SJIF 2015 : 3.801
ISSN: 2348-3083
An International Peer Reviewed & Referred
SCHOLARLY RESEARCH JOURNAL FOR HUMANITY SCIENCE & ENGLISH LANGUAGE

A SPEECH RECOGNITION AND SYNTHESIS TOOL

Prof. Mapkar Atiqua Yunus

Asst. Prof. D.G.Tatkare Mahavidyalay, Mangaon- Raigad, Maharashtra-402104

Abstract

Many of the new technologies designed to help worldwide communication – e.g. telephones, fax machines, computers – have created new problems especially among the hearing and visually impaired. A person, who has severe hearing impairments, particularly to the extent in which deafness occurs, may experience difficulties communicating over a telephone as he or she is unable to hear the recipient's responses. Conversely, someone with visual impairments would have little inconvenience using a telephone but may not be able to communicate through a computer because of the difficulties (or, in the case of blindness, impossibility) in reading the screen. The goal of this paper is to incorporate current speech recognition (speech-to-text) and speech synthesis (text-to-speech) technology into a chat room, thus, providing a solution to communication between the hearing and visually impaired that is free and does not require any additional equipment besides a computer.

Introduction:

For most people, communication with others is quite simple. There are many options available: telephones, mail, electronic mail, chat rooms, instant messaging, etc. However, this becomes a more difficult task for those with disabilities. A deaf person does not have the luxury of being able to dial someone on the telephone and talk without additional equipment. Similarly, it is difficult for a blind person to communicate through mail or electronic means that require being able to see the screen. Direct communication between a deaf and blind person is almost impossible without a mediator.

There currently does not appear to be much software that directly addresses communication between the blind and the deaf. Much of the technology found is quite expensive, involves additional hardware, or uses Braille rather than speech. For example, Freedom Scientific offers a package solution, which allows for communication face-toface,

over telephones, and over the Internet; however, this costs $6,705. The goal of this paper is to incorporate current speech recognition (speech-to-text) and speech synthesis (text-to-speech) technology into a chat room, thus, providing a solution to communication between the deaf and blind that is both free and does not require any additional equipment besides a computer.

There were several programming languages that could have been used in the creation of this project. Ultimately, C++ was chosen because of its speed, cross-systems capabilities, and the fact that well-tested packages pertaining to speech recognition and synthesis were found to be written in C++ and C. Due to time constraints, it would have been unwise to create a speech synthesis or recognition program from scratch. A search of the Internet produced several packages, which had been written over the course of several years and involved groups of highly skilled individuals who specialized in speech recognition and synthesis. Two of the packages found, Festival, and Sphinx-3 were incorporated into the project. The Festival Speech Synthesis System, written in C++, offers a framework for building a speech synthesis system. Sphinx-3 provides the means for the speech recognition aspects.

2. Background

The main focus of this application is to provide an environment in which speech synthesis and recognition may be successfully implemented. This is done in the hopes that the accuracy ratings of this system would range in the 90-percentile. In order to accomplish, it was essential to develop strong background knowledge of the current processes and terminologies associated with speech synthesis and recognition.

2.1 Speech Synthesis:

As Text-to-Speech implies, speech synthesis involves two basic processes: the reading in of text and the production into sound. For simplification purposes, call these the front end and back end, respectively. First, the front end must read the text and transform any numbers or abbreviations into text. For example, "lol" would be changed to "laugh out loud." A phonetic transcription is then transcribed to each word using text-tophoneme (TTP) or grapheme-to-phoneme (GTP) processes. How a text should be spoken, including the pitch, frequency, and length of the phonemes is determined in this stage. This makes up the symbolic linguistic representation. The back end then takes that representation and attempts to convert it into actual sound output according to the rules created in the front end.

Speech synthesis has little, if any, understanding of the actual text being read. Such software is, typically, not concerned with what a sentence or word actually means. Rather, it simply uses dictionaries or rules to make guesses as to how the text should be read. Text-to-phoneme conversion guesses the pronunciation by using either the dictionary-based approach or the rule-based approach. In the dictionary-based approach, a large dictionary of words and spellings is stored by the program and accessed at appropriate times. This method, however, is very space consuming. The other option, rule-based approach uses preset rules of

pronunciation to sound out how a word should be pronounced. Most speech synthesizers use a combination of both approaches .

2.2 Speech Recognition

Speech recognition allows a computer to interpret any sound input (through either a microphone or audio file) to be transcribed or used to interact with the computer. A speech recognition application may be used by a large amount of users without any training or may be specifically designed to be used by one user. In this speaker dependent model, accuracy rates are typically at their highest with approximately a 98% rate (that is, getting two words in a hundred wrong) when operated under optimal conditions (i.e. a quiet room, high quality microphone, etc).

Generally, modern speech recognition systems are based on the hidden Markov models (HMMS). HMMS is a statistical model which attempts to determine the hidden components by using the known parameters. For example, if a person stated that he wore a raincoat yesterday, then one would predict that it must have been raining. Using this technique, a speech recognition system may determine the probability of a sequence of acoustic data given one word (or word sequence). Then, the most likely word sequence may be determined using Baye's rule:

$$\Pr(\text{word} \mid \text{acoustics}) = \frac{\Pr(\text{acoustics} \mid \text{word})\Pr(\text{word})}{\Pr(\text{acoustics})}.$$

According to this rule, for any given sequence of acoustic data (for example, an audio file or microphone input), Pr(acoustics) is a constant and, thus, ignorable. Pr(word) is the prior probability of the word according to a language modeling. [As an example, this should ensure that Pr(mushroom soup) > Pr(much rooms hope).] Pr(acoustics|word) is obtained using the aformentioned HMMS.

3. Related Work

In the past decade there has been an increasing amount of work dedicated to web based education. Benefits of such systems are clear: distance is no longer an issue, feedback is expedient, and assignments may be catered to specific classes or individuals. Several systems have been proposed to change web-based education from meaning a reference of hyperlinks to an interactive system involving adaptation and artificial intelligence. Generally, there are two types of adaptive educational systems that are employed: intelligent tutoring systems (ITS) and adaptive hypermedia systems. ITS technologies involve curriculum sequencing, intelligent analysis of student's solutions, and interactive problem solving support. The goal

of the system is to provide an optimal path for a student to reach a goal lesson using a series of questions, problems, and examples. These are presented in varying orders and difficulties according to what the student is accurately able to answer. If an answer is incorrect, the system should be able to figure out where the student went wrong and provide detail information about the mistakes followed by additional examples and lessons. Adaptive hypermedia systems simply adjust what links are shown according to the student's learning level.

Additional web-based education systems include virtual environments, which provide a multimedia experience without the student leaving the computer, and whiteboards. Virtual environments can include field trips through museums or historical sites or scientific experiments and dissections. Interactivity allows the student to observe at his or her own speed and learn more about specific areas at a click of the mouse. Whiteboards provide a space for the user to type, draw, or present other data that can be viewed by anyone else connected to the board. In this manner, the student is not limited to simple text communication but can easily include his or her own drawings and images. Speech synthesis and recognition may also be used in web-based education to provide another means for communication and interactivity.

4. Implementation

4.1 GUI:

As a primary concern for of the application is to increase the ease in which the visually and hearing impaired can communicate, it was essential for a simple and straightforward GUI to be created. Gtkmm is a C++ wrapper for GTK+, a popular GUI library that is often packaged with UNIX system. It provides a basic GUI toolkit with widgets, such as windows, buttons, toolbar, etc. While the primary development for the application is taking place in MSVC++, there has been an effort to remain open for the possibility of cross platform capabilities. Gtkmm meets this by ensuring cross platformon such systems as Linux (gcc), Solaris (gcc, Forte), Win32 (gcc, MSVC++ .Net 2003), MacOS X (gcc), and more.

At the current phase, a simple GUI exists, which displays widgets for the basic text input and networking capabilities. There are two basic GUI views that exist according to whether or not the user is connected to a server. The first view serves primarily as a text editor. It consists of a window with one text area in which text can be inputted either through the keyboard or (in the future) using speech recognition. This text can then be saved on to the computer. As long as the user is disconnected, text files may also be opened and will be displayed in the text area.

4.2 Networking

In addition to incorporating speech recognition and synthesis, the application acts as a chat room to allow for communication over long distances. Thus, a basic client/server network was necessary. SDL_net fit the needs precisely. SDL_net is a small, sample cross-platform networking library that uses the much larger C-written SDL (Simple DirectMedia Layer). The aim of SDL_net is to allow for easy cross-platform programming and simplify the handling of network connections and data transfer. It accomplishes this through its simple and portable interface for TCP and UDP protocols. With UDP sockets, SDL_net allows half-way connections; binding a socket to an address and thus avoiding filling any outgoing packets with the destination address. A connectionless method is also provided.

The current implementation in the application uses SDL_net to open a socket which waits for a user to connect. Once the user connects, a test message is sent. If it is successfully received, then the client's GUI changes to reflect that it has successfully connect. At this stage, the server simply closes once the message is received.

4.3 Speech Synthesis

The Festival Speech Synthesis System was chosen to accomplish the task of speech recognition. Festival provides a general framework for mutilingual speech synthesis systems, although it has the most features with English. It is useable through a multitude of APIs, including C++, Java, Emacs, and through SCHEME command interpreter. Three specific classes of users are targeted: speech synthesis researchers, speech application developers, and the end user. As such, Festival is open source although many of the alterations can occur through its functions rather than altering of its code.

Festival has been successfully run using either Cygwin or the Visual Studio's Command Prompt. Overall, the basic commands are quite simple. For example, the command *(SayText "Hello world.")* successfully produces the spoken words "Hello world." Festival also comes with the ability to directly read a text file with only a few commands. A variety of voices are available, which allows the user to find the most "natural" sounding voice to him or her. Furthemore, any produced speech may also be saved as a sound file for future use.

4.4 Speech Recognition

Although the application was coded and primarily tested on Windows XP, an effort has been made to maintain the possibility for easy cross platform capabilities. Sphinx-3 continues this trend as it is workable on GNU/Linux, UNIX variants, and Windows NT or later. Written in C++, Sphinx-3 is a product of Carnegie Mellon University and is well known in the speech recognition community for its large vocabulary and, relatively, timely results. It includes both

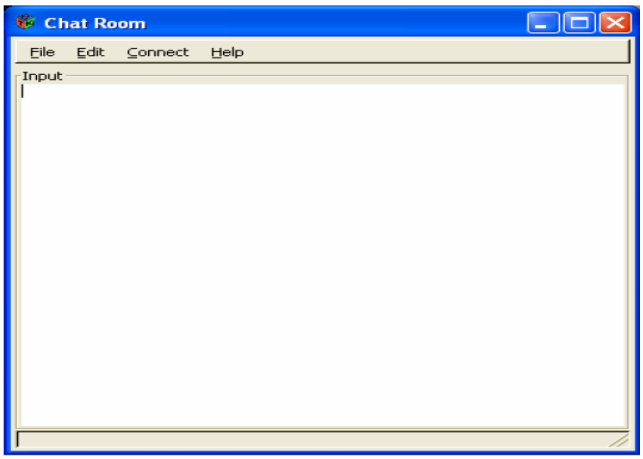
an acoustic trainer and various decoders (e.g. text recognition, phoneme recognition, N-best list generation).

In order to test that it has successfully been installed and allow a sample program of how to incorporate it, Sphinx-3 comes with a sphinx3-simple.bat, which allows the user to practice its speech recognition abilities using a limited vocabulary. A user who has never used a speech recognition program or has not used the SphinxTrain yet can expect 30-40% successful speech results. One early test of the recognition process involved stating, "Start recording. One two three four five six." This produced "START THREE CODE R E Y TWO THREE FOUR I SIXTH" as Sphinx-3's hypothesis as to what was spoken. For the most part, it is evident where these mistakes may have come from. "SIXTH" is very close to "six." In fact, the added "-th" is, most likely, a misinterpretation of the ending pause and possible background noise that was picked up.

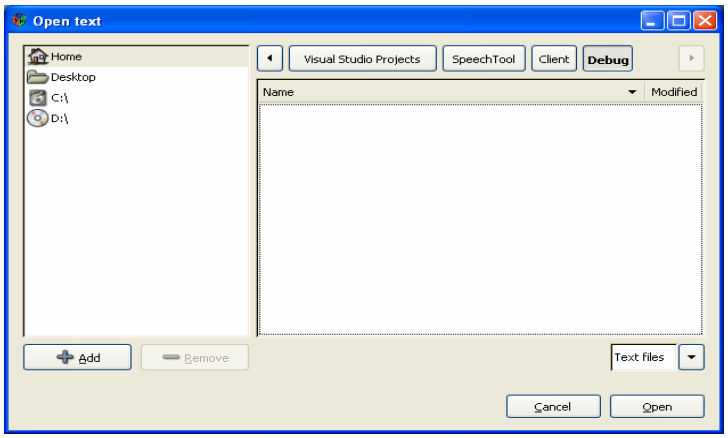
5. APPLICATION SCREENSHOTS

A. Disconnected User Images

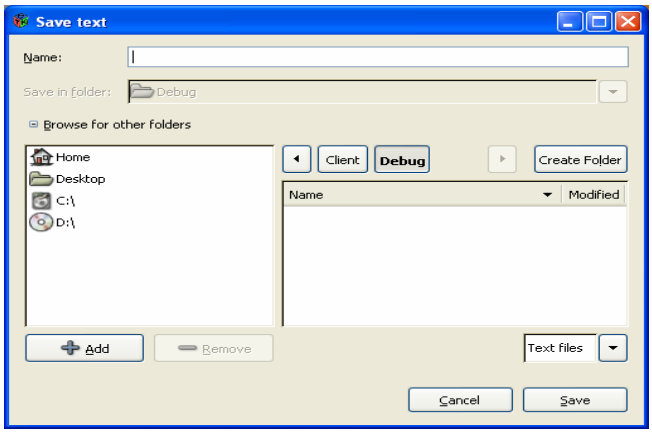
i. Disconnected User Window



ii. Opening a file

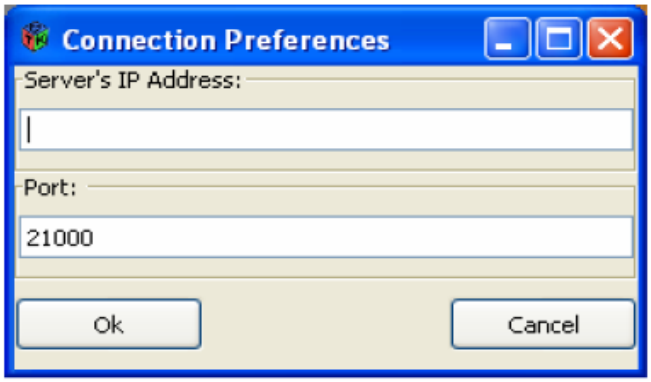


iii. Saving text to a file

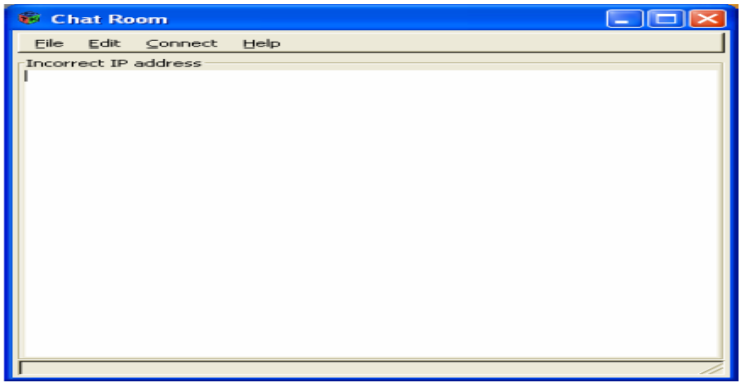


B. Connected User Images

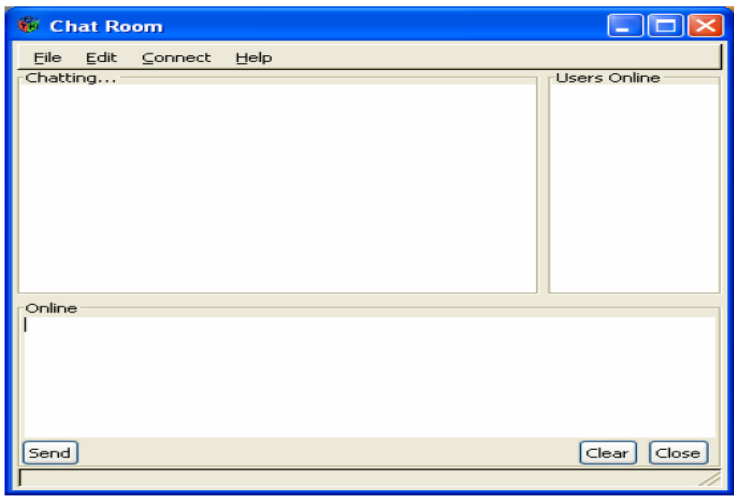
i. Connection Preferences



ii. Failed to connect because of IP



iii. Connected Client



C. Server Images

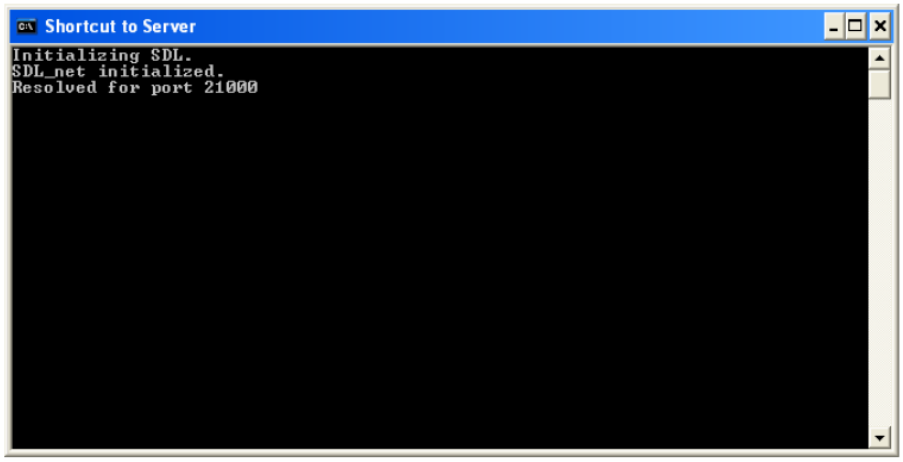


Figure B.7: Initialized server

6. Conclusion

Vigorous searching of the internet produced a variety of applications or packages which incorporated speech synthesis and recognition. However, few seemed to incorporate both. Furthermore, some had not been update for several years or had limited testing or documentation and thus were not particularly useful in studying the technologies and developing the application. The successfulness of such applications (as measured by the accuracy rating) is currently mediocre in normal conditions. Under optimal conditions (often requiring quiet isolation) significant increases of accuracy ratings were observed in speech recognition. Speech synthesis had high accuracy; however, many of the noncommercial voices still have a computerized, abnormal sound with slight imperfections in pitch, intensity, or duration.

The application created in this project uses two popular speech recognition and synthesis packages – Sphinx-3 and Festival, respectively – to ensure accurate and well tested results. Basic networking allows for a simple text chatting. Conversion using recognition or synthesis

occurs at the user end and is toggled by the user so that either feature may be turned off if they are not needed. While it has currently only been tested on Windows XP, all of the packages and implementations have been tested on Linux systems, and, thus, should be cross-platform with little, if any, changes.

Through this application, it is hoped that a simple solution is provided to communication between the hearing and visually impaired that is free and does not require any additional equipment besides a computer. Additionally, it is hoped that this application may also be used in educational settings, regardless of students' or teachers' disabilities, as a teaching aid. Thus, it may be employed for web-based education purposes.

REFERENCES

Black, Adam, et al., "Festival Speech Synthesis System," [Online], Available: http://www.cstr.ed.ac.uk/projects/festival/

"The CMU Sphinx Group Open Source Speech Recognition Engines," [Online], Available: http://cmusphinx.sourceforge.net/html/cmusphinx.php

Cumming, Murray, et al. "Gtkmm – the C++ Interface to GTK+," [Online], Available: http://www.gtkmm.org/

"Speech Synthesis," [Online], Available:http://en.wikipedia.org/wiki/Speech_synthesis

"TTS FAQ," [Online].Available:http://www.research.att.com/projects/tts/faq.html#TechWhat

"Speech Recognition," [Online], Available:http://en.wikipedia.org/wiki/Speech_recognition "Speech Research Lab," [Online], Available: http://www.asel.udel.edu/speech/ModelTalker.html

A. Le et al., "The 2002 NIST RT Evaluation Speech-to-Text Results," Proc. RT02 Workshop, 2002. Available: http://www.nist.gov/speech/tests/rt/rt2002/

"Cygwin Information and Installation," [Online],

Ravishankar, Mosur K. "Sphinx-3 s3.X Decoder (X=5)," [Online], Available: http://cmusphinx.sourceforge.net/sphinx3/

"IEL. Live Chat. Glossary," [Online], Available: http://www.illinoisearlylearning.org/chat/marks-glossary.htm

"Phones," [Online]. Available: http://en.wikipedia.org/wiki/Phones