Recognizing Young Readers’ Spoken Questions

Wei Chen and Jack Mostow, Project LISTEN, School of Computer Science, Carnegie Mellon University, USA

and Gregory Aist, Communication Studies and Applied Linguistics, Department of English, Iowa State University, USA

Abstract. Free-form spoken input would be the easiest and most natural way for young children to communicate to an intelligent tutoring system. However, achieving such a capability poses a challenge both to instruction design and to automatic speech recognition. To address the difficulties of accepting such input, we adopt the framework of predictable response training, which aims at simultaneously achieving linguistic predictability and educational utility. We design instruction in this framework to teach children the reading comprehension strategy of self-questioning. To filter out some misrecognized speech, we combine acoustic confidence with language modeling techniques that exploit the predictability of the elicited responses. Compared to a baseline that does neither, this approach performs significantly better in concept recall (47% vs. 28%) and precision (61% vs. 59%) on 250 unseen utterances from 34 previously unseen speakers. We conclude with some design implications for future speech enabled tutoring systems.

Keywords. Intelligent tutoring system, spoken dialogue system, language modeling, question generation, predictable response training, children’s reading, comprehension strategy instruction, Project LISTEN’s Reading Tutor.

This article revises and extends our ITS2010 paper (Chen, Mostow, & Aist, 2010). The results and analyses it presents are new.

INTRODUCTION

Speech is a natural way for humans to communicate. Intelligent tutoring system developers have started to treat automatic speech recognition (ASR) as a desirable way to enhance human-computer interaction, especially in a learning environment (Hagen, Pellom, Vuuren, & Cole, 2004; Litman & Silliman, 2004; Meron, Valente, & Johnson, 2007; Pon-Barry, Schultz, Bratt, Clark, & Peters, 2006). Compared to typing (Wijekumar & Meyer, 2006), verbal input is especially convenient for children in the early years of elementary schools (i.e., first and second grades, roughly ages 6–7). Unlike older students, young children have trouble typing accurately or quickly. An easy-to-use talking-menu interface can remind children of answers they do not recall on their own, but recognize upon seeing and hearing them. However, answering out loud is faster, easier, and more natural (especially in response to a spoken question) than incurring the temporal, cognitive, and motor costs of seeing and hearing a list of options, deciding which one to choose, and clicking on it. Spoken responses are especially useful when it is important for the child to recall or generate an acceptable answer, not just recognize one. Moreover, multiple-choice input is very constraining, in contrast to a free-form spoken response.

However, accepting children’s free-form speech poses multiple challenges for an automated tutor. First, the flexibility of free-form speech compared to menu-based input is a double-edged sword,
because it allows such a broad range of responses. It may invite off-task speech that a menu does not. Compared to the task of recognizing which menu item to select, the task of recalling or constructing a response to a tutor prompt is more open-ended and cognitively harder. Moreover, children are creative in syntactic and lexical use of language, and their speech can be ungrammatical (Grosa, Giuliani, & Narayanan, 2006). Second, children’s speech is hard for automatic speech recognition (ASR) (Potamianos & Narayanan, 2007; Russell & D’Arcy, 2007). Acoustic parameters of children’s speech, such as formants, are harder to capture and more variable than those of adult speech (Eguchi & Hirsh, 1969). Such difficulties are aggravated for tutors in noisy classrooms. Thus children’s spoken responses are unpredictable at multiple levels.

To reduce this unpredictability, we use predictable response training as introduced by Aist and Mostow (2009a). Predictable response training strikes a balance between the twin goals of technological feasibility and educational effectiveness. The tutorial dialogue should elicit utterances that are highly constrained given a particular task, yet are the evidence of a student following an educationally valuable process that maps an input to a response. That is, predictable response training calls for the design of the automated tutor’s instruction to elicit predictable yet educational responses, and then the exploitation of knowledge of the constraints under which the predictable responses are produced in designing the language model for the speech recognizer.

The idea of designing prompts to elicit predictable responses has a long history in dialogue systems (Hansen, Novick, & Sutton, 1996). The goal there was two-fold: first, to help automatic speech recognition, and second, to make the dialogue appear more natural to human users. Its instantiation in intelligent tutoring systems adds another goal, namely to scaffold student learning.

In this article, we instantiate the predictable response training framework in the context of a computer tutor for oral reading, and on the specific task of teaching young readers to generate questions about the text they are reading. Project LISTEN’s Reading Tutor listens to children read aloud, and responds with spoken and graphical feedback, as described in previous publications (e.g., Mostow & Aist, 1999; Mostow & Aist, 2001; Mostow et al., 2003). A session in the Reading Tutor consists of logging in by clicking on name and password, and taking turns with the Reading Tutor to pick the next story or activity, until the session ends due to logging out or prolonged inactivity. During a story reading, the Reading Tutor displays a story on the computer screen, adding one sentence at a time, and providing help when necessary. More recent work (Mostow & Chen, 2009) extends the Reading Tutor to teach children to generate questions about the text they are reading. When taught by human teachers, this self-questioning strategy has been shown to improve children’s reading comprehension (NRP, 2000; Rosenshine, Meister, & Chapman, 1996).

This article addresses the challenge of automatically recognizing these spoken questions, as a case study of the overall framework for using predictable response training to reduce the unpredictability of children’s speech in order to recognize it (Aist & Mostow, 2009a; Aist & Mostow, 2009b). However, the Reading Tutor recognizes oral reading in real-time, but not yet free-form responses. Therefore instead of responding directly to the spoken content, the Reading Tutor asks the child Did you say what you’re wondering? and presents two choices – Yes I said it and No I need to try again. Fig. 1 shows an animated wonderer prompting a child to ask a question. The speech recognizer highlights O’s in the speech balloon. The number of highlighted O’s is very roughly proportional to the number of spoken words or syllables. As this example illustrates, we designed the Reading Tutor to prompt for children’s spoken questions but record them without recognizing them in real time, responding independently of what they said. Instead, we used the recorded data for off-line evaluation of the speech recognition approaches reported in this article.
The rest of this article is organized as follows. The next section introduces the reading comprehension strategy of self-questioning and the challenges of recognizing children’s spoken questions in a reading tutor. We then describe how we automatically generate predictable response training instruction for self-questioning. We further describe how we generate and refine a language model that exploits such training. We then present ASR accuracy on a corpus of children’s spoken responses to self-questioning prompts. Finally we list design implications for future automated tutors that use speech input, and summarizes contributions, limitations, and future work.

CASE STUDY: RECOGNIZING CHILDREN’S QUESTIONS

In the following sections, we present a case study of the predictable response training framework: prompting and recognizing children’s questions in a reading comprehension activity – self-questioning. Our self-questioning instruction (Mostow & Chen, 2009) aims to teach a young child to wonder about text while reading it aloud to Project LISTEN’s Reading Tutor. The National Reading Panel (NRP, 2000) found self-questioning the most effective strategy to teach for reading comprehension, based on effect size for comprehension gains. In a self-questioning activity, the Reading Tutor prompts the child now and then to ask a question out loud about the text, and records the free-form spoken responses.

Self-questioning as a reading comprehension activity can be difficult for young children. In an initial design for eliciting self-questioning in which after introducing the strategy the Reading Tutor simply prompted the student to make up a question, we found that more than 70% of the time the child chose to give up and jump past the prompt to the next activity step. Partially due to this difficulty, it is also challenging to predict the behavior of children, including their questions, when they take part in this activity. Unpublished data from a previous study (Zhang et al., 2008) found considerable variation in children’s free-form responses to self-questioning prompts such as What else are you wondering about rainbows? Ask a question out loud. Out of 23 recorded responses, only one response was a grammatical question relevant to the text (Does a rainbow come out when it snows?). The rest contained only classroom background noise, did not take a question form (e.g. Nothing, Thank god I could make a promise about rainbow), were ungrammatical (e.g., How they get the colors where they come from yada yada I’m done), or were irrelevant to the text (Why do you ask so many questions).
GENERATING PREDICTABLE RESPONSE TRAINING FOR SELF-QUESTIONING

We face a dual problem in implementing self-questioning activity in a reading tutor. On the one hand, we need to reduce the difficulty of the task to children so that they are more willing to learn the strategy. On the other hand, we need to reduce the unpredictability of children’s responses so that the reading tutor can extract useful information from those responses. To cope with this dual problem, we built predictable response training into the instruction and constructed language models that exploit the instruction to improve speech recognition of the questions.

Our Reading Tutor trains three types of questions, namely Why, How, and What. These question types are specific to the text, unlike previously studied generic multiple-choice wh- questions with text-independent choices, e.g., When does this take place? in the present; in the future; in the past; I can’t tell (Mostow et al., 2004). Besides, Why and How questions generally require inference, rather than being answerable directly from facts explicit in the text. Our instruction guides students to compose questions in multiple steps, so as to elicit predictable segments. We use automated question generation in generating instruction for a given text. As we will demonstrate in the next section, automatically generated self-questioning instruction not only offers flexibility in teaching with any text, it also provides an opportunity to build language models automatically for questions given the text. This article discusses self-questioning instruction generation only for narrative fiction; extending such generation to informational text is discussed elsewhere (Chen, Aist, & Mostow, 2009).

We factor a question about a fictional text into a question stem (e.g., Why was), a character to ask about (e.g., the country mouse), and a question completer (e.g., surprised). This factorization takes into account the observation that a fictional story typically contains characters and events that the characters are involved in. Following this factorization, we generate a question by automatically extracting characters and events from a story to fill in question templates. A good question should make sense in the story context. We therefore need to represent and reason about the meaning of the story sufficiently to generate questions that make sense.

To generate questions about characters’ mental states (Chen, 2009), we represent and reason about them using the SCONE knowledge representation (Fahlman, 2006). SCONE represents mental states such as beliefs, desires, and intentions (Bratman, 1999) as nestable contexts and applies inference rules within and across them. Contexts encapsulate statements about beliefs, desires, and intentions. For example, to represent She was surprised to find the cottage-door standing open, SCONE represents the surprise as an event with two belief contexts attached to it – the girl’s old belief that the cottage-door was closed, and her new discovery that the cottage door was open. It generates this representation using an inference rule that can be paraphrased as If something surprised someone, she knew about it afterwards but not beforehand. From this representation, we can generate questions such as Why was the girl surprised? and Did the girl know that the cottage door was open?

Given the automatically generated questions, we generate self-questioning instruction by filling in example questions and question segments in a hand-authored generic instruction template. This instruction template follows a 5-step instructional model that gradually releases task responsibility from tutor to student (Duke & Pearson, 2002). The last step of the 5-step instructional model – independent use of reading comprehension strategy – does not involve the tutor. Therefore we only implement the first four steps. In the current implementation, the Reading Tutor does not record a child’s spontaneous questions, unless the child was presented with a sentence to read but instead spontaneously asked a question. Given a text, we insert spoken tutorial interjections that describe a comprehension strategy, model its use, scaffold its application, and prompt the child to use it.
Compared to text display, spoken prompts are more natural for a child to respond to by speaking his or her question. To describe the resulting dialogue, we show a prompt in *italics* if the tutor speaks it, in **boldface** if the tutor displays it as text, in ***bold italics*** if the tutor displays and reads it aloud, and with an asterisk * if the prompt asks the student to speak a question.

(1) **Describe the strategy**: the tutor introduces the strategy of self-questioning and explains an important component of a question, namely the question stem:

   Tutor: *I’m going to tell you about a reading strategy called QUESTIONING.**

   Asking yourself questions while you read can help you understand better. A good way to start a question is with a question word. These are some good question words: *why*, *who*, *where*, *when*, *what*, and *how*.

(2) **Model the strategy**: the tutor models the strategy with an example question.

   Tutor: *This part of the story makes me think of this question: “Why was the country mouse surprised?”*

   [Student reads more text]

(3) **Scaffold the strategy**: To help the child make a question, the tutor provides multiple choices for all or some question segments.

   (3a) **Full completion**

   Tutor: *Let’s make a question about ___ (the town mouse; the country mouse; the man of the house; the cat).*

   Student:  

   [In the on-screen menu of 4 choices, the student clicks on the country mouse.]  

   Tutor: *Let’s ask a ___ (what; why; how) question.*

   Student:  

   [The student chooses why.]  

   Tutor: *Let’s complete your question: Why did the country mouse ___ (decide to send the cat; try to taste everything before his tummy was full; run)?*

   Student:  

   [The student chooses decide to send the cat.]  

   * Tutor: *Ok, now I want you to read your question out loud before you continue the story.*

   Student reads aloud: *Why did the country mouse decide to send the cat?*

   [Student reads more text]

   (3b) **Semi-free completion**

   The tutor again prompts the child to choose a character and question type, and asks the child to complete the question by saying the whole question out loud.

   Student:  

   [The student chooses the cat and how.]  

   * Tutor: *Now finish your question by saying the whole thing out loud, and completing the rest.*

   Student: *How did the cat see the mice?*

   [Student reads more text]
Prompt the use of the strategy: the tutor prompts the child to ask a question without assistance.

*Tutor: Think of a question to ask about the story, and say it out loud.
*Student: Why did the two mice come out?

The instructional dialogue is interspersed within story readings. Depending on the story, some steps, such as step (4), may appear more than once. The duration of a completed story reading with the self-questioning instruction averages approximately 12 minutes, including the inserted tutor prompts (which typically total around 1 minute of instruction time) and the time for the student to respond.

As this example scenario illustrates, adapting an instructional method developed for human classroom instruction to use in automated individual tutoring requires altering the instruction to accommodate the capabilities and limitations of the tutoring technology. For example, a classroom teacher would likely invite group participation in applying the strategy, understand a wide range of variation in the phrasing of individual students’ proposed questions, and respond by paraphrasing and refining them. Such behavior relies on spoken and non-verbal communication skills feasible for humans but not yet for computers.

The instruction illustrated here differs in a number of respects. First, it trades group discussion for one-on-one attention. Second, it incorporates a more systematic approach to question construction, designed not only to scaffold the student’s task of generating questions, but to simplify the tutor’s task of recognizing them. Third, the tutorial responses reflect current limitations in automated speech recognition and natural language understanding. Thus by design, the resulting questions differ from the questions—if any—that children would ask in a classroom. What matters educationally is not these differences themselves, but how the differences in the quantity, quality, and process of students’ question generation affect their learning—an important issue that lies beyond the scope of this article.

**LANGUAGE MODELS FOR RECOGNIZING CHILDREN’S QUESTIONS**

Automatic speech recognition uses an acoustic model of how sounds represent words, and a language model of how words are combined into utterances. Generally, the better the acoustic model captures how users pronounce words, and the better the language model captures how users construct utterances out of words, the more accurate the recognition. Thus, researchers seeking to improve speech recognition performance typically focus on improving the acoustic model, the language model, or both. Researchers also seek to improve audio quality and reduce the range of likely ways to say things within the user’s task. This article focuses on language modeling approaches that exploit knowledge of a constrained range of likely utterances.

When little training data from real user is available, researchers in dialog systems have built language models from hand written grammars (Jurafsky & Martin, 2008), out-of-domain corpora such as the Web (Zhu & Rosenfeld, 2001), and corpora that are synthesized using in-domain phrases and out-of-domain sentences (Chung, Seneff, & Wang, 2005). In contrast, we synthesize our language model by generating questions from story text, and we expand the coverage of our language model by exploiting out-of-domain corpora such as the Dolch list (Dolch, 1936), stories written for children, and Google ngrams.

Table 1 summarizes the two data sets used in this paper. The pilot data is a 168-word corpus from a spring 2009 pilot test of self-questioning instruction generated for Aesop’s fable The Country Mouse and the Town Mouse. This corpus consisted of 12 responses by 7 second graders to the self-questioning prompts starred with * in the previous section. We used the pilot data to tune the noise...
insertion penalty of our speech recognizer and the interpolation weights of our language model components so as to minimize the word error rate (WER), which penalizes ASR substitutions, deletions, and insertions. Our test data consisted of 2,538 word tokens and 419 word types from 250 free-form self-questioning responses spoken by 34 children while reading 10 fictional narratives. None of the 34 children was in the pilot study, but 18 of the 250 self-questioning responses, by 11 children, were for the same story used in the pilot, namely *The Country Mouse and the Town Mouse*. The other 232 self-questioning responses, by 30 children (7 of whom also read the previous story) were for 8 different stories. We used the test data to evaluate our language models using various metrics, including the vocabulary coverage of the language model, WER, and the number of content words recognized.

<table>
<thead>
<tr>
<th>Data set</th>
<th># utterances</th>
<th># children</th>
<th># words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot data</td>
<td>12</td>
<td>7</td>
<td>168</td>
</tr>
<tr>
<td>Test data</td>
<td>250</td>
<td>34</td>
<td>2,538</td>
</tr>
</tbody>
</table>

**Core Language Model**

The goal for our language model generator is to better predict children’s self-questioning responses during the predictable response training instruction. In this case study, children’s spoken responses that the Reading Tutor listens to include step (3b): complete a partial question and step (4): say a whole question.

To exploit predictable response training, we generate questions automatically from the text and build them into the language model. The question generation process is fully automatic, as detailed elsewhere (Mostow & Chen, 2009). In brief, it uses the ASSERT semantic role labeler (Pradhan, Ward, & Martin, 2008) to label verbs and their arguments such as subjects and objects (more precisely, agents and patients). These semantic roles are then mapped to SCONE representations of characters and their mental states (as in the *She was surprised* example in the previous section). To generate a question, the question generator combines a question stem with a character to ask about, and a question completer. It uses agents of mental states and actions as characters, and mental state or action verbs and their patients as completers.

Our language model generator then compiles the resulting questions into a finite state grammar (FSG). Fig. 2 shows an example language model that incorporates the questions from step (3a) in the previous section. It also shows the equivalent regular expression in Java Speech Grammar Format (JSGF) (Sun.Microsystems, 1998).
Our general objective in language modelling is to increase the number of words recognized and reducing the WER. The coverage of the core language model is quite limited. A realistic model of children’s oral speech must account for the common phenomenon of disfluency (Potamianos & Narayanan, 2007), which includes repetition (e.g., *How did how did the cat see the mice*?), early termination (e.g., *Why did the cat*), hesitations, and filled pauses (e.g., *uh, um*).

Repetition alone affected 62 utterances of the total 250 utterances in our test data. To model repetition, we added transition arcs from segment boundaries to previous segment boundaries. To model early termination, we added transition arcs from segment junctions to the end state. Fig. 3 shows part of the resulting “core language model” and the JSGF regular expression for the *how-why* question grammar.

Modeling disfluency increased the number of correctly recognized words on our test data by 12.7\% absolute (48.1\% relative). However, it also increased WER by 7.6\% absolute (9.7\% relative), due mainly to increased insertions.
To model insertions such as hesitations, filled pauses, false starts, and out-of-vocabulary (OOV) words, we exploited the recognizer’s ability to allow silences and noises between words. We used an “all-phone” noise dictionary consisting of all the phonemes. We used a 0.1 noise insertion penalty to reduce ASR hypothesis probability by a factor of 10 for each inserted noise. By absorbing such insertions, noise models reduced WER by 13.7% absolute (15.9% relative), at the cost of reducing words correct by 6.8% absolute (17.4% relative) due to absorbing some non-noises as well.

Extensions to the Core Language Model

The automatically generated questions serve only as a starting point for the language model. A model built solely from the questions themselves has very limited predictive power, especially for the question completers and free-form questions. For example, the child may say the man instead of the man of the house. This section presents methods to improve the coverage of the language model while still restricting the search space.

We used the pilot data as a guidance to help us improve the core language model. In principle, we could train a language model directly from questions spoken by trained students, but in practice we would need a substantially larger corpus. For the related task of recognizing children’s spontaneous summarization, Hagen et al. (2004) trained language models on the text of 10 stories and different numbers of students’ summaries. They reported needing at least 40 summaries to achieve better recognition than the initial language model trained on 10 stories.

The language model predicts both the content of the questions and their form. Predictable response training mainly elicits the form of children’s questions, with limited possibilities for the question stem and character, but the question completer segment is more open-ended both in the words it can use and the order they can occur.

There is a trade-off between the coverage and precision of the language model. As ASR vocabulary grows, coverage of children’s speech increases, but so does the risk of misrecognition. Hence we want only words likely to appear in children’s responses, namely story words and common question words such as what, why, and how.

Children’s questions can reach beyond the vocabulary output by our question generator. The core language model vocabulary consisted only of words in the generated questions, and covered only 38% of the 60 word types in our 168-word pilot corpus. To improve coverage, we added the Dolch list (Dolch, 1936) of 220 words common in children’s books, which improved vocabulary coverage to 84% on the pilot corpus. We expected children’s questions to be about story text, so we added all the story words. We used a morphology generator to add all inflections of each verb. We used the resulting vocabulary for the rest of the models that we now describe.

To boost robustness, we tried interpolating the core language model with broader models: a unigram model, a bigram model, a part of speech (POS) bigram model, a trigram model, and a POS trigram model.

Core LM + unigrams: A unigram model is less likely to overfit than higher order n-grams. We trained the unigram model on 158,079 words in 673 children’s stories from Project LISTEN. These stories range in difficulty from kindergarten to grade 7 and are drawn from various sources including Weekly Reader, Project Gutenberg (www.gutenberg.net) and other public-domain Web sources, and stories authored in-house. We incorporated this unigram model by inserting a self-looping state in the core FSG to allow any sequence of words after the character segment, using the unigram probability for each word. We give the transition into this state a low weight (.0001) tuned on the pilot data as a penalty so as to give such sequences lower probabilities than generated questions.
Core LM + bigrams: Unigram models do not contain any information about word order. A bigram model provides information about neighbouring words by specifying the probably of a word given its previous word. We trained a bigram model on the 673 children’s stories and interpolated it with the core LM.

Core LM + POS bigrams: A bigram model is finer-grained than a unigram model. However, a bigram model has a higher risk of overfitting if the size of the training corpus is limited. In order to improve the balance between model complexity and robustness, we trained a bigram model for parts of speech. Our POS bigram language model approximates bigram probability \( P(w_2 | w_1) \) as \( P(\text{POS}(w_2) | \text{POS}(w_1)) \), e.g. \( P(\text{mice} | \text{the}) \) as \( P(\text{NNS} | \text{DT}) \), where NNS means a plural noun, and DT means a determiner. We tagged all 673 stories using the Stanford POS tagger [Toutanova et al., 2003], and trained a bigram model on the resulting POS sequences using the SRILM language modeling toolkit (Stolcke, 2002). To incorporate this model in the FSG, we added a state for each POS tag. We assigned the transition from the character segment to the VB (verb) state a heuristic probability \( 0.0001 \), and transitions between POS states their POS-bigram probabilities. We tagged each word with its most frequent POS. Thus this model approximates \( P(\text{find} \hspace{1pt} \text{the} \hspace{1pt} \text{mice}) \) as \( 0.0001 \times P(\text{DT} | \text{VB}) \times P(\text{NNS} | \text{DT}) \).

Core LM + trigrams: A trigram model includes more context than unigrams and bigrams. However, to avoid overfitting, we need a bigger training corpus than the 673 children’s stories. Therefore we collected trigrams from the following sources:
- We approximated our FSG in trigram form by enumerating predicted questions and a subset of their disfluent forms (restricting repetition to 2 times) and collecting their trigram counts.
- We used 62,263 human transcripts of oral reading utterances and free-form responses collected by the Reading Tutor. The oral reading data contained a large proportion of disfluent oral reading and off-task speech. The free-form response data came from tutorial activities such as summarizing and thinking aloud. These data provided valuable examples of children’s spoken interaction with the tutor, where characteristics of spoken language such as disfluency are likely to be observed.
- From the 976,992,639 Google trigrams (Brants & Franz, 2006), we extracted the 727,348 trigrams consisting solely of words in predicted questions, the story, and the Dolch list (Dolch, 1936). We multiplied the question trigram counts by 1000 to weight them more heavily, added the transcript and Google trigram counts, and used the combined counts to train our trigram language model.

Core LM + POS trigrams: Trigrams have an even higher risk of overfitting than bigrams. Instead of word trigrams, we trained POS trigrams on the same corpus used to train the regular word trigram model. We grouped trigrams by their POS, aggregated the counts, and assigned the aggregated counts to all trigrams with the same parts of speech. As a result, all the word trigrams with identical trigram POS share the same probability.

Improving Precision by Reducing Insertions
Most ASR errors were insertions caused by background speech and noise rare in carefully controlled lab settings, but unavoidable in real-world classroom environments. To improve precision, we tried two approaches: (1) tightening search by lexicalizing question segments; (2) post-processing ASR output to filter out low-confidence words.

Lexicalizing the language model. User testing showed that children often paused between question segments and within question completers, but not within question stem and character
segments, as in *Why did ... the man of the house ... try to hurt things, um, the mice?* These pauses suggest a high cognitive load (Berthold & Jameson, 1999) when starting a new segment or thinking up a question completer.

The recognizer can insert silences between but not within words, so we lexicalized question stems and character segments in order to exclude unlikely pauses. Thus the stem *Why did* mapped to a single lexical item *why-did*, and the character segment *the man of the house* to *the-man-of-the-house*.

**Confidence thresholding.** The speech recognizer estimates for each hypothesized word how likely it was recognized correctly (Huang, Acero, & Hon, 2001, Ch. 9.7). Sphinx3 uses an ASR confidence score, ranging from a large negative value to a large positive value, to capture this likelihood in the logarithmic domain. To distinguish correctly recognized words from misrecognized words, we used Sphinx3’s default confidence score threshold (>-2000). We filtered out hypothesized words with confidence scores smaller than the threshold.

**Comparison Models**

In addition to the configurations described above, we implemented two models to compare them against.

**Baseline:** As a baseline, we trained a trigram language model on a corpus of 403,867 word tokens and 11,856 word types consisting of the same 673 stories and 62,263 human-transcribed oral reading and free-form responses used for the “Core LM + trigrams,” but without adding Google trigrams and generated questions.

**Cheating:** Given the acoustic model, how well can a language model possibly do in terms of ASR accuracy? To obtain a rough upper bound on ASR accuracy, we did a “cheating experiment” using a FSG language model consisting of just the transcribed word sequences from the test set.

**EVALUATION**

We conducted ASR experiments to evaluate language models that exploit predictable response training against a baseline that does not. To compare language models, we used otherwise identical configurations of the Sphinx3 speech recognizer, including continuous HMM acoustic models trained on 495 children’s oral reading. The training data consists of 60,542 transcribed utterances totaling 43 hours of audio recording.

As in previous work (Zhang, et al., 2008), our 250-utterance test data contained ungrammatical questions and off-task speech (i.e., utterances not considered self-questioning). Each utterance was recorded by the Reading Tutor in a classroom setting, with partial assistance from an on-site research assistant. Some of the recordings contain the staff’s speech for helping children with the activity, such as *Say what you’re wondering*.

ASR performance depends on several parameters: search beam width to restrict search to relatively good candidates, language model weight to specify its importance relative to acoustic scores, and insertion penalties for words, silences, and noises. To make the evaluation competitive, we tuned these parameters for the baseline configuration, achieving 28.0% words correct and 79.3% WER. We then used the same parameter values for the other conditions, rather than sacrifice any of our already limited test data as development data to retune them for each condition. Thus the resulting comparison is conservative, biased in favor of the baseline.

Table 2 shows the evaluation results on the test data. The baseline configuration uses a trigram language model described previously. The cheating experiment used a FSG language model consisting of just the 250 transcribed word sequences from our test set. The Core LM includes
automatically generated questions and disfluency. The next five models interpolate it with n-gram models to improve recall by covering question completers better. The last two models were attempts to improve precision. This table shows the highest non-cheating value(s) in each column in boldface.

Table 2.
Results on unseen test data (250 utterances); bigger is better except OOV and WER.
b / c / 2 means significantly \((p<0.05)\) better than baseline / core LM / Core LM + POS bigrams.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Out-of-Vocabulary Rate</th>
<th>Word Error Rate</th>
<th>Words Correct</th>
<th>Concept Recall</th>
<th>Concept Precision</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>19.7%</td>
<td>79.3%</td>
<td>28.0%</td>
<td>28.4%</td>
<td>38.5%</td>
<td>32.7%</td>
</tr>
<tr>
<td>Cheating</td>
<td>0.0%</td>
<td>47.2%</td>
<td>61.9%</td>
<td>68.1%</td>
<td>73.0%</td>
<td>70.5%</td>
</tr>
<tr>
<td>Core LM</td>
<td>32.7%</td>
<td>72.6%(^b)</td>
<td>32.3%(^b)</td>
<td>39.1%(^b)</td>
<td>59.1%(^b)</td>
<td>47.1%(^b)</td>
</tr>
<tr>
<td>Core LM + unigrams</td>
<td>19.7%</td>
<td>74.6%</td>
<td>31.4%</td>
<td>42.1%(^b)</td>
<td>53.2%(^b)</td>
<td>47.0%(^b)</td>
</tr>
<tr>
<td>Core LM + bigrams</td>
<td>19.7%</td>
<td>70.8%(^b)</td>
<td>34.4%(^b)</td>
<td>45.3%(^bc)</td>
<td>56.1%(^b)</td>
<td>50.1%(^b)</td>
</tr>
<tr>
<td>Core LM + POS bigrams</td>
<td>19.7%</td>
<td>68.4%(^bc)</td>
<td>37.7%(^b)</td>
<td>47.2%(^bc)</td>
<td>60.9%(^bc)</td>
<td>53.2%(^bc)</td>
</tr>
<tr>
<td>Core LM + trigrams</td>
<td>19.7%</td>
<td>78.9%</td>
<td>24.1%</td>
<td>30.6%</td>
<td>58.7%(^b)</td>
<td>40.2%(^b)</td>
</tr>
<tr>
<td>Core LM + POS trigrams</td>
<td>19.7%</td>
<td>78.2%</td>
<td>24.3%</td>
<td>30.9%</td>
<td>58.2%(^b)</td>
<td>40.4%(^b)</td>
</tr>
<tr>
<td>Lexicalized model</td>
<td>19.7%</td>
<td>79.5%</td>
<td>23.6%</td>
<td>30.7%</td>
<td>58.4%(^b)</td>
<td>40.2%(^b)</td>
</tr>
<tr>
<td>Core LM + POS bigrams + Confidence thresholding</td>
<td>19.7%</td>
<td>72.1%(^b)</td>
<td>31.1%</td>
<td>37.1%(^b)</td>
<td>62.8%(^bc2)</td>
<td>46.6%(^b)</td>
</tr>
</tbody>
</table>

The standard measure of ASR accuracy is word error rate (WER), which penalizes substitutions, deletions, and insertions. To present the impact of words not covered by the ASR vocabulary, we also list OOV rate along with WER. Each OOV word causes at least one ASR error. Therefore, it can be viewed as a lower bound on WER. To control for effects of OOV, we used the same vocabulary for the baseline as for all refinements of the core configuration, as described previously.

Since insertions are numerous and often harmless to our application, we also report words correct, computed as the number of correctly recognized words divided by the total number of words in the human transcript; this ratio penalizes substitutions and deletions but not insertions. For WER, larger values are worse; in contrast, for words correct and the other metrics we now discuss, larger values are better.

From an application point of view, words correct is not the ultimate objective function. The more important goal is to extract spoken meaning, not to transcribe the exact words spoken, especially function words such as the and of. We used a list of 513 function words extracted from the American National Corpus’ list of lemmas. Our test data contained 103 distinct function words and 316 distinct
content words. We ignored function words, and measured recall (#content words correctly recognized / #content words), precision (#content words correctly recognized / #recognized words), and F1 score (harmonic mean of recall and precision) of concepts, operationalized as word classes defined by word stems – i.e., two words denote the same concept if they share the same stem. If a child says the same thing twice and the speech recognizer hears it only once (or vice versa), concept recall and precision are unaffected. For assessing a child’s question, recall measures how fully the ASR credits all the words the child said, and precision measures how discriminately the ASR credits only those words.

To test statistical significance of differences between configurations, we used paired T-tests with n = number of utterances. Using n = number of transcribed words would be more sensitive but ignore statistical dependencies among recognition errors in the same utterance. Conversely, using n = number of speakers would account for dependencies among utterances by the same speaker, but be overly conservative. Following common practice in ASR evaluation (e.g., Gillick & Cox, 1989), we ignore within-speaker dependencies, treating utterances as independent.

Even with a data set of only 250 responses, the difference between recall for Core LM and the baseline was again sufficiently dramatic to be statistically significant (n = 250 utterances, p < 1e-7). Expressed as an effect size, the mean difference in accuracy between the two methods was 0.70 of the standard deviation in baseline accuracy across the 250 utterances. Both concept recall and precision on unseen data were encouraging. Due to the balance between model complexity and robustness, Core LM + POS-bigrams yielded F1 score significantly higher than both the baseline and the Core LM. The WER of the Core LM was higher than the baseline, partly due to the fact that the ASR parameters were tuned to favor the baseline. As a comparison, our ITS2010 paper (Chen, et al., 2010) experimented on a subset of the current test data, with only 18 utterances by 11 children. Even with so little data, the difference between all-concept recall for Core LM+POS-bigram and the baseline was again statistically significant (n = 18, p < 0.0001, Cohen’s d = 1.364). The baseline and POS-bigram models had WER 93.4% and 64.2%, respectively. F1 scores of the baseline and POS-bigram models were 16.9% and 52.3%, respectively.

Among the interpolated n-grams, bigrams achieved the best performance, both in WER and concept F1 score. Unigrams turned out to be too coarse while trigrams overfit. For both bigram and trigram models, the POS variant yielded lower WER and higher F1 scores. Lexicalization did not improve the Core LM, probably because children seldom speak the whole phrases it predicts. Confidence thresholding achieved the best concept precision by filtering out hypothesized words likely to be misrecognized. However, confidence thresholding also removed many words correctly recognized, and hence lowered the concept recall.

The difficulty of recognizing children’s self-questioning responses depends partially on the instruction that prompted the responses. Our instruction prompted two types of self-questioning responses. A semi-free completion consists of a question spoken after first selecting a question type and a character, illustrated by step (3b) in the example instruction scenario. A free response consists of an entire question spoken without scaffolding, illustrated by step (4) in the example instruction scenario. Semi-free completions contain predicted question stems and characters, so they are easier to recognize. Of the 250 utterances in the test data, 52 were semi-free completions, with 56.9% recall by the Core LM + POS bigram LM; 198 were free, with 42.9% recall by the same model. The corresponding accuracy for the baseline language model is lower but shows a similar contrast: 33.7% vs. 27.4%. Although recall on free responses was low, it was still higher than with the baseline language model. We also found that 3 of the 52 semi-free completions were purely off-task, while 61 of the 198 free responses were off-task. Since the language model covers only on-task speech, the
recognizer either rejects or mis-recognizes off-task speech. However, the higher rate of off-task speech does not fully explain the difference in accuracy. In fact recall was only slightly higher for the on-task responses than on- and off-task responses combined: 57.7% (on-task) vs. 56.9% (on-task + off-task) for completions and 43.2% (on-task) vs. 42.9% (on-task + off-task) for free responses.

CONCLUSIONS, LIMITATIONS, AND FUTURE WORK

This article describes a general framework for recognizing children’s free-form speech. The framework includes two suggestions: (1) to balance the tradeoff between educational effectiveness and technical feasibility, engineer tutor instruction to elicit predictable responses; (2) to improve speech recognition of children’s free-form spoken responses, especially in the absence of large numbers of transcribed children’s responses, exploit predictable response training to build a language model, and then improve the coverage of the language model by interpolating it with a back-off model trained from external corpora. We instantiate this framework in a reading tutor, using a language model based on the text being read and the strategy being taught. The framework can be applied to other learning environments. For a language tutor, a back-off model might include a language model trained from texts containing the target vocabulary. For a science tutor, a back-off model might include a language model trained from science articles containing related science concepts and terminologies that the learner has acquired.

We use the framework in a case study of recognizing free-form questions asked by children while reading. In the case study, we describe how to generate self-questioning instruction by combining predictable response training with automatic question generation. Due to the limitations of acoustic models of children’s speech in a noisy environment, we need a strong language model that can predict children’s likely responses. We describe how to generate language models that exploit predictable response training. Finally we evaluate our approach on unseen test data. We demonstrate ASR accuracy higher than for a baseline language model.

The case study in this article instantiates the framework of predictable response training only for a single form of instruction, namely teaching self-questioning. Testing the generality of the framework will require applying it to additional types of instruction.

The language modeling technique presented in this article provides a starting point when little or no existing free-form speech is available. With more such data, training acoustic models may improve recognition of children’s free-form speech in noisy classroom environments.

This article reports only off-line recognition experiments. Future challenges include using ASR in Project LISTEN’s Reading Tutor to analyze and respond to children’s free-form speech in real-time – that is, to advance from listening to children read aloud to listening to them think aloud. A key open question is how accurate ASR must be in order to enhance tutoring more when it is correct than it degrades tutoring when it is wrong. The answer depends on what tutorial decisions depend on ASR, and their robustness to ASR error. The Reading Tutor can benefit merely from distinguishing among null, terse, and fuller responses. Further benefits of ASR need not wait for perfect accuracy, but can accrue by identifying tutorial decisions that even imperfect ASR can improve.

ACKNOWLEDGMENTS

The research reported here was supported by the Institute of Education Sciences, U.S. Department of Education, through Grant R305B070458. The opinions expressed are those of the authors and do not necessarily represent the views of the Institute and the U.S. Department of Education. We also thank
the educators, students, and LISTENers who helped generate and analyze our data, and Nell Duke and Christina Trotochaud for their expertise, comments, and advice.

REFERENCES


