Mining Free-form Spoken Responses to Tutor Prompts

Xiaonan Zhang[1], Jack Mostow[1], Nell Duke[2], Christina Trotochaud[3], Joseph Valeri[1], Al Corbett[1]
{xiaonanz, mostow, jmv, corbett}@cs.cmu.edu, nkduke@msu.edu, TrotochaudC@glps.k12.mi.us
[1]Project LISTEN, School of Computer Science, Carnegie Mellon University
[2]Literacy Achievement Research Center, Michigan State University
[3]Grand Ledge Public Schools, Grand Ledge, Michigan

Abstract. How can an automated tutor assess children's spoken responses despite imperfect speech recognition? We address this challenge in the context of tutoring children in explicit strategies for reading comprehension. We report initial progress on collecting, annotating, and mining their spoken responses. Collection and annotation yield authentic but sparse data, which we use to synthesize additional realistic data. We train and evaluate a classifier to estimate the probability that a response mentions a given target.

1 Introduction

Speech is the easiest, most natural way for students to respond to tutors. Speech is faster than typing on a keyboard and more expressive than clicking on a menu item. Speech is especially useful for young children because they type slowly and spell poorly. Ideally an intelligent tutor for children would understand their spoken responses to its prompts. Unfortunately, current technology for speech recognition and language understanding has poor accuracy – especially for children's spontaneous speech, which can be difficult even for adults to understand. An intelligent tutor that relies on accurate transcription and interpretation of children's unconstrained speech appears infeasible for years to come. Consequently, the rare intelligent tutors that recognize children's speech constrain it. For example, Project LISTEN's Reading Tutor operates on oral reading of a known text [1].

Thus methods for intelligent tutors to respond effectively to children's unconstrained speech despite imperfect speech understanding could be very useful. We report here on progress toward this goal in the context of a project to teach children explicit strategies for reading comprehension. This project is extending Project LISTEN's Reading Tutor, which listens to children read aloud, so that it also listens to children *think* aloud. This work builds on previous ideas for word spotting (e.g. [2]), confidence annotation in spoken dialogue systems (e.g. [3]), and generating synthetic data (e.g. [4]). This endeavor is relevant to educational data mining in a number of ways. We describe an efficient way to collect authentic student responses with expert tutorial labels. We show how to augment sparse training data by using it to generate realistic synthetic data. Finally, we present empirical evaluations of classifiers trained on this data.

Acknowledgments: The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305B070458 to Carnegie Mellon University, by the National Science Foundation under ITR/IERI Grant No. REC-0326153, and by the Heinz Endowments. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute, the U.S. Department of Education, the National Science Foundation, or the Heinz Endowments. We also thank the educators, students, and LISTENers who helped generate and analyze our data, in particular Ravi Mosur for developing Sphinx-II's acoustic confidence measure.

2 Collecting authentic student and tutor responses

To mine children's spoken responses to automated tutoring on comprehension strategies, we must first collect a goodly amount of data. An obvious approach is to record children's responses to a human tutor. However, this approach suffers from two shortcomings. First, the process is labor-intensive. Second, children respond differently to a human tutor than to an automated tutor, for example because they have a different social relationship to an adult than to a computer. This difference is problematic because our purpose is to enable an automated tutor to assess children's spoken responses. A Wizard of Oz simulation mitigates the social relationship issue, but not the labor-intensiveness.

Instead, we took a different approach. We extended the Reading Tutor by implementing comprehension strategy instruction that prompts and records spoken responses without analyzing them. Children use the Reading Tutor simultaneously on multiple computers, which can record them at the same time, unlike a human tutor or Wizard of Oz limited to tutoring one child at a time. The spoken responses are logged to a database and transcribed by hand.

The instruction itself is scripted by an expert reading researcher and practitioner based on their experience in teaching comprehension strategies to children. They carefully select texts conducive to tutoring particular strategies. Tutorial sequences in the instruction consist of a few basic step types: assisted oral reading by the student; reading aloud by the tutor to the student; multiple choice questions answered by clicking on a menu item; short-answer fill-in questions answered by keyboard input; and prompts for free-form spoken responses.

Our reading experts' involvement did not end with scripting instruction. Once 299 spoken responses to 33 prompts were recorded and transcribed, our expert practitioner annotated each student utterance with how she would have replied to it, and why. An example of a short-answer prompt was *What do you think the fifth sense is?* The expected answer is *touching*. The expert's recommendation for the student responses *touching*, *seeing*, or *use our nose to smell* was to say nothing in reply, since "This question is a "what do you think" question," so any reasonable answer is acceptable. In contrast, the recommended reply to the response *uh, apples?* was *Think about the senses that we already talked about in this text…try again*, because "Apples is not a sense."

Thus our data consists of tutorial prompts, transcribed spoken responses to them, expert annotations that recommend how to reply, and rationales for those recommendations in terms of features of the student responses. The purpose of this data is to train a decision function that uses those features to classify future responses by how the tutor should reply to them. In the examples above, the recommended reply to the student's spoken response depends on whether it mentions a target concept. This type of decision problem is a simple but useful case of the general problem of classifying responses by how the tutor should reply, and applies to many of our expert's annotations. The remainder of this paper focuses on this problem.

3 Detecting a target

We first address how to determine whether a spoken response mentions a given target, which for now we define as a word or phrase plus its variant forms. For example, our target for *touch* includes the forms *touches* and *touching*. We include variant forms because we care if the response mentions the concept, but not which specific word it uses. For the same reason, we plan in future to include synonyms for the context-appropriate word meaning. For example, *feel* and *feeling* are synonyms for *touch* as a sense, but not for the colloquial meaning of *touch* as "ask for money." We also care about our confidence in whether a response mentions a given concept. More precisely, our challenge is to learn the probability that a given response mentions a given target.

3.1 Stretching sparse training data

A difficult learning task requires as much training data as possible. A training example for our task includes an utterance, a target, and a label classifying the utterance by whether it mentions the target. To expand our limited set of authentic data, we generate a much larger set of synthetic data – in fact, so much larger that we hold out authentic data to use for testing. This held-out set consists of the 64 recorded responses to 5 questions where the expected target is clear, e.g. *What do you think the fifth sense is? (touch)*.

Each authentic datum is an annotated utterance labeled as a positive or negative example of mentioning an expected target concept. For example, the utterances *touching* and *uh, apples?* serve respectively as authentic positive and negative examples for the target concept *touch*. We have only a limited amount of such data. To generate a large set of training examples, we reuse the transcribed free-form responses many times as synthetic positive and negative examples of mentioning other concepts. These 471 utterances include 172 free-form responses previously recorded and transcribed but not annotated. The idea is to pretend that each utterance has a different target that it does or does not mention. Thus each utterance generates multiple training examples, one for each hypothetical target. The utterances in the synthetic data are actually authentic; only their labels are not. Thus the utterances *touching* and *uh, apples?* also serve as synthetic examples of mentioning (or not mentioning) hypothetical targets, such as *butterfly*. As this example suggests, the synthetic data is heavily skewed toward negative examples.

As targets for the synthetic data we use the 21 words that occur more than 10 times in the transcribed responses, such as *butterfly*, and include their variants, such as *butterflies*. We exclude the most frequent 100 words of English, such as *the*, because they might differ systematically from authentic target words in how they are spoken. For example, function words tend to have reduced pronunciations. The resulting synthetic data set has $471 \times 21 = 9891$ examples.

3.2 Configuring the speech recognizer

To decide whether an utterance contains a given target, an obvious solution is to use automatic speech recognition (ASR) to decode the utterance, and see if the ASR output contains any of the target words. However, children's free-form speech is too

unpredictable for ASR to transcribe accurately, in contrast to oral reading of known text. How can we configure the ASR to increase its accuracy on this target-spotting task?

A key point here is that if we care only about a specific target, we do not need to know what else the student said. We therefore configure the ASR to listen only for the target words and to insert arbitrary phoneme sequences to model other words. We penalize such insertions to make the ASR prefer target words unless they match the speech poorly.

If this configuration detected the target perfectly, our problem would be solved. However, there are still many cases where the ASR errs. Fortunately, our ASR (http://sourceforge.net/projects/cmusphinx) reports not only which words are recognized, but also an acoustic confidence score for each recognized word. We compute the acoustic confidence for a target concept, e.g., *touch*, as the maximum score in the ASR output of any of the target words, e.g., *touch, touches, touching*. To decide whether the utterance mentions the target, the tutor can test whether this score exceeds some threshold that determines the tradeoff between false negatives and false positives. But can it do better?

3.3 Using a logistic regression model to combine various evidence

The simple acoustic confidence threshold model ignores some relevant factors. A single threshold may not be appropriate for different targets. For example, the larger the set of words or phrases comprising the target, the higher their maximum confidence score may tend to be. If longer words or phrases tend to score lower than shorter ones, the threshold should decrease with target length. Conversely, the longer the utterance, the likelier that it will randomly contain a good match to the target, so the threshold should increase with utterance length. To take these factors into account, we use predictors derived from the utterance and ASR output and listed in Table 1.

Table 1: Predictors used in the logistic regression model

#	Predictor	Description
1	MaxConf	Maximum confidence score of all target words
2	TargetSize	Number of target words (words that belong to the target)
3	WordLen	# of letters in the top-scored target word; if none, 0; if there's a tie, their average
4	HypLen	Length of the ASR output, measured by number of words
5	UttDur	Duration of the utterance, measured by the size of its audio file in kilobytes

To combine this information, we use binomial (or binary) logistic regression, which estimates the probability of an event Y as a logistic function of a set of input predictors $X_1, X_2, \ldots, X_n$. In our case, $Y = 1$ iff a target occurs in an utterance, and $X_1, \ldots, X_5$ are the five predictor variables in Table 1. The logit (*i.e.*, the logarithm of the odds) of the target occurring is modeled as a linear function of the X_i, as shown in Equation 1:

$$\ln\left(\frac{\Pr(occur)}{1-\Pr(occur)}\right) = \beta_0 + \beta_1 * MaxConf + \beta_2 * TargetSize + \beta_3 * WordLen + \beta_4 * HypLen + \beta_5 * UttDur \quad \textbf{(1)}$$

Here Pr(*occur*) is the probability that the target occurs in the utterance, β_0 is the intercept, and β_1, …, β_5 are the respective regression coefficients for the predictors in Table 1. The regression coefficient for each predictor describes the change in the logit associated with a unit change in that predictor. A positive (negative) β means that an increase in the predictor will increase (decrease) the probability of the outcome. To make different β's comparable, we first normalize the input predictors to range from 0 to 1, so that the absolute value of β measures the impact of that predictor compared to the others. Given Pr(*occur*) for a target, we decide whether the target occurs by comparing Pr(*occur*) to a threshold, e.g. 0.5. We decide yes if it's larger than the threshold, otherwise no.

We use a logistic regression model for several reasons. First, it's compact to represent, fast to compute, and easy to interpret. Second, unlike linear regression it does not assume normally distributed variables. Third, rather than a binary judgment as to whether the target occurs, it outputs a probability that a tutor could use to decide more judiciously which feedback to provide. For example, if the tutor thinks the student said the target but is not very confident, it should hedge its reply rather than praise an answer that may well be wrong. Finally, logistic regression outperformed the alternatives we compared it to. In cross-validation tests, it achieved higher precision, recall, and AUC (described in Section 4) than a Naïve Bayes classifier or a J48 decision tree.

We used Weka 3.5.7 (from weka.sourceforge.net) to train the logistic regression model on the 9891 synthetic examples. As noted earlier, the class distribution on synthetic training data is skewed, with 9547 negative examples but only 344 positive examples for the 21 targets defined. In contrast, the 64 held-out authentic utterances are more balanced, comprising 30 positive instances and 34 negative instances. Differences in class distribution between training data and test data can hurt classifier performance, for instance by biasing the classifier against a class rare in the training set but common in the test set. To address this problem, we used Weka's cost-sensitive classification mechanism to balance the training data, so that its distribution of positive and negative instances resembles the distribution on authentic data. Table 2 shows the resulting β parameter estimates for our five predictors.

Table 2: Parameter estimates of the logistic regression model

Predictor	MaxConf	TargetSize	WordLen	HypLen	UttDur
β **value**	9.8659	0.5802	1.5780	2.7986	0.2606

As Table 2 shows, all predictors are positively correlated with the odds that the target occurs, but acoustic confidence is the strongest predictor. Although one might expect long responses to be likelier to contain the target than short responses, the UttDur predictor is very weak, probably because we measured it by the size of the audio

recording. This recording includes the tutor prompt in the background, so its size reflects the combined duration of the prompt and the student's utterance.

4 Evaluation

We tested our logistic regression model on both synthetic and authentic data. We used 10-fold cross validation on the synthetic training data. We also evaluated the model on the 64 authentically labeled utterances we used as held-out test data. We compared against a majority class baseline model, which simply predicts the most common class for all instances. Table 3 compares the model performance on both data sets.

We evaluate the classifiers on several metrics. Overall accuracy is the fraction of cases classified correctly, i.e. (# TP (true positive) + # TN (true negative)) / # total cases, so it reflects the class distribution. The TP rate, also called sensitivity or recall, is the fraction # TP / (# TP + # FN) of actual positive cases correctly classified as positive. The FP (false positive) rate is the fraction # FP / (# TN + # FP) of actual negative cases misclassified as positive. Its complement, called specificity, measures what fraction of actual negative cases is classified correctly as negative. All these metrics depend on the probability threshold for classifying a case as positive – namely 0.5 for our model.

Cross validation of the majority class baseline shows very high accuracy and zero FP rate because the synthetic data is highly skewed toward negative examples; its accuracy is much lower on the authentic data. More importantly, such a classifier is useless because it cannot detect any mention of the target: its TP Rate is 0. In contrast, the logistic regression model is much more sensitive to positive examples.

Table 3: Model performance under different testing options

Testing method	Classifier	Accuracy	TP Rate	FP Rate	AUC
10-fold cross validation	Majority class	96.52 %	0	0	0.496
	Logistic	80.15 %	0.765	0.197	0.867
Test on authentic data	Majority class	54.67 %	0	0	0.5
	Logistic	75.00 %	0.552	0.086	0.796

In practice, for the probabilistic output of logistic regression model Pr(*occur*) to be useful, we need to turn the probabilities into discrete decisions so as to provide tutorial feedback accordingly. For example, if the tutor is very sure that the target didn't occur, it should give corrective feedback; but if it's not sure, then a hedged reply is probably preferable. With this intuition, we decide on a preliminary division of Pr(*occur*) into 3 disjoint regions, based on two threshold values t_h and t_l ($0 < t_l < t_h < 1$):

- Yes: confident that the target occurred in the utterance ($\Pr(occur) \geq t_h$);
- No: confident that the target didn't occur in the utterance ($\Pr(occur) \leq t_l$);
- Unsure: neither ($t_l < \Pr(occur) < t_h$).

These thresholds control the tradeoff between coverage and precision. The higher the value of t_h, the fewer Yes decisions the tutor will make, but the more confident it can be of these decisions (assuming we have a reasonable model). On the other hand, the tutor will hedge more of its feedback, presumably making it less helpful to students.

To describe this tradeoff, Table 4 shows model coverage and precision on the set of 64 authentic responses for various threshold values. In the table, *Pr(Yes)* and *Pr(No)* mean the probability of outputting a Yes and a No decision, respectively. Precision is the proportion of Yes (No) decisions that are in fact correct, i.e., positive (negative) examples. For example, with high_threshold = 0.9 the tutor will decide only about 14% of the time that Yes, the student mentioned the target – but roughly 89% of these decisions will be correct. By dropping this threshold to 0.5, it can decide Yes more than twice as often – almost 30% of responses – and still be right about 84% of them.

Table 4: Model coverage and precision with different threshold values

Deciding Yes			Deciding No		
high_threshold	**Pr(Yes)**	**Precision**	**low_threshold**	**Pr(No)**	**Precision**
0.5	0.2969	0.8421	0.5	0.7031	0.7111
0.6	0.2500	0.8750	0.4	0.5938	0.7105
0.7	0.1875	0.8333	0.3	0.4688	0.7667
0.8	0.1719	0.8182	0.2	0.2500	0.8125
0.9	0.1406	0.8889	0.1	0.1094	1.0000

Table 4 provides guidance both about where to set the threshold values, and about how definitively to phrase tutor feedback. For example, it indicates that precision for Yes decisions is roughly the same (81%-89%) for thresholds from 0.5 to 0.9, so the tutor may as well set high_threshold at 0.5 (possibly even lower) in order to decide Yes more often, but its feedback must reflect that the student response probably contains the target but may well not. For example, the tutor might refrain from confirming the answer as correct, but still treat it as correct in updating its student model. In contrast, precision for No decisions is much more sensitive, ranging from 71% to 100% as low_threshold varies from 0.5 down to 0.1 – but with coverage ranging from over 70% to below 11%. So the tradeoff between coverage and precision differs for the No case. If our authentic training data is representative, setting low_threshold to 0.1 will avoid any false rejections, allowing definitively phrased corrective feedback. However, at this threshold value, the tutor will decide No less than 11% of the time, even though the target will be absent about half the time. On the other hand, a value of 0.5 will let the tutor decide No for 70% of student responses, but only 71% of these decisions will be correct. In this case, tutor feedback must be phrased to avoid characterizing the student response as wrong.

5 Contributions and future work

This paper formulates the general problem of extracting reliable, tutorially useful information from children's free-form spoken responses despite imperfect speech recognition, so as to assess their comprehension and select appropriate tutor feedback.

We focus on the simpler but common and useful case of estimating the probability that an utterance mentions a given target concept.

We describe efficient methods to collect authentic student data labeled by expert tutors, and to expand it into a much larger set of synthetic yet realistic data. We present a logistical regression model to estimate the probability of a target by combining features of the target and utterance with the acoustic confidence output by a speech recognizer. We cross-validate the accuracy of the resulting probability estimates on synthetic data, and evaluate it on a smaller held-out set of authentic data.

Concept mention is just one useful feature for tutors to detect. We need to extend it to handle synonyms, but we have already extended it (in work omitted here to save space) from the single-target problem addressed in this paper to the multiple-target problem of deciding whether an utterance mentions any, all, or none of *N* given targets. Another useful feature is the distinction between confident and tentative responses [6, 7]. Other distinctions in our expert tutor's annotations include correct vs. incorrect, vague vs. detailed, and answered easily vs. with difficulty. Future work includes using these distinctions to update student models and guide tutor decisions.

References

[1] Mostow, J. Is ASR accurate enough for automated reading tutors, and how can we tell? *Proceedings of the Ninth International Conference on Spoken Language Processing (Interspeech — ICSLP)*, 837-840. 2006. Pittsburgh, PA: International Speech Communication Association.

[2] Wilpon, J.G., L.R. Rabiner, C.H. Lee, and E.R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden Markov models. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1990. 38(11): p. 1870-1878.

[3] San-Segundo, R., B. Pellom, K. Hacioglu, W. W., and J.M.A. Pardo. Confidence measures for spoken dialogue systems. *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, p. 393-396. 2001.

[4] McCandless, M. Word Rejection for a Literacy Tutor. In *Department of Electrical and Computer Engineering*. 1992, Massachusetts Institute of Technology: Cambridge, MA.

[5] Duke, N.K. and P.D. Pearson. Effective Practices for Developing Reading Comprehension. In A.E. Farstrup and S.J. Samuels, Editors, *What Research Has To Say about Reading Instruction*, p. 205-242. International Reading Association: Newark, DE, 2002.

[6] Pon-Barry, H., K. Schultz, E.O. Bratt, B. Clark, and S. Peters. Responding to Student Uncertainty in Spoken Tutorial Dialogue Systems. *International Journal of Artificial Intelligence in Education*, 2006. 16(2): p. 171-194.

[7] Forbes-Riley, K., D. Litman, and M. Rotaru. Responding to Student Uncertainty during Computer Tutoring: A Preliminary Evaluation. *Proceedings of the 9th International Conference on Intelligent Tutoring Systems (ITS)*. 2008. Montreal, Canada.