SCIENTIFIC LEARNING READING ASSISTANT™: CMU SPHINX TECHNOLOGY IN A COMMERCIAL EDUCATIONAL SOFTWARE APPLICATION

Valerie L. Beattie, Ph.D.

Scientific Learning Corporation
300 Frank H. Ogawa Plaza, Suite 600, Oakland, CA 94612
www.scilearn.com
vbeattie@scilearn.com

ABSTRACT

The Reading Assistant™ software application is a reading tutor designed to provide the support and practice needed to improve reading skills. CMU Sphinx speech recognition technology is used to provide the guided oral reading experience that is key to building reading fluency and ability. The application listens and follows along as a student reads the displayed text, providing feedback and help if the student gets stuck or makes an error.

In many ways this application of speech recognition technology is different from typical commercial applications, and we have made many modifications to the CMU Sphinx technology to adapt it to the needs of the Reading Assistant application. The goal for our application is what we term 'reading verification': rather than trying to determine *what* the user said, our goal is to determine *whether* the user read the presented text, and *how well* it was read. In addition, the majority of users are children, requiring the development of custom acoustic models.

Index Terms— CMU Sphinx-2, CMU PocketSphinx, children's speech, reading tutor

1. BACKGROUND

The application of speech recognition technology to enable reading tutor and other educational software has been an active area of research and development over the past several years. Notable efforts in this arena include CMU's Project LISTEN[1,2], IBM's Watch-me!-read[3], SRI's EduSpeak®[4], and reading tutor work in CSLR (now CLEAR) at the University of Colorado[5].

Guided oral reading is recommended as reading instruction practice and its value is supported by research[6]. However, given public school class sizes and budget constraints, it is not possible for teachers or other trained professionals to provide sufficient practice with this kind of one-on-one help and feedback to students while they read. Speech recognition technology has the potential to fill this gap, but the challenges are also considerable. On the one hand, the speech recognition software must accommodate the acoustic variability from a wide range of user ages, accents, reading rates, and noise conditions. At the same time the application must strive to identify 'undesirable' variations in the user's reading of the text, i.e. reading errors.

The development of the Reading Assistant application began in 2000 with the foundation of Soliloquy Learning, Inc. In 2008 Soliloquy Learning was acquired by Scientific Learning Corporation and the Reading Assistant product continued its development as part of Scientific Learning's suite of educational software products. Since its commercial launch, we estimate that more than 50,000 students have used the Reading Assistant application.

2. THE READING ASSISTANT APPLICATION

Reading Assistant is a comprehensive reading program designed to build vocabulary, comprehension, and fluency. In the most recent release, Reading Assistant Expanded Edition, the user progresses through a library of reading selections, grouped by reading level.

For each selection, the user first previews a selection, then reads it silently, and answers guided reading questions. At this point he or she also has the option to listen to a model narration of the selection, and to use the glossary capabilities to get context-sensitive definitions for key terms. Figure 1 shows a page of a selection while in the preview mode.

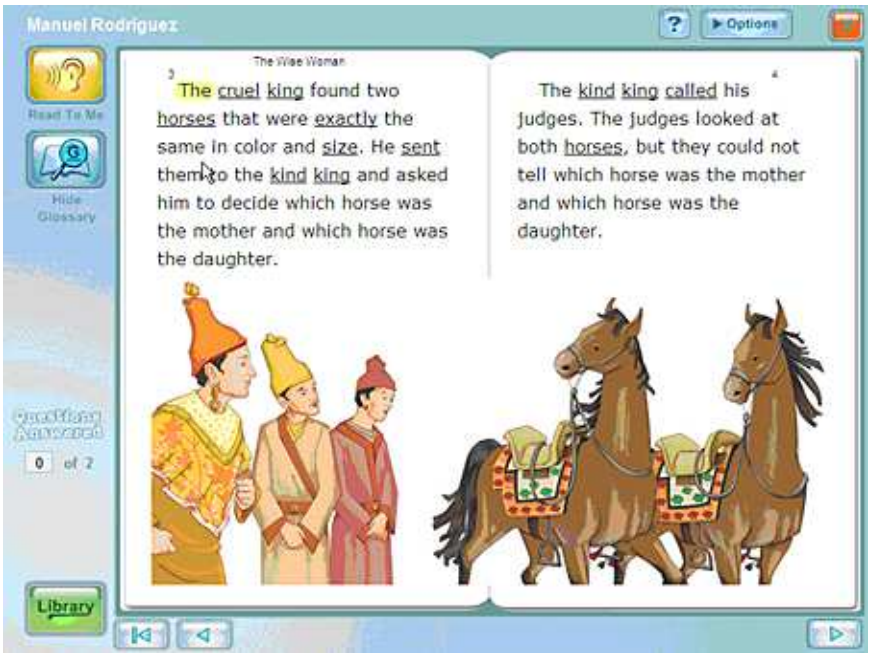

Figure 1 - Preview Mode Page Spread in Reading Assistant Expanded Edition

Next, using a headset microphone, the student reads the selection aloud. Sphinx speech recognition technology is used to interpret the reading. When the application detects that the student is unable to read a word, or has made a significant error, the software will intervene with visual and/or audio prompting. The application will first highlight the word so the student can try again, and then, if needed, it will read the word to the student. Due to the importance of this reading practice, Reading Assistant Expanded Edition requires users to read a selection two or three times. After reading a selection, a user may listen to his or her reading of the selection.

In the final step, users take a quiz to measure their comprehension of the material. The quiz questions may be multiple choice, true/false, or ask the student to select the sentence or paragraph that communicates a particular idea.

The student is given feedback on their performance in each of the task areas (preview, read aloud, and quiz). The user is also given a review list of words, including words that required prompting (intervention) during the oral reading, as well as words that the software determined had more subtle errors or hesitations associated with them.

The software also provides a wide range of reports to teachers, and administrators. These include measuring performance and tracking progress over time in different areas (fluency, comprehension strategies). Teachers can listen to recorded samples of oral reading and see words the students had difficulty with. Reports can be generated at the student, class, grade, school or even district level.

3. THE READING VERIFICATION TASK

As mentioned previously, the application of speech recognition technology to the guided oral reading process in Reading Assistant is different from a typical speech recognition application. This difference has motivated many of the modifications made to the recognition technology, and necessitated the development of application components outside of the recognition technology that manage the recognizer configuration, data processing, position tracking, and user interaction functionality. Figure 2 illustrates the architecture of the guided oral reading functionality in Reading Assistant.

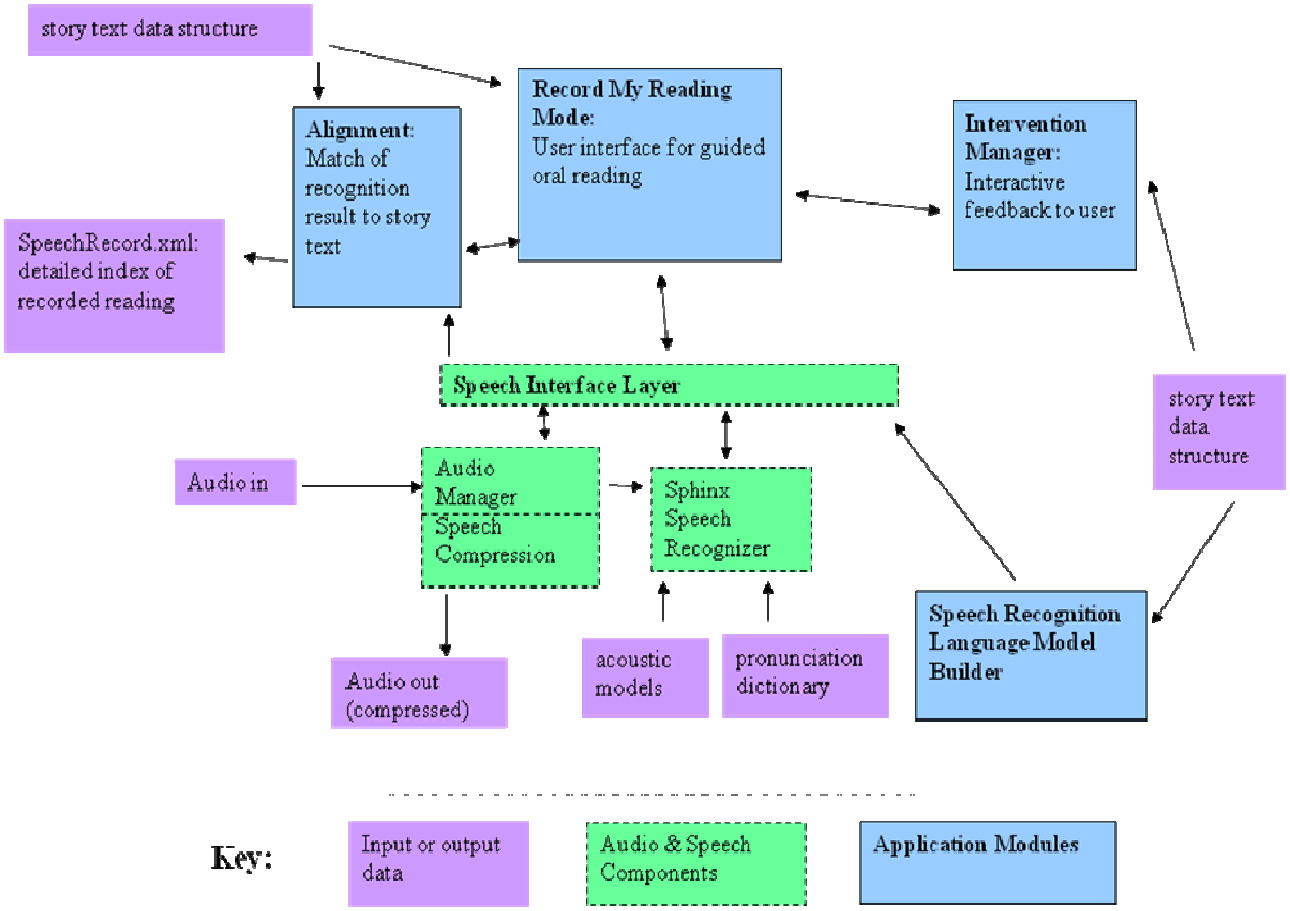

Figure 2 - Guided Oral Reading Components

Fundamentally this is a verification task where we wish to determine whether or not the student read out loud each word of the text correctly. This influences both how the recognizer is configured for this task and how we measure performance. In terms of configuration, we know the exact text the user is supposed to be reading and we track progress through this text during the oral reading. Therefore we can reasonably limit the number of words from the text that we include in the language model to words in a relatively small window of text around where the user is currently reading. At the same time, the user may misread words and interject unrelated speech or non-speech sounds. Therefore the language model configuration at any given point in the text consists of a combination of words local to that point in the text, plus 'competition' elements. Competition elements include word foils[7] (models of mispronunciations or partial pronunciations of the word), noise models, and context-independent phoneme models.

The guided oral reading task, with the goal of promoting fluency in reading, also influences the design of the application. Aside from whether the recognizer heard a user read a word, we also measure whether it was read fluently – i.e. without inappropriate hesitations or false starts[8]. Therefore timing information from the recognition process, and not just word sequence, is important and must be accurate.

In addition we have implemented a hierarchy of word importance which is used both in the configuration of the recognizer and in the processing of recognition results. In terms of comprehension, some words (such as articles and prepositions) are less likely to be critical to meaning. Readers are expected to know these short common words well. Because they are short and less important to meaning, these words are often de-emphasized or mumbled in read text, and often misrecognized. Therefore we may not prompt or stop the user even if we do not get a correct recognition on a word in this category. Whether or not a word should be placed in this category depends both on text reading level and on context, since at lower levels readers may still be learning some of these words.

In addition, some application and recognizer changes were motivated by the difficult acoustic environments in schools and the challenges of obtaining a high quality audio

signal, especially when the users are children. A significant amount of effort has gone into a microphone check capability which detects signal quality issues[9] and instructs the user regarding correct microphone headset placement.

Finally, it is worth briefly discussing performance measurement for this application and the guided oral reading activity. Since this is a verification task, the most useful metrics are:

- **false negative rate:** the percentage of the time we prompt or intervene on a word that was read correctly
- **false positive rate:** the percentage of the time we do not prompt or intervene on a word that was read incorrectly

In our experience, the fundamental usability requirement is that the false negative rate be kept very low, typically 1% or less on average across a corpus of test data. Higher false negative rates lead to frustration and detract from the goal of building and promoting fluency. False negative and false positive rates trade off against one another when tuning a system, depending on the language model weights that are used and the penalties applied to 'competition' elements such as phoneme filler models. This trade-off led to the development of a graded categorization of errors, where less severe errors such as mispronunciations and hesitations are marked by the software and placed on the word review list, but there is no intervention or real-time correction of the user. Mispronunciations are detected using a word confidence metric, and hesitations are categorized using timing analysis.

4. CMU SPHINX TECHNOLOGY IN READING ASSISTANT

Work on the development of Reading Assistant using CMU Sphinx technology began in 2002. Among available recognition technologies at the time, Sphinx was chosen because of its accessibility and its application orientation. Sphinx-2 in particular was designed to be real-time and had a basic application interface. The version initially incorporated into the application was Sphinx-2 version 0.4.

Since elementary-age students were the primary initial focus of the software, the application also required the development of acoustic models based on children's speech. To develop these we used the SphinxTrain suite of programs for acoustic modeling. In more recent releases of the software, we have created improved models for adults as well as children using data collected with the application software.

In 2006 we updated the Sphinx recognizer code base by merging the PocketSphinx code base into our code base. The goal of this merge was to obtain code fixes, support for running the recognizer on an embedded device, and the ability to evaluate fully continuous models.

5. MODIFICATIONS TO SPHINX

Developing the Reading Assistant application for the education market required many modifications to the Sphinx software. These changes were largely driven by the requirements of the education market as well as by the unique needs of the Reading Assistant application.

First of all, the source code had to be ported to the Apple Macintosh platform because Macs represent a significant share of the education market and install base. The work required was mainly in developing the audio input and output component for this platform.

Another area of modifications was to make memory use, management, and load time more efficient for recognizer configuration data. In particular, the way Reading Assistant creates and uses language models required significant re-work of the corresponding code. Even though Reading Assistant does use the Sphinx-2 'n-gram' language model implementation designed for statistical language models, true statistical language models (generated from a large corpus of text) are not appropriate for the reading verification task.

Therefore we have implemented a language model generation process which creates language models by rule from the given text. Trigram models represent the expectation that the user will mostly adhere to the presented text, with back-offs to bigram and unigram models allowing for departure from the correct word order. Word-specific competition (word foil or mispronunciation) can occur in the same n-gram positions as the correct word. Items that are not text-specific, such as noise models, context independent phoneme filler models, and silence models, can be inserted at any point.

In fact, finite-state grammar language models might be more appropriate for the reading verification task, in order to model user behaviors such as word repeats and sentence re-starts. The Sphinx version originally integrated did not include finite-state grammar support, so we adapted the n-gram implementation to suit our application as described above. However finite-state grammars are an approach we wish to investigate in future work.

Since we know the text the user is reading and are following along during reading, we can use this information to 'focus' the language model on the area of text the user is reading at the moment. This implies that we need to generate on-the-fly, and switch between, many 'small' language models, each representing a short segment of text. These needs led to a number of changes to the language modeling portion of the code including the development of a binary format for language models, the ability to load language models from a stream, and other

changes to optimize memory usage and memory management.

In the Reading Assistant application acoustic models may also need to be switched or re-loaded. The changes made to enable this included adding the capability to re-initialize the recognizer within the application, and binary formats for some acoustic model files to reduce load time. These improvements were needed in particular to support an automatic model selection capability that was introduced in Reading Assistant 4.1. Reading Assistant has models which cover a spectrum of users from 5 or 6 years old to adults, but it is not necessarily easy or obvious for a user (or even a teacher) to select the model set that is going to work best, particularly for older children and teenagers. Model selection is done automatically based on a one-time enrollment process where the user is asked to repeat a small number of phrases.

At the same time as automatic model selection, a vocal tract length normalization (VTLN) factor[12] is also calculated. This capability was added to the front end feature extraction in Sphinx and is used to further improve the match of the user's speech to the model selected.

Another modification made to the Sphinx front-end was the detection of specific noise quality problems. The following signal quality issues are detected: breath noise (breath pops), low signal to noise ratio, and hum noise in the signal due to 60Hz power signal harmonics. These signal quality issues occur frequently in school situations; school-age children have a lot of difficulty using a headset microphone correctly, and poorly designed audio hardware and environment contribute to a high incidence of hum noise interference in the audio signal. Detection of these issues can be used to give instruction regarding corrective action (e.g. better microphone placement). This capability can also be used to alert users and teachers that a significant problem exists, and prevent use of the recording functionality when signal quality is too poor.

Another significant group of modifications to CMU Sphinx is aimed at providing 'competition' elements during speech recognition processing in order to determine if the user has made an error in reading. As mentioned previously, an additional 'filler' dictionary consisting of context independent phoneme models has been added to the recognizer. This is implemented as a separate dictionary from the noise fillers so that separate penalties can be applied. A word confidence score measure[10,11] has also been implemented, which also required the addition of a context-independent phoneme network in parallel with the main recognition search. The score from this 'competitor' network is used in the word confidence calculation. Finally, the dictionary implementation has been modified to accommodate the addition of partial pronunciations and mispronunciations of words (word foils) to the dictionary at load time.

6. ACOUSTIC MODELS

The development of acoustic models using SphinxTrain for the Reading Assistant application began with the development of models based on children's speech in order to obtain adequate performance on the Reading Assistant's largest target user group. The initial acoustic models were developed from commercially available children's speech corpora. More recent work in acoustic model development has included developing improved models for adult female speakers (another important user group since the majority of primary education teachers are women). All of the models developed for the application have been semi-continuous acoustic models.

Finally, in our most recent release of the software, we enhanced our set of acoustic models further by adding adult and child acoustic models focused on the dialects of the Southern United States. These models were developed using more than 110 hours of audio data from 685 speakers, collected at schools using a customized version of the application. With the addition of the new models to the application, the false negative error rate on test speakers from the Southern region was reduced from 1.5% to 1%, while the false positive rate stayed constant.

7. SUMMARY

Reading Assistant is an interactive reading tutor which uses speech recognition technology to provide a helpful listener for guided oral reading practice. CMU Sphinx recognizer technology, including Sphinx-2, PocketSphinx, and SphinxTrain, has been used to develop and deploy this commercially successful application. The unique requirements of the recognition task for this application, and of the education market, have led to many modifications of the original CMU Sphinx technology.

8. REFERENCES

[1] J. Mostow, and J. Beck. When the Rubber Meets the Road: Lessons from the In-School Adventures of an Automated Reading Tutor that Listens. In B. Schneider and S.-K. McDonald (Eds.), *Scale-Up in Education* (Vol. 2, pp. 183-200). Rowman & Littlefield Publishers, Lanham, MD, 2007.

[2] J. Mostow, S. Roth, A. G. Hauptmann, and M. Kane, "A Prototype Reading Coach that Listens", *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, American Association for Artificial Intelligence, Seattle, WA, pp. 785-792, August 1994.

[3] S. Williams, D. Nix, P. Fairweather, Using Speech Recognition Technology to Enhance Literacy Instruction

for Emerging Readers, In B. Fishman and S. O'Connor-Divelbiss (Eds.), *Fourth International Conference of the Learning Sciences*, Erlbaum, Mahwah, NJ, pp. 115-120, 2000.

[4] H. Franco, V. Abrash, K. Precoda, H. Bratt, R. Rao, J. Butzberger, "The SRI EduSpeak™ System: Recognition and Pronunciation Scoring for Language Learning", *Proceedings of InSTIL 2000 (Integrating Speech Technology in (Language) Learning)*, Dundee, Scotland, 2000.

[5] A. Hagen, B. Pellom, and R. Cole, "Highly accurate children's speech recognition for interactive tutors using subword units", *Speech Communication*, Elsevier, Amsterdam, The Netherlands, pp. 861-873, December 2007.

[6] National Institute of Child Health and Human Development, *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* (NIH Publication No. 00-4769), U.S. Government Printing Office, Washington, DC, 2000.

[7] S. Barker, "Word competition models in voice recognition", U.S. Patent Application 20060058996, U.S. Patent and Trademark Office, Washington, DC, October 2008.

[8] M. Adams and V. Beattie, "Assessing fluency based on elapsed time", U.S. Patent 7433819, U.S. Patent and Trademark Office, Washington, DC, October 2008.

[9] S. Barker and J. Wolf, "Microphone setup and testing in voice recognition software", U.S. Patent 7243068, U.S. Patent and Trademark Office, Washington, DC, July 2007.

[10] F. Alleva, D. Beeferman, and X.D. Huang, "Confidence measures and their application to automatic speech recognition", IEEE International Workshop on Automatic Speech Recognition and Understanding (ASRU), Vol. 01, pp. 173-174, Snowbird, UT, December 1995.

[11] H. Jiang, "Confidence measures for speech recognition: A survey", *Speech Communication*, Elsevier, Amsterdam, The Netherlands, pp. 455-470, April 2005.

[12] P. Zhan and A. Waibel, "Vocal Tract Length Normalization for Large Vocabulary Continuous Speech Recognition", Technical Report CMU-CS-97-148, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, May 1997.