

# Speech Based English Learning Games on the Smartphone

**Neb Tadesse**  
Computer Science  
University of Washington  
[nebiyt@gmail.com](mailto:nebiyt@gmail.com)

Faculty Mentor: **John F. Canny**  
Graduate Mentor: **Divya Ramachandran**  
Summer Undergraduate Program in Engineering at Berkeley (SUPERB) 2006

*Abstract--* From our field work with school children in rural and urban areas in India, we have seen that the use of computers for education has been widely accepted by the students, teachers, and parents, but the lack of infrastructure, hardware and appropriate software poses barriers to their widespread use. Furthermore we have seen that attendance in schools is low. However the recent expansion of mobile phones presents them as a potential platform for English learning outside of the school setting. We focus on electronic mobile games for English learning and we believe that speech recognition capability enables more flexible interfaces and more advanced methods for learning conversational English. We ported an open source live speech recognizer, CMU Sphinx III to the Windows Mobile platform for use on an I-Mate SP5 Smartphone. In order to plug the recognizer into multiple games, we designed a simple interface that could be superimposed on different applications. This will allow us to write applications that support speech recognition using the interface. The front-end application we designed is a multiple choice quiz game called A+KID that utilizes this interface. The game uses speech input to gather the user's response to the given question and choices. Because Sphinx III is not optimized to be used in real time on the mobile platform, we ran a number of tests to optimize performance and have high accuracy of speech recognition for the A+Kid application.

## I. INTRODUCTION

### A. Motivation

Technology has changed the way we do every day tasks and made our life better in countless ways. But this is not true for developing regions, where many people are under the poverty line and technology is still in its early stages. Technology plays an important role in many fields such as education and healthcare. It has the potential to empower those living in poverty in developing regions, but the lack of English literacy is often a barrier to information access. About 95% of the web sites in the world are in English, which makes it hard for non-English speakers to do a task as simple as surfing the web.

There are already hundreds of projects trying to bring technology to underserved regions but not all have succeeded. One of the main reasons for the failure is that they try to use technologies made for industrialized countries. The fundamental problem with using off-the-shelf technologies is simply that they are designed for an entirely different user base, with different needs, resources and abilities. For example, even continuous electricity is not available in some regions: personal computers requiring a continuous power supply are obviously not fit for these regions. Another issue is that even if one is provided access to information via the web, the content is likely to be in some global

language, i.e. English, and literacy is an assumed prerequisite.

Thus, we are exploring the design of a viable technological solution to the English literacy problem. Our target population is primary school-aged children in rural and urban slum areas. We have conducted fieldwork in schools in the state of Uttar Pradesh, India and are currently looking at designing educational mobile games for English learning.

## B. Why Mobile Games

### I. Smartphones

Cell phones are growing exponentially today even in developing regions. They can be easily shared by many, consume less power, and they are less expensive than a PC. In 2005, the global mobile phone shipment was 795 million units and by 2008 there will be an estimated 800 million phone shipment [1]. We are still far from 100% coverage, but it looks promising.

Electric power in rural areas is very limited. Students in a village near Mumbai, India use PCs that run on car and truck batteries. These batteries regularly need recharging and the public electrical power system cannot always handle the demand [2]. Therefore the batteries could only be used as a backup in case power is out.

## II. Mobile, Immersive Learning

Absenteeism at schools in developing regions is a big problem. Most students have responsibilities of helping the family through farming or other labor. This poses a great challenge because learning needs to be continued outside of the classroom setting. We have seen that electronic games can provide an engaging method for learning for students in our target population, and so propose that fun learning activities on Smartphones can provide a way for students to practice their lessons in their own environments, during their leisure time

## C. Why speech recognition

The Samsung P207 was the first phone to come out with large-vocabulary speech recognizer. Intel came out with speech recognition remote control and Microsoft has integrated speech recognition inside the Window Vista operating system. These have changed the views many people have on speech recognition and created many fields of interest for the use of speech interfaces. In addition, there are open-source recognizers such as Sphinx from CMU. Recently Sphinx has come out with semi-continuous speech recognition for PDAs. We believe that large-vocabulary continuous speech recognition is also possible on the mobile platform. We see this as an enabler for a better English learning platform that can focus on not just reading and writing skills, but also on speaking (pronunciation), as well as provide a more flexible interface for mobile immersive learning.

## II. BACKGROUND INFORMATION

### Sphinx:

Sphinx is a Hidden Markov Model (HMM) speech recognition system designed at Carnegie Mellon University (CMU). In automatic speech recognition (ASR) systems, the recognizer first has to learn the models for each speech sound, or phoneme. This is done by providing it with training data - transcribed audio data. During the training phase, the system extracts features from the audio data which are represented as 13-dimensional vectors. It then generates parameters for Gaussian mixture models that represent each phoneme. These make up the acoustic model of the speech recognizer. Also during training, the recognizer incorporates information about probabilistic sequences of phonemes and words into a language model. Once these two models are generated, the recognizer can decode new audio input by matching it to the word with the highest probability as suggested by the acoustic and language models.

Recognition performance depends largely on the quality of the training data. The size of the training data affects both speed and accuracy. The size of dictionary controls how big the decision tree gets. The bigger the decision tree is, the slower the search through the decision tree gets. An example of a decision tree is follows [3].

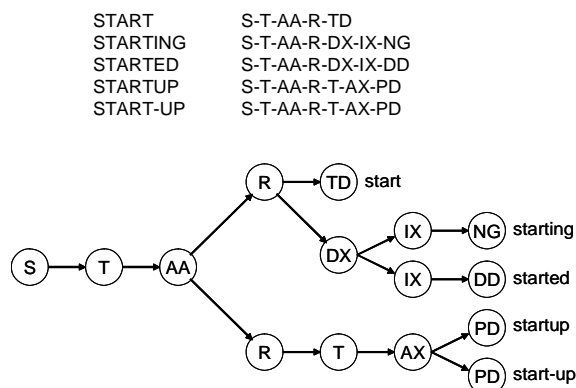


Figure 1. Decision Tree

Accuracy is a bit tricky. You want to train your speech recognizer with every word possible but at the same time the more words you train it with the closer the probabilities get. Also the variety of training data will help decrease the word error rate of the recognition. For example, there are two common ways of saying the word ONE (HH W AH N & W AH N) . If we train our speech recognition system with only the first way of pronouncing ONE, it may not detect the second the second pronunciation correctly.

The Sphinx version we are using, Sphinx III, includes three speech to text decoders. The first decoder is `s3livepretend` and it runs in batch mode. It takes its input from pre-recorded speech in raw format, extracts features from the audio, and then proposes a hypothesis based on the acoustic and language models. The next one is `s3decode`, which is similar to `s3livepretend` but takes as input audio data that is preprocessed into cepstral features. The last decoder in the Sphinx III package is the `s3livedecode`. This decoder uses live speech straight from audio card and gives you the ability to do live speech recognition. For our tests, we used all three decoders.

### III. SYSTEM DESIGN & RESULTS

#### A. Sphinx on PC

The Sphinx speech recognition system we are using is made for use on the PC. The PC we performed our tests on has a Pentium 4 processor with 3.00GHz speed and 1.00GB of RAM. In order to run Sphinx, it first needs to be trained with a speech database. The two databases we are using are RM1(1.52hr) and AN4(.07hr). These databases contain the transcriptions, control files, dictionary, phone list, and language model, for both training and decoding. The transcriptions are sentences of input data and a mapping to correspondence audio file. The control files are files containing the arguments to inputs for both training and decoding. The inputs contain words, phonemes, and a mapping to their pronunciations in audio form. Currently there are about 41 phonemes. A dictionary is a file containing a mapping between words and their phone translation. A language model is a model of the input speech using a finite state grammar. CMU has a tool that builds a language model given the speech input string.

We began by training and decoding the two databases using `s3decode` and `s3livepretend` decoder. Since the AN4 database contained audio files in form of cepstral features, it could only be used with the `s3decode` decoder. It is important to have good performance and accuracy when running the speech recognition on the phone; therefore we have run several test cases for both accuracy and speed on both databases.

We believed the size of decoding dictionary will be a key optimization. To prove our hypothesis we ran accuracy tests using different sizes of the dictionary beginning with the RM1 database dictionary. Then we ran the same test measuring the speed of the decoding process on the Smartphone. Finally we constructed a test comparing the accuracy of the two databases using the same size of decoding dictionary. The accuracy is in Word Error Rate (WER).

Decoding dictionary	Result (WER)
Large size dictionary (129 kb)	65.0%
Medium size dictionary (5 kb )	65.0%
Small size dictionary (>1 kb )	45.0%

Table 1. Accuracy test on RM1 database

Decoding dictionary	Decoding speed (sec)
Large size dictionary (129 kb)	214.05477238
Small size dictionary (>1 kb)	185.56306744

Table 2. Speed test on RM1 database

Databases	Result (WER)
RM1	43.75%
AN4	57.15%

Table 3. Accuracy between RM1 and AN4 using 4-word dictionary (ONE, TWO, THREE, FOUR)

According to our result, the smaller dictionary size will give us better accuracy and speed. These are essential when running the speech recognition on a phone. The accuracy difference between the RM1 and AN4 is mainly because in the AN4 database there was not enough training data for a couple of key words (ONE and TWO)

### B. Porting Sphinx III on the phone

I-Mate is a Windows Mobile 5.0 Smartphone with 200 MHz processor, 64 MB RAM, and 64 MB Flash ROM. Our primary goal is to run Sphinx in real time live mode on a mobile platform. Sphinx is optimized to run on Win32 platform and in order to run it on mobile phone we had to change the code to use the Windows Mobile API. Since the s3livepretend decoder is the most up to date we started optimizing it for use on the phone. After we had s3livepretend decoder working on the phone we took the front end code of the s3livedecode and superimposed

it with the s3livepretend code, which let us have live speech recognition running on the Smartphone.

### C. Learning Activity/Game Design

Once we had Sphinx running on a mobile platform in live mode we then started building speech interface that can easily be used with any application. We believe the speech interface is important because it allows application developers to use speech recognition simply by calling a few functions in the speech interface. In addition, the code does not depend on any optimization that will be made in future. This will come useful because currently the speech recognition is slow, but since we designed the interface to be independent of the recognizer optimization we don't have to modify it. The design of the speech interface is followed.

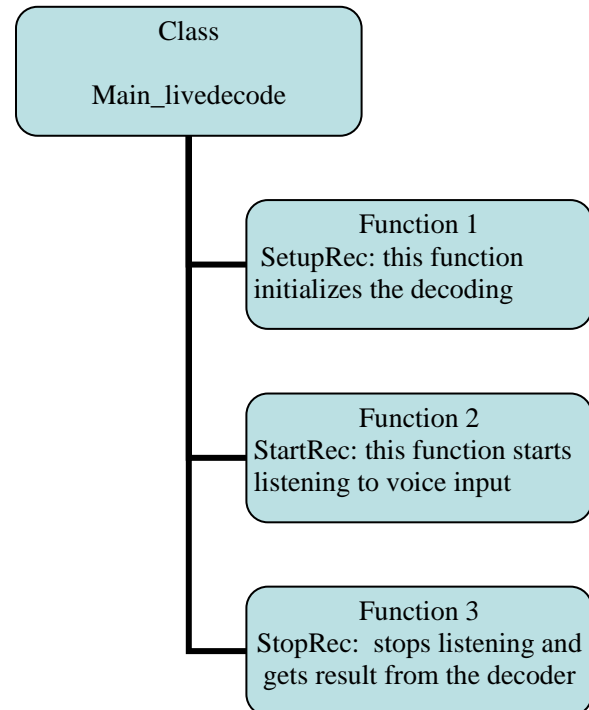


Figure 2, Speech interface design

We have developed a multiple choice game called A+KID that uses the speech recognition interface we implemented. When developing A+KID we used the technique of building

prototypes on paper. This is called low-fidelity prototyping also known as “lo-fi”. Lo-fi prototypes are extremely fast to develop and you can go through many iterations flushing problems out in the design process. In contrast high-fidelity or hi-fi is prototype that uses from demo-builders to high level languages. The problem with hi-fi prototypes is that they take long time to build and change. Typically interface designers spend 95% of their time thinking about the design and only 5% thinking about the mechanics of the tool [4]. Starting with hi-fi prototypes could cause designers to spend lots of time on mechanics because they would run into problems frequently.

The game is an object oriented design where each question is an object containing variables and functions. The design helps break up the complexity of the project into manageable chunks. Each question has behaviors and states that defines it. A picture showing what an object of question looks like in A+KID is followed.

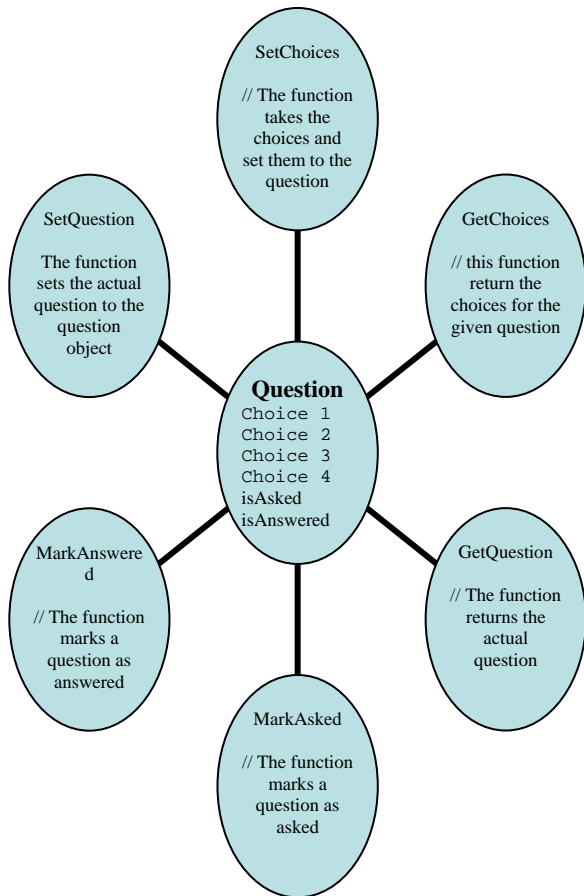


Figure 3. A+KID Design

As you run A+KID it display four images numbering them one through four as multiple choices and a question will display in the bottom of the screen. Once a user sees the display of images and question s/he could answer each question using two different ways. The first way of answering each question is using the number pad on the phone, number pad one through four corresponding to choices one through four. A user can also press the “Talk” key in the right upper corner and say his/her answer choice using the microphone on the phone. Once the user responds with an answer s/he needs to press the “Talk” button again to tell the application to stop taking input. To proceed with the next question a user could press the “Next” key in the left upper corner. Once a user is done going through and answering all the questions a display of his/her score will appear at the end of the Game.



Figure 4. A+KID game

#### IV. FUTURE WORK

1. *Accuracy*: we saw how accuracy needs some improvement. One suggestion is training the speech recognition only with data that is needed. In this case speed may also get

improved because the decision tree will be smaller. We found that there is a collection of clean digits, audio recording of numbers, provided by CMU. We believe using the data to train the system will improve both accuracy and speed.

2. *Emulator*: Running A+KID on the Smartphone was difficult for several reasons. The Emulator for the phone didn't support live audio, input from audio card. It made it hard to debug and know if the speech recognition was working or not. We had to run all the tests on the Smartphone. We suggest having a test code to check the speech recognition. This test could be as simple as taking your voice input and writing it into a file or playing back the sound it took. This would help with debugging and testing the speech recognizer on the phone.

3. *Continuous Listening*: The speech interface we designed does not support continuous speech listening. The user has to let the system know when to start and stop taking input. The system could even use a key word to initialize taking input and listen to silence for some period of time to stop taking input.

4. *Application Extension*: A+KID could use some extensions. Currently the game is designed to be a multiple choice quiz game and it doesn't teach the user. It would be really useful if the game have a training part before quizzing. The user could also have the option to skip the tutorial and jump into the quiz. In addition, it would be really useful if the game provides feedback after each question. As the current state of A+KID, the user cannot know if s/he got a question right wrong. A score at the end tells you how well you did in the overall session, therefore feedback would be useful. Finally, A+KID supports 4-word dictionary and we believe larger vocabulary would be useful if there is a high accuracy of speech recognition.

## V. CONCLUSION

The idea of using speech based games for English learning shows some promise and the speech interface we developed could easily be integrated with different applications. The speech recognition system will need some improvement but the speech interface itself will not be affected with any optimization made on

the system. The game framework also could be modified to easily change the subject and type of the game. Extensions can be made to the current stage of the project to expand and improve the performance of A+KID.

## VI. ACKNOWLEDGMENTS

I would like to express my great appreciation to my faculty mentor Dr. John Canny, graduate mentor Divya Ramachandran, fellow graduate students in Berkeley Institute of Design, the SUPERB-IT program, the University of California at Berkeley, and the National Science Foundation.

## REFERENCES

- [1]. The Development of Camera Phone Module Industry, 2005-2006, [http://www.okokok.com.cn/Abroad/Abroad\\_sho\\_w.asp?ArticleID=1034](http://www.okokok.com.cn/Abroad/Abroad_sho_w.asp?ArticleID=1034)
- [2]. Kanellos, Michael. "Rural India's Rough Road to Computer Literacy." (2005). July 2006 [http://news.com.com/Rural+Indias+rough+road+to+computer+literacy/2100-1047\\_3-5700701.html](http://news.com.com/Rural+Indias+rough+road+to+computer+literacy/2100-1047_3-5700701.html)
- [3]. Sphinx-III – Lexical Tree Structure [www.liacs.nl/~erwin/SR2003/Students/10\\_SR\\_Triphones.ppt](http://www.liacs.nl/~erwin/SR2003/Students/10_SR_Triphones.ppt)
- [4]. How Designers Word – making sense of authentic cognitive activities, 1998, <http://www.lucls.lu.se/People/Henrik.Gedenryd/HowDesignersWork>
- [5]. Brewer, Eric, Michael Demmer, Bower Du, Melissa Ho, Matthew Kam, Sergiu Nedeveschi, Joyojeet Pal, Rabin Patra, Sonesh Surana, and Kevin Fall. "The Case for Technology in Developing Regions." (2005): 25-38.