only part of the sentence within the time interval covered by the utterance. The false alarm rate and miscue detection rate of the baseline (without a confidence measure) on the test set are 3.94% and 56.33% respectively.

5. Experimental Results

Figure 3 shows a Receiver Operating Characteristics (ROC) curve that analyzes the tradeoff between the correct acceptance rate (=100% - false alarm rate) and the miscue detection rate.

5.1. Results using both Classifiers

We used estimated confidence probabilities from both classifiers to plot the ROC curve as shown in Figure 3. It shows that when the threshold is between [0.20, 0.35], the confidence measure improves the false alarm and miscue detection rates relative to the baseline. We observe that given the same false alarm rate as the baseline, the confidence measures improves the miscue detection rate relatively by 4.1% (from 56.33% to 58.64%). Moreover, given the same miscue detection rate as the baseline, the confidence measures reduces the false alarm rate relatively by 25.89% (from 3.94% to 2.92%).

5.2. Performance of Individual Classifier

From Table 4, we can see that at the same FA rate as the baseline, the first classifier (which attempts to fix insertion and substitution errors described in Section 4.2) improves miscue detection by 357 (348+9) words respectively, while the second classifier (which attempts to fix deletion errors described in Section 4.2) reduces the false alarms (-17) to compensate for the increase in false alarms (+17) made by the first classifier. On the other hand, from Table 5, we can see that at the same MD rate as the baseline, both classifiers reduce false alarms by 56 and 37 words respectively. The first classifier also improves detection of miscues (+240-14) to compensate for the loss made by the second classifier (-221-5).

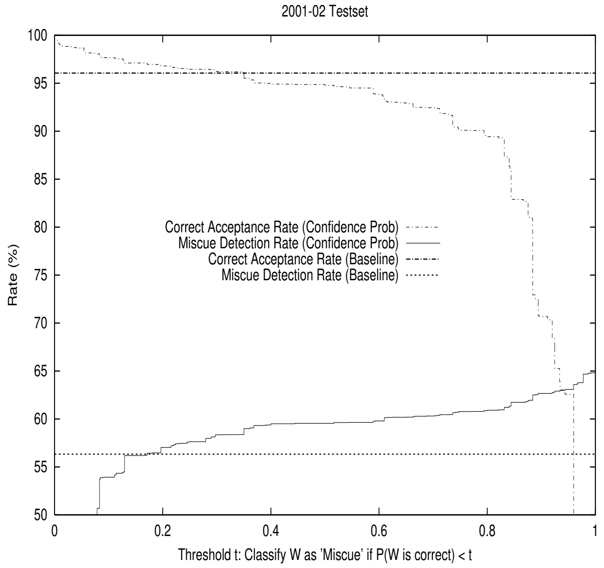

Figure 3: ROC curve of the test set using estimated probabilities of both classifiers.

Table 4: Change of distribution of content-word data of the test set (relative to Table 3) after applying classifier A and B given fixed false alarm rate of 3.94% (threshold=0.3502).

	H Correct	H Omission	H Substitution
T Correct	0	-17	+17
T Omission	-241	-107	+348
T Substitution	-8	-1	+9

Table 5: Change of distribution of content-word data of the test set (relative to Table 3) after applying classifier A and B given fixed miscue detection rate of 56.33% (threshold=0.1706)

	H Correct	H Omission	H Substitution
T Correct	+93	-37	-56
T Omission	-19	-221	+240
T Substitution	+19	-5	-14

6. Discussion: Which features are informative?

It is interesting to know which subset of features are most informative in terms of classification. When decision-tree learners are employed, questions located near the root node of the tree reflect the informativeness of features in terms of maximizing the information gain. In the first classifier (which attempts to fix insertion and substitution errors described in Section 4.2), averaged phone perplexity is used at the first level of the tree while log energy and the alignment indicator are applied at the second level. In the second classifier (which attempts to fix deletion errors described in Section 4.2), length of the left context is applied at the first level of the tree while length of the right context is applied at the second level. Contextual information may help "guess" when a word is read by the child but the speech decoder does not hear it.

7. Conclusions and Future Work

We successfully developed a confidence measure for an automated Reading Tutor that listens. We estimated confidence probabilities using two decision tree learners. The first learner addresses insertion and substitution errors while the second learner tries to correct deletion errors made by the speech decoder. Compared to the baseline, the confidence measure achieves a relative reduction of false alarm rate by 25.89% at the same miscue detection rate as the baseline in the test set. Moreover, it improves the miscue detection rate by 4.1% (relative) at the same false alarm rate as the baseline in the test set. In addition, there exists a range of thresholds in the ROC of the test set such that both the false alarm rate and miscue detection rates of the Reading Tutor with confidence measure are better than the baseline results. In future work, we would like to further explore the application of history-based features.

8. References

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

[2] F. Wessel, R. Schlter, K. Macherey, and Hermann Ney, "Confidence measures for large vocabulary continuous speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 9, no. 3, pp. 288–298, 2001.

[3] T. Schaaf and T. Kemp, "Confidence measures for spontaneous speech recognition," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 875–878, 1997.

[4] Silke Maren Witt, *Use of Speech Recognition in computer assisted language learning*, Ph.D. thesis, University of Cambridge (ftp://svr-ftp.eng.cam.ac.uk/pub/reports/witt_thesis.ps.gz), 1999.

[5] X. Huang et al., "The SPHINX-II Speech Recognition System: An Overview," *Journal of Computer Speech and Language*, vol. 7, no. 2, pp. 137–148, 1993.

[6] J. Mostow, S. Roth, A. G. Hauptmann, and M. Kane, "A Prototype Reading Coach that Listens [AAAI-94 Outstanding Paper Award]," in *Proceedings of the Twelfth National Conference on Artificial Intelligence*, pp. 785–792, 1994. Seattle, WA: American Association for Artificial Intelligence.

[7] 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)*, pp. 355–361, 1997. Providence, RI: American Association for Artificial Intelligence.

[8] E. Dolch, "A basic sight vocabulary," *Elementary School Journal*, vol. 36, pp. 456–460, 1936.

[9] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*, Morgan Kaufmann, 1999.