Exploiting Predictable Response Training to Improve Automatic Recognition of Children's Spoken Responses

Wei Chen[1], Jack Mostow[1], and Gregory Aist[1,2]

[1]Project LISTEN, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213, USA
[2]Applied Linguistics and Communication Studies, Iowa State University, Ames, IA 50011, USA
{weichen,mostow}@cs.cmu.edu, gregory.aist@alumni.cmu.edu

Abstract. The unpredictability of spoken responses by young children (6-7 years old) makes them problematic for automatic speech recognizers. Aist and Mostow proposed predictable response training to improve automatic recognition of children's free-form spoken responses. We apply this approach in the context of Project LISTEN's Reading Tutor to the task of teaching children an important reading comprehension strategy, namely to make up their own questions about text while reading it. We show how to use knowledge about strategy instruction and the story text to generate a language model that predicts questions spoken by children during comprehension instruction. We evaluated this model on a previously unseen test set of 18 utterances totaling 137 words spoken by 11 second grade children in response to prompts the Reading Tutor inserted as they read. Compared to using a baseline trigram language model that does not incorporate this knowledge, speech recognition using the generated language model achieved concept recall 5 times higher – so much that the difference was statistically significant despite small sample size.

Keywords: children's free-form spoken responses, predictable response training, automatic speech recognition, language model, self-questioning strategy for reading comprehension, Project LISTEN's Reading Tutor

1 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 [1-3]. Compared to typing [4], 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

[1] 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.

children have trouble typing accurately or quickly. Compared to multiple choice interfaces, a speech interface is less distracting, and it allows a broader range of input.

However, recognizing children's free-form speech is a tricky problem [5, 6]. Acoustic parameters of children's speech, such as formants, are harder to capture and more variable than those of adult speech [7]. Besides, children are creative in syntactic-lexical use of language, and their speech can be ungrammatical [8], which increases the unpredictability of the speech.

To reduce this unpredictability, we apply predictable response training [9]. We then exploit knowledge of predictable responses in the language model of a speech recognizer. We develop this approach in a Reading Tutor that teaches young children to generate questions about story texts (also known as "self-questioning"). Teaching this strategy has been shown to improve children's reading comprehension [10, 11].

The rest of this paper is organized as follows. Section 2 introduces predictable response training for self-questioning. Section 3 and 4 respectively describe how to generate and improve a language model that exploits such training. Section 5 reports results. Section 6 summarizes contributions, limitations, and future work.

2 Predictable Response Training in Self-questioning Instruction

Our self-questioning instruction [12] attempts to teach a young child to wonder about text while reading it aloud to Project LISTEN's Reading Tutor [13]. 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.

Unpublished data from a previous study [14] found considerable variation in children's 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*).

To reduce the unpredictability of children's responses in self-questioning, we built predictable response training into the instruction. We train three types of questions, namely *Why*, *How*, and *What*. Our instruction guides students to compose questions in multiple steps, so as to elicit predictable segments. We decompose 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*). We follow an instructional model that gradually transfers responsibility from tutor to student [15]:

(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[3]: *I'm going to tell you about a reading strategy called QUESTIONING.*
QUESTIONING means you ask YOURSELF questions WHILE you read.

[3] Tutor prompts: *italics* = spoken; **boldface** = displayed; ***bold italics*** = both; * = elicits speech.

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.

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]

After the child chooses a character to ask about and a question type, the tutor asks him or her 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]

(4) 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 inserted tutor prompts typically total around 1 minute of instruction.

3 Core Language Model

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 better 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 paper focuses on language modeling approaches that exploit knowledge of a constrained range of likely utterances.

To exploit predictable response training, we build into the language model questions generated automatically from the text. Our question generator [12] combines a question stem with two other segments it extracts from the text – a character to ask about, and a question completer. Our language model generator then compiles the resulting questions into a finite state grammar (FSG). Fig. 1 shows an example language model that incorporates the questions from step (3) in Section 2.

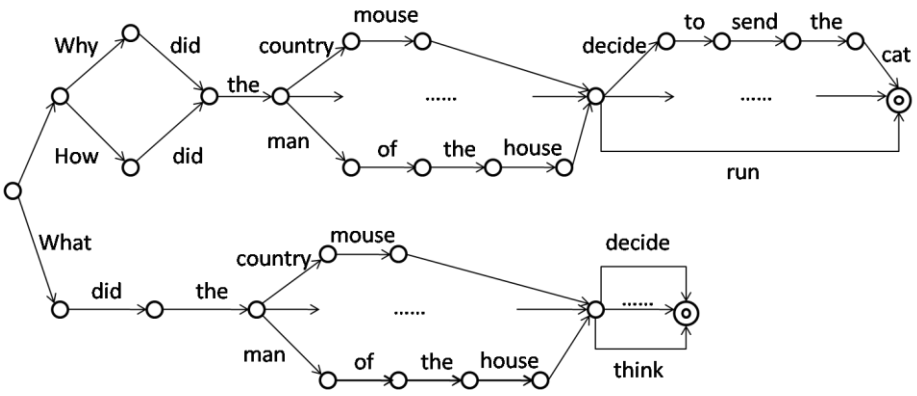

Fig. 1. Example language model.

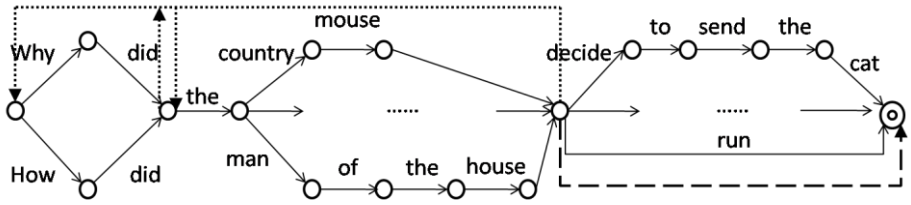

Fig. 2. A fragment of core language model with disfluency modeling.
Dotted arrows represent repetition; dashed arrows represent early termination.

Modeling disfluency. Disfluency, a common phenomenon in children's speech [6], includes hesitations, filled pauses (e.g., *uh*, *um*), repetition (e.g., *How did how did the cat see the mice?*), and early termination (e.g., *Why did the cat*). To model hesitations

and filled pauses, we exploit the recognizer's ability to insert silences and noises between words, using a noise dictionary including every phoneme. To model repetition, we add transition arcs from segment boundaries to previous segment boundaries. To model early termination, we add transition arcs from segment junctions to the end state. Fig. 2 shows part of the resulting "core language model."

4 Enhancing Robustness of the Language Model

For guidance to help us improve the core language model, we used 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 consists of 12 responses by 7 second graders to self-questioning prompts starred with * in Section 2.

In principle, we could train a language model directly from questions spoken by trained students, but practically speaking we'd need a substantially larger corpus. For the related task of recognizing children's spontaneous summarization, Hagen et al. [1] trained language models from 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 from 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.

Expanding the vocabulary with story words and common words. There is a tradeoff 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. Children's questions can reach beyond vocabulary output by our question generator: the core language model vocabulary covers only 38% of the 60 word types in our 168-word pilot corpus. To improve coverage, we add the Dolch list [16] of 220 words common in children's books. We expect children's questions to be about story text, so we add all the story words. We further expand the resulting vocabulary by using a morphology generator to add all inflections of each verb.

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

We trained the unigram model on 158,079 words in 673 children's stories from Project LISTEN. We incorporated it 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) as a penalty so as to give such sequences lower probabilities than generated questions.

Our POS-bigram language model approximates bigram probability $P(w_2 \mid w_1)$ as $P(POS(w_2) \mid POS(w_1))$, e.g. P(*mice* | *the*) as P(NNS | DT), where NNS means a plural noun, and DT means a determiner. We tagged all 673 stories using the Stanford POS tagger, and trained a bigram model on the resulting POS sequences using the SRILM

language modeling toolkit [17]. 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 the probability .0001, and transitions between POS states their POS bigram probabilities. We tagged each word with its most frequent POS. Thus this model approximates P(*find the mice*) as .0001 * P(DT | VB) * P(NNS | DT).

To construct a trigram language model, we first extracted from the 976,992,639 Google 3-grams [18] the 727,348 consisting solely of the 477 words in predicted questions, the story, and the Dolch list. Next, 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 multiplied them by 1000 to weight them more heavily, added them to the Google n-gram counts, and used the combined counts to train our interpolated trigram language model.

5 Evaluation and Results

We conducted ASR experiments to evaluate predictable response training by comparing language models that exploit such training against a baseline that does not.

5.1 Evaluation Metrics

To evaluate how many words our model correctly recognizes, we report word accuracy (WA), measured as the number of correctly recognized words divided by the total number of words in the human transcript. WA penalizes substitutions and deletions by the ASR; word error rate (WER) additionally penalizes insertions.

Concept coverage. From an application point of view, WA 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 therefore ignore function words, and measure precision and recall of *concepts*, which we operationalize 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, concept precision and recall are unaffected.

Upper bound of a language model. 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 12 transcribed word sequences from our pilot set.

5.2 Evaluation Results on Pilot Data

Table 1 shows results for the various language models described in Sections 3 and 4. As a baseline, we trained a trigram language model on the same 673 stories, but restricted its vocabulary to the words from "The Country Mouse and the Town Mouse." Exploiting predictable response training increased WA from 8.9% (WER 118%) for the baseline model to as high as 73.2% (WER 57.1%) for the core language

model interpolated with a POS bigram model. To evaluate how well the speech recognizer performs with different vocabularies, we report recall of concepts from the core language model, from the story, and from all transcribed responses. The core LM + POS-bigram model achieved the highest all-concept recall – significantly higher (despite the small sample size) than the baseline model that did not exploit this training ($n = 12$ responses, $p < 0.001$ on a paired T-test, Cohen's $d = 1.362$). To our surprise, it actually beat the cheating model on 2 of the 3 recall measures, presumably due to greater flexibility in recognizing speech atypical of the acoustic models.

Table 1. ASR results on pilot data (168 words). The baseline model is a trigram LM trained on children's stories. The Core LM covers automatically generated questions and disfluency. The next three models interpolate it with n-gram models to cover question completers better. This and subsequent tables show the highest non-cheating value(s) in each column in **boldface.**

Language Model	Word Accuracy	Recall			Precision
		Core LM concepts	Story concepts	All concepts	
Baseline (3-gram)	8.9%	16.7%	17.6%	11.7%	**81.8%**
Core Language Model	67.9%	**92.6%**	80.4%	64.9%	65.8%
Core LM + unigram	68.5%	**92.6%**	80.4%	64.9%	69.4%
Core LM + POS-bigram	**73.2%**	88.9%	**88.2%**	**68.9%**	57.0%
Core LM + Google 3gram	42.3%	59.3%	56.9%	42.9%	57.9%
Cheating Experiment	89.3%	87.0%	86.3%	84.4%	87.8%

5.3 Improving Precision by Reducing Insertions

Most ASR errors were insertions caused by background speech and noise. To improve precision, we tried two approaches: (1) post-processing ASR output to filter out low-confidence words; (2) tightening search by lexicalizing question segments. Table 2 shows their effects on the output of the Core LM+POS-bigram model.

Table 2. Improving precision on pilot data (168 words)

Configuration	Word Accuracy	Recall			Precision
		Core LM concepts	Story concepts	All concepts	
Core LM+POS-bigram	**73.2%**	88.9%	**88.2%**	**68.9%**	57.0%
Confidence thresholding	64.3%	79.6%	82.3%	62.3%	72.7%
HMM filter	57.2%	74.1%	70.6%	57.1%	75.9%
Lexicalized model	47.5%	**94.4%**	78.4%	66.2%	**76.1%**

Confidence thresholding. The speech recognizer we used assigns each hypothesized word a confidence score between 0 and 1 to indicate how likely it was recognized correctly. To separate correctly recognized words from misrecognized words with maximum accuracy, we chose a threshold on the confidence score that minimized the sum of false positive rate plus false negative rate.

Training an HMM sequential model for filtering. The confidence score rates each hypothesis word independent of its context. However, misrecognized words tend to appear in a row, and so do correctly recognized words. A sequential model, such as a Hidden Markov Model (HMM), can capture this characteristic.

Our HMM filter combines the confidence score with an intensity threshold to filter out background speech and noise, which typically have a lower intensity than student speech into a close-talking headset microphone. Since the speech recognizer may have trouble distinguishing background speech or noise from user speech, a threshold on intensity can help indicate which regions of the signal to ignore. Most of our recordings start with silence and speech by the Reading Tutor. Thus, to set an intensity threshold, the first 0.5 seconds of speech is assumed to be a silence or noise region. Then the threshold is set to be the average intensity of this noise region plus 20 times its standard deviation. We classify regions that exceed the intensity threshold as foreground speech. We used this classification and the confidence score for each hypothesis word as feature vectors to train an HMM with two states (each with a 2-dimensional Gaussian emission distribution and diagonal covariance matrix). We expect these two states to represent correct and incorrect recognition.

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 [19] when starting a new segment or thinking up a question completer.

To exclude unlikely pauses from the language model, we lexicalized question stems and character segments. 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*.

5.4 Results on Unseen Test Data

Table 3 shows results on 18 self-questioning responses by 11 students, collected after the analyses reported above. Even with so little data, the difference between all-concept recall for Core LM+POS-bigram and the baseline was again sufficiently dramatic (5x) to be 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.

Table 3. Results on unseen test data (137 words)

Configuration	Word Accuracy	Recall			Precision
		Core LM concepts	Story concepts	All concepts	
Baseline	6.6%	14.0%	17.5%	10.3%	46.7%
Core LM	**60.6%**	**80.0%**	**77.5%**	**58.8%**	50.6%
Core LM+POS-bigram	40.9%	68.0%	65.0%	50.0%	54.8%
Confidence filter	38.5%	64.0%	57.5%	49.7%	**84.4%**
HMM filter	31.2%	50.0%	50.0%	43.4%	75.6%
Lexicalized model	54.7%	78.0%	75.0%	57.4%	73.6%

Both overall and story-concept recall on unseen data were encouraging, but lower than on the pilot data we used to tune the language model weight, repetition weight, vocabulary, filler word penalty, silence penalty, and filter model parameters. This tuning likely overfit the small amount of pilot data we used for development.

6 Contributions, Limitations, and Future work

This paper describes a 2-part approach to improve ASR of children's free-form spoken responses. One part trains children to make more predictable responses. Ideally we could evaluate this part by comparing speech with versus without predictable response training as the only manipulation, but the training is inextricably interwoven with the strategy instruction itself, and ASR performance reported earlier on free-form responses elicited by different instruction [14] was very low.

The other part generates language models to exploit this predictability by integrating constraints on expected content and form, not just interpolating n-gram models from different sources [20]. We constrain content by limiting vocabulary to the story, questions generated from it, common words, and verb inflections. We constrain form based on the instruction and on word order in the story and other text.

We demonstrated ASR accuracy 5-fold higher than for a baseline language model, tested various methods to improve precision and recall, and compared their effects. Future work includes generalizing to other text, and to tasks besides self-questioning.

As a reviewer of this paper pointed out, predictable response training may itself have educational benefits. A direct benefit to the student comes from the schema that gives rise to the predictability: the same scaffold that makes responses predictable also makes them easier for the student to generate, and hence to learn. An indirect benefit is to facilitate assessment: predictable responses are easier to score. This paper has shown how to exploit predictable response training in ASR, paving the way to realize this benefit in intelligent tutors that listen to children not just read but talk.

References (LISTEN publications are at www.cs.cmu.edu/~listen)

1. Hagen, A., Pellom, B., Vuuren, S.v., Cole, R.: Advances in Children's Speech Recognition within an Interactive Literacy Tutor. In: HLT-NAACL, pp. 25--28. Association for Computational Linguistics, Boston (2004)
2. Litman, D.J., Silliman, S.: ITSPOKE: an intelligent tutoring spoken dialogue system. In: Demonstration Papers at HLT-NAACL, pp. 5--8. Association for Computational Linguistics, Boston, Massachusetts (2004)
3. Meron, J., Valente, A., Johnson., W.L.: Improving the Authoring of Foreign Language Interactive Lessons in the Tactical Language Training System. In: SLaTE. Farmington, PA (2007)
4. Wijekumar, K., Meyer, B.J.F.: Design and pilot of a web-based intelligent tutoring system to improve reading comprehension in middle school students. International Journal of Technology in Teaching and Learning 2(1), 36--49 (2006)

5. Russell, M., D'Arcy, S.: Challenges for computer recognition of children's speech. In: SLaTE, pp. 108--111. Pittsburgh, PA (2007)
6. Potamianos, A., Narayanan, S.: A Review of the Acoustic and Linguistic Properties of Children's Speech. In: Proceedings of IEEE Multimedia Signal Processing Workshop, pp. 22--25. IEEE, Chania, Crete, Greece (2007)
7. Eguchi, S., Hirsh, I.J.: Development of speech sounds in children. Acta Oto-Laryngologica Supplementum 257, 1--51 (1969)
8. Gerosa, M., Giuliani, D., Narayanan, S.: Acoustic analysis and automatic recognition of spontaneous children's speech. In: Interspeech, pp. 1886--1889. Pittsburgh, PA (2006)
9. Aist, G., Mostow, J.: Designing Spoken Tutorial Dialogue with Children to Elicit Predictable but Educationally Valuable Responses. In: Interspeech. Brighton, UK (2009)
10. Rosenshine, B., Meister, C., Chapman, S.: Teaching students to generate questions: A review of the intervention studies. Review of Educational Research 66(2), 181--221 (1996)
11. NRP: Report of the National Reading Panel. Teaching children to read: An evidence-based assessment of the scientific research literature on reading and its implications for reading instruction. http://www.nichd.nih.gov/publications/nrppubskey.cfm, Washington, DC. (2000)
12. Mostow, J., Chen, W.: Generating Instruction Automatically for the Reading Strategy of Self-Questioning. In: 14th International Conference on Artificial Intelligence in Education, pp. 465--472. IOS Press, Brighton, UK (2009)
13. Mostow, J., Beck, J.: When the Rubber Meets the Road: Lessons from the In-School Adventures of an Automated Reading Tutor that Listens. In: Schneider, B., McDonald, S.-K. (eds.) Scale-Up in Education, vol. 2, pp. 183--200. Rowman & Littlefield Publishers, Lanham, MD (2007)
14. Zhang, X., Mostow, J., Duke, N.K., Trotochaud, C., Valeri, J., Corbett, A.: Mining Free-form Spoken Responses to Tutor Prompts. In: Proceedings of the First International Conference on Educational Data Mining, pp. 234--241. Montreal (2008)
15. Duke, N.K., Pearson, P.D.: Effective Practices for Developing Reading Comprehension. In: Farstrup, A.E., Samuels, S.J. (eds.) What Research Has To Say about Reading Instruction, pp. 205--242. International Reading Association, Newark, DE (2002)
16. Dolch, E.W.: A Basic Sight Vocabulary. The Elementary School Journal (1936)
17. Stolcke, A.: SRILM -- An Extensible Language Modeling Toolkit. Proc. Intl. Conf. on Spoken Language Processing 2, 901--904 (2002)
18. Brants, T., Franz, A.: Web 1T 5-gram Version 1. Linguistic Data Consortium. (2006)
19. Berthold, A., Jameson, A.: Interpreting Symptoms of Cognitive Load in Speech Input. In: User Modeling: Proceedings of the Seventh International Conference, pp. 235--244. Banff, Canada (1999)
20. Jang, P.J., Hauptmann, A.G.: Improving Acoustic Models with Captioned Multimedia Speech. ICMCS 2, 767--771 (1999)