Abstract

Although language learners often desire to improve their pronunciation of a foreign language, the software to help them do so is limited in scope. Most commercial software for pronunciation evaluation and training focuses on the acoustic signal in the evaluation and training of a learner. However, few systems, if any, give visual feedback of the learner's articulators (lips, tongue, jaw). In this paper, we describe the ongoing development of a GUI that is programmed in Objective-C for Mac OS X. Our software uses QTKit framework for video recording and playing, and some open source libraries for audio recording, audio playing, and pitch detection. The GUI incorporates and links together many kinds of phonetic data for the pronunciation learner - for example, real-time frontal video of the learner, recorded frontal and side videos of a native speaker's face during pronunciation, an ultrasound movie of the tongue moving in the mouth, and MRI images of the native speaker's tongue during the production of all the sounds in the training text.

Keywords: Interactive GUI, Pronunciation evaluation/training, Articulatory feedback, Ultrasound, MRI

1. Introduction

The pronunciation ability of second language (L2) learners is one of the most noticeable and influential factors when a native listener makes a snap judgement of the learner's proficiency. Many L2 learners want to improve their pronunciation, but individual feedback from a teacher is often impossible due to time constraints or class sizes. Thus, many L2 learners turn to software to help them with their pronunciation. Unfortunately, most commercial software for pronunciation evaluation and training focuses on the acoustic signal in the evaluation and training of a learner. The acoustics of the learner's speech signal are evaluated and displayed to the learner, but the learner is left to interpret the link between acoustics and articulation (i.e., what changes he/she should make to his/her tongue, jaw, and lips to produce required changes in the acoustic output). Few systems, if any, give visual feedback of the learner's articulators (lips, tongue, jaw), and this is counterintuitive because it is easier for a learner to interpret articulatory feedback (e.g., the movement of the tongue) than acoustic feedback (e.g., the formant values in the acoustic signal).

Because of the complete lack of this type of pronunciation evaluation software, we decided to create a GUI that incorporates visual and audio information, both native speaker model data and L2 learner feedback. At the University of Aizu, construction was recently completed on two new high-tech classrooms that contain 48 iMac computers each. Each iMac contains a built-in web camera that enables real-time recording and display of the L2 learner as he/she speaks. For this reason, we decided to develop our pronunciation evaluation GUI using Objective-C in the Mac environment.

The rest of this paper proceeds as follows. Section 2 describes some typical pronunciation evaluation software and outlines the problems with these programs. Section 3 lists the phonetic data provided by the native speaker model (the second author) and explains how this data was collected and its use in the GUI. In Section 4, we elucidate the motivation behind the GUI design and give details about its functions. In Section 5, we describe the use of Praat (open source acoustic analysis software) and a rudimentary speech recognition script (written by the first author) that finds and labels syllables for analysis within the speech signal. Finally, Section 6 presents conclusions and future work.

2. Existing Pronunciation Evaluation Software

One type of commercial pronunciation software available is AmiVoice CALL Lite [1]
(see Figure 1). The software has a number of predetermined phrases that the L2 learner must repeat. The learner's waveform and pitch are plotted and compared to the native speaker teacher's example. The speech recognition component is acceptable, but certainly far from perfect. Unfortunately for the L2 learner, the only phonetic information available is the approximate duration (from the waveform) and the pitch changes. The system gives advice to the learner, but it is always of the form of warnings not to get close to native Japanese sounds. Absolutely no video data are available and no articulatory feedback is available.

Another type of pronunciation evaluation software, one that is available online at <http://www.eyespeakenglish.com/> EyeSpeak English [2] (see Figure 2). A free demo version is available for one month before purchasing it. Like AmiVoice CALL Lite, this software has predetermined phrases to read. It also displays the teacher's and student's waveform and evaluates student's segmental pronunciation, pitch, timing (presumably duration), and loudness (measured as intensity). This software adds one visual feedback feature that is not existent in AmiVoice software - a not-very-accurate, motionless, animated image of the vocal tract showing the tongue, palate, jaw, and lips. The software grades the student, but it does not give any explicit feedback on how to improve pronunciation.

3. Native Speaker Model Data

Our pronunciation evaluation system focuses on one specific paragraph of speech (approximately 20 seconds in duration, when spoken by a native speaker). The text (Stella paragraph) comes from the Speech Accent Archive <http://accent.gmu.edu/>, a set of over 1,200 speakers from various language backgrounds reading the same paragraph. To give native pronunciation feedback to the L2 learner, our pronunciation system uses various data types, such as text, audio, video, and images, all described in the subsections below. One of the valuable points of our system is that the native speaker data we provide is real data, not simply animated versions of textbook images of the vocal tract. Our native speaker data includes ultrasound movies of the tongue moving during speech (with a CT image overlay of the palate), MRI images of the vocal tract during every English phoneme, front and side video movies of the head during speech, and high-quality audio data. We designed the GUI so that we could link these data types together and instantly and efficiently provide images, audio, and video.

3.1. Ultrasound and CT Data

Ultrasound data was collected showing the second author's tongue moving while reading the Stella paragraph. The tongue is the most important articulator and its position is crucial for pronunciation. The ultrasound data was collected in the University of Aizu's CLR Phonetics Lab using a Toshiba Famio 8 ultrasound monitor. The output of the ultrasound machine is NTSC video at 29.97 frames per second. The ultrasound video signal was captured using iMovie software on a Mac Pro computer. Since the ultrasound display only shows the tongue but no palate, we showed the palate by overlaying palate data from a Computerized Tomography (CT) still image that had been taken previously. As head movement was
minimized during ultrasound data collection, the palate could be assumed to be stationary.

3.2. MRI Data

MRI data of the second author's vocal tract was collected previously at the Brain Activity Imaging Center (BAIC), an affiliate of Advanced Telecommunications Research Institute International (ATR) in Kyoto. The MRI data was in DICOM format and was read using Adobe Photoshop CS3 software. It was converted to PNG format and resized for use in the GUI. One MRI image (side view of the vocal tract) per English phoneme had been collected and was used in the GUI. The MRI data has very high spatial resolution, giving a very clear image of the whole vocal tract.

3.3. Video Data

During the collection of the ultrasound data, two video cameras were used simultaneously, one filming the front of the head and the other filming the side, to collect video data of the face and lips. One of the cameras was a digital video camera (hard disk drive) and the other was a MiniDV video camera. No zoom feature was used, to ensure that each video was undistorted.

3.4. Audio Data

Although the video cameras contained built-in microphones, the sound quality was greatly improved by using our own DPA 4080 miniature cardioid microphone together with a Korg MR-1000 digital recorder. The audio data was aligned with the two pieces of video data using a waveform display in Final Cut Express (see section 4).

4. GUI Design

In this section, we describe the underlying programming environment, as well as the individual components of the GUI.

4.1. Cocoa and Objective-C

Apple's Objective-C based programming environment for Mac OS X is called Cocoa. We can best use the functions of Mac OS X by combining frameworks in Cocoa. Cocoa applications are typically developed using an integrated development environment called Xcode and Interface Builder, and using the Objective-C language. Interface Builder is a part of Xcode and is used for construction of the interface of an application that uses a GUI. Using Interface Builder, we can arrange the components of a GUI by dragging and dropping them with a mouse, so it is useful for designing a GUI.

Other than being able to create a native Mac application, another advantage of using Cocoa for our GUI is being able to use Objective-C. Objective-C is based on C, so it can coexist with C, and moreover, we can use a lot of open-source libraries written in C. Actually, our pronunciation system uses some of these libraries for audio recording, audio playing, and pitch detection. A GUI basically consists of objects such as buttons or windows, so Objective-C, which has the attribute of being object-oriented, is convenient for constructing a GUI.

Two main Cocoa frameworks are Foundation, the service layer organizing the functions of the OS, and Application Kit (AppKit), a collection of the control parts of a GUI (e.g., windows, buttons, menus, and text fields). Additionally, we used another framework called QTKit to realize movie playing, and to capture and record real-time movies from the internal camera.

4.2. Basic Features of GUI

One of the important requirements for our GUI is the ability to play various media data types synchronously: the native speaker's front movie, side movie, and ultrasound movie. Each movie was recorded with a different device, so they each had a different frame rate and pixel aspect ratio. To standardize them and overlay movies and images, we used Final Cut Express movie editing software.

Another important requirement is the ability to link text and each object so that the system can immediately show movies and images of a word when the word is clicked in the text. This function was realized by treating each word as an object and sending temporal and phoneme information when it was clicked.

Our GUI consists of 7 different windows (see Figure 3). The top-left window is the text of the Stella paragraph. This is to be read by the student and it also serves as a navigation tool for choosing what to display in other windows. Immediately under the text window is the waveform and pitch of the student's speech. At the bottom-left is results and advice - the evaluation of the student's
pronunciation. The four windows on the right half of the GUI show (clockwise from top-right) a teacher front view, a teacher side view with ultrasound tongue movie and CT overlay, an MRI still image with the phonemes in the selected word shown.

4.3. Text window

In Figure 3, the Text window shows the Stella paragraph. Each word is made as a button. When a word is clicked, the Teacher (front & side) windows navigate to that word and show a movie of the native speaker pronouncing that word. Also, the Teacher (phoneme) window shows the phonetic symbols of the word that was clicked. Those symbols are also created as buttons, and if one symbol is selected, the lower part of the Teacher (phoneme) window shows an MRI image of that phoneme.

4.4. Waveform and pitch window

The Waveform and Pitch window shows the L2 learner's waveform and pitch contour. When the "Record" button is pressed, the system starts recording until the "Stop" button is pressed. If the "Stop" button is not pressed within 60 seconds, the system automatically stops recording. After recording, the Waveform and Pitch window shows the learner's waveform, pitch contour, and max/min pitch value. Pitch values are filtered in this process: reject pitch values not in range (75,600) Hz, reject pitch jumps up more than 20Hz, reject if pitch values are discontinuous (less than 4), calculate the average and reject pitch not in range (average-100, average+150) Hz. The learner can select a range and play it or zoom in. If no range is selected, the system plays from the beginning to the end. To show the whole waveform after zooming, the learner can press the "All" button.

4.5. Results and advice window

Results & Advice shows the result of evaluation and advice for learner's speech. If a word was selected in Text, Results & Advice shows evaluation for the word. System uses praat to find syllable nuclei and to evaluate learner's pronunciation. For syllable nuclei detection, system calculates intensity median and reject intensity values less than the median, finds peaks more than 2dB, rejects intensity values where there are not pitch values, finds peaks again. For pronunciation evaluation, system analyzes formant for tongue movement, pitch for intonation, and duration for speech speed.
4.6. Four windows on right side

Before recording, You (front) shows real-time leaner's frontal movie from internal camera. After recording, You (front) shows recorded frontal movie during recording. Learner can watch this movie for whole paragraph by using control bar. Teacher (front & side) shows native speaker's frontal and side movie with ultrasound and palate contour. Learner can watch this movie for whole paragraph and each word by selecting from Text. Teacher (phoneme) shows native speaker's midsagittal MRI image of each phoneme. Learner can select a word from Text and a phoneme from Teacher (phoneme).

5. Praat Acoustic Analysis Software

We used Praat open-source acoustic analysis software [3] for pronunciation evaluation. Praat has a script function to automate processes, and we used some scripts to find syllables from the learner's audio and to measure pitch, formants, and duration for the syllables in the background. A view of the output of the syllable labeling script can be seen in Figure 4. The process was described in Section 4.5.

![Figure 4. Praat Script Result](image)

6. Conclusions and Future Work

In Conclusion, we have developed an interactive GUI that takes pronunciation evaluation and training to a much higher level than exists now. We use various types of real multimedia data to provide realistic and accurate views of pronunciation. In future work, we plan to test the system in a series of pronunciation classes at the University of Aizu in January 2010, further developing the system as needed. We would like to continue enhancing the speech recognition function and improving the pitch-tracking function. If we can realize very accurate speech recognition, we can link the learner's speech (front-view movie and audio) to the native speaker's, further improving the system. Ultimately, if we could use image processing to evaluate the learner's front-view movies, it would improve the system. Finally, if ultrasound were also available for the learner, image processing could be used to evaluate tongue shapes directly.

References

