SPEECH COMPARISON IN The Rosetta Stone ™

John FAIRFIELD
Department of Computer Science
James Madison University
Harrisonburg, VA, 22807
fairfijr@jmu.edu

Abstract

The Rosetta Stone™ is a successful CD-ROM based interactive program for teaching foreign languages, that uses speech comparison to help students improve their pronunciation. The input to a speech comparison system is N+1 digitised utterances. The output is a measure of the similarity of the last utterance to each of the N others. Which language is being spoken is irrelevant. This differs from classical speech recognition where the input data includes but one utterance, a set of expectations tuned to the particular language in use (typically digraphs or similar), and a grammar of expected words or phrases, and the output is recognition in the utterance of one of the phrases in the grammar (or rejection). This paper describes a speech comparison system and its application in The Rosetta Stone™.

Introduction

Funding for this research came from the developers[1], of The Rosetta Stone™ (TRS), a highly successful interactive multimedia program for teaching foreign languages. The developers wanted to use speech recognition technology to help students of foreign languages improve their pronunciation and their active vocabulary. As of this writing TRS is available in twenty languages, which was part of the motivation to develop a language independent approach to speech recognition. Classical approaches require extensive development per language.

TRS provides an immersion experience, where images, movies and sounds are used to build knowledge of a language from scratch. Since there is no concession to the native language of the learner, a German speaker and a Korean speaker both learning Vietnamese have the same experience--all in Vietnamese.

The most recent release of TRS includes EAR, the speech comparison system described in this paper. The input to a speech comparison system is N+1 digitized utterances--in the case of TRS, that includes N utterances by native speakers recorded in a studio with quality microphones, and one utterance by a student recorded in a sometimes very noisy environment with a built-in or handheld microphone. The output is a measure of the similarity of the last utterance to each of the N others. Which language is being spoken is irrelevant.

Speech comparison differs from classical speech recognition, where the input data includes one utterance, a set of expectations tuned to the particular language in use (typically digraphs or similar), and a grammar of expected words or phrases, and the output is recognition of the utterance as one of the phrases in the grammar, or rejection.

The TRS CD-ROM contains tens of thousands of utterances by native speakers. Thus the TRS data set already included the necessary input for speech comparison, but not for classical speech recognition. The first application we developed was a pronunciation guide (see Fig. 1). The user clicks on a picture, hears a native speaker's utterance, attempts to mimic that utterance, sees a display of two images visually portraying the two utterances, and observes a gauge which shows a measure of the similarity between the two utterances. The system normalizes both voices (native speaker's and student's) to a

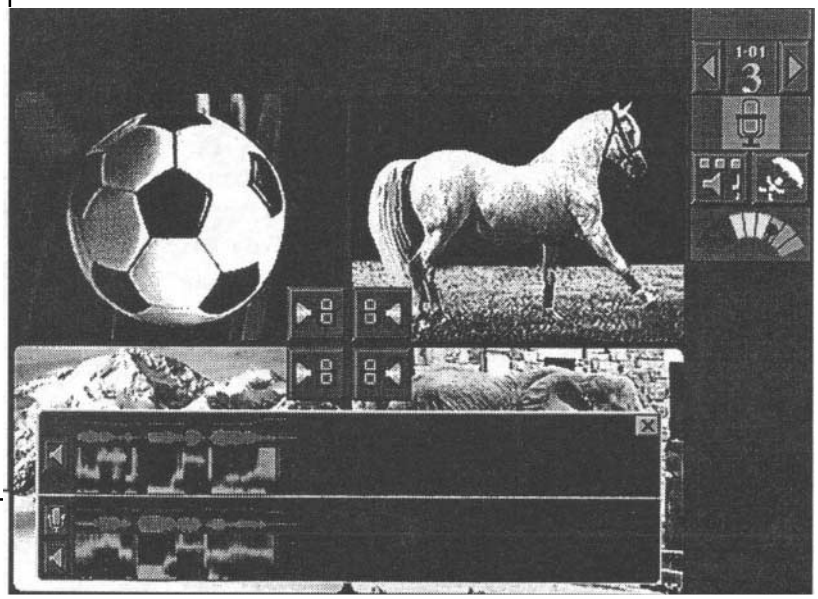
[1] FLT, 165 South Main St., Harrisonburg, VA 22801. 540-432-6166 www.trstone.com

Fig. 1. TRS pronunciation evaluation

Clicking on an image brings up the speech comparison panel, seen here imposed over the lower two images. The upper half of this panel displays a visualization of the native speaker's phrase describing the image. The student then attempts to mimic the pronunciation of the native speaker.

The visualization of the student's utterance is displayed in real time. Each visualization includes pitch (the fine line at the top), emphasis (the line varying in thickness) and an image of highly processed spectral information of the normalized voice.

The meter to the right gives an evaluation.



common standard, and displays various abstract or at least highly processed features of the normalized voices, so that differences irrelevant to speech (such as how deep your voice is, or the frequency response curve of the microphone) hopefully do not play a role.

The second application, currently under development, is active vocabulary building. The user sees four pictures and hears four phrases semantically related to the pictures. This is material they have already worked over in other learning modes designed to build passive vocabulary, i.e. the ability to recognize the meaning of speech. However in this exercise the user must be able to generate the speech with less prompting. The order of the pictures is scrambled, and they are flashed one at a time. The user must respond to each with the phrase that was given for that picture. The system evaluates their success, i.e. whether they responded with the correct phrase, one of the other phrases, or some unrelated utterance. One difficulty for the system is that frequently the four phrases are very similar, so that the difference between them might hinge on a short piece in the middle of otherwise nearly identical utterances (for example "the girl is cutting the blue paper", "the girl is cutting the red paper").

EAR is written in C. Since TRS is written in MacroMedia Director™, EAR is interfaced to TRS using Director's interface for extending Director with C code. TRS is multithreaded, so EAR is able to do its work incrementally since it must not take the CPU for extended periods of time. Indeed EAR itself contains multiple threads of two kinds: description threads and comparison threads.

Since the system might load several prerecorded utterances of native speakers at once, it is desirable that the work of computing the normalized high-level description of each utterance be done while the user is listening to those utterances, in parallel. Thus each stream of sound data (22050 Hz sound samples) is analyzed by a separate description thread, with a visual display in real time being an option. Similarly, sound data from the microphone is analyzed in real time while the student is speaking by a description thread, and the resulting visual display is displayed in real time. Description threads are discussed in Section 1.

Once the user has finished speaking, a comparison thread can be launched for each of the native speaker descriptions, which compare those descriptions to the description of the student's utterance. Comparison threads are discussed in Section 2.

1 Utterance Description

An EAR utterance description is a vector of feature vectors. Of these, only pitch, emphasis and a dozen spectral features are portrayed in the visual display. An utterance description contains one feature vector for each 1/100 of a second of the utterance.

1.1 Filters

Description of a sound stream begins with 48 tuned Danforth(1997) filters developing a mel-scale frequency domain spectrum. They are tuned 6 per octave to cover 8 octaves, the highest frequency of the highest octave being 8820 Hz, well below the Nyquist limit for a 22050 Hz sample rate. Within each octave each filter is tuned to a frequency $2^{1/6}$ times as high as the next lower filter, so that they are geometrically evenly spaced over the octave.

1.2 Speech Detection

Every 220 sound samples, i.e. about 100 times per second, the response of each of the filters is sampled. Call the resulting 48-value vector the "raw spectrum". EAR automatically detects the onset and end of speech by the following method. Let S be the sum of the upper half of the raw spectrum. If S is greater than five times the least S observed during this utterance, EAR considers that speech is occurring. This method makes EAR insensitive to constant background noise, but not to varying background noise

1.3 Voice Normalization

The natural logarithm of the raw spectrum values are smoothed in the frequency domain, using kernel widths adequate to bridge the distance between the voice harmonics of a child. This over-smoothes the signal for adults, especially males, but it makes the resulting spectral curve less dependent on the pitch of the voice and more accurately reflect formants.

The min and max of the smoothed result are mapped to 0 and 1 respectively, and multiplied by the volume, to give a measure of the distribution of energy in the spectrum. This is the data displayed in the voice panel in Fig. 1, and the data (combined with pitch and emphasis) used in the comparison discussed in the following section.

2 Comparison

This section describes the dynamic template matching approach used in EAR to match two utterances. The result of a comparison between two utterance descriptions A and B is a mapping between the two, and a scalar that on the range 0-1 gives a measure of similarity between the two utterances. A threshold on the scalar can be used to accept or reject the hypothesis that the two utterances are the same.

In a real-time thread, EAR dynamically matches a pair (A,B) of descriptions by means of a zipper object. Remember that a description contains one feature vector for each .01 second of utterance. A zipper object implements a mapping from description A to description B in patches. A patch is a segment (time-contiguous series of feature vectors) of A that is mapped to a segment of identical length (duration) in B. A zipper is a series of compatible patches--no overlaps, and the nth patch, timewise, in A is mapped to the nth patch in B. In the gaps between patches, A is mapped to B by interpolation. If the gap in A is x times as long as the gap in B, then each feature vector in the gap in B is mapped to, on the average, x consecutive feature vectors in A, such that the time discrepancy between the two patches is made up incrementally as you traverse the gap.

Initially several identical zippers are made by interpolating the two utterances onto each other wholesale--beginning to beginning, end to end, and everything in between is time interpolated. EAR then goes about randomly improving them as will be described shortly. When the zippers cease improving significantly, the best one is taken as the mapping between the two utterances.

A track(A,B) maps each feature vector of description A onto a feature vector of description B in a time non-decreasing fashion. A zipper object defines two compatible tracks, one from A to B and the other from B to A. The goodness of zipper z is defined as the least goodness of its two tracks. The goodness of a track(A,B) is the trackValue minus the trackCost.

The trackCost penalizes tracks where the timing of A relative to B is not uniform. It accumulates cost whenever timing is advanced, then retarded, etc., but permits a smooth movement in one direction without cost, so that an utterance that is uniformly slower or faster than another is not penalized.

The trackValue favors tracks which match better than would be expected--the null hypothesis. Since a track maps each feature vector of A onto one of B, the trackValue is the sum of the vectorMatches of those pairs of vectors, divided by the null hypothesis value of the match of A.

The vectorMatch(Fa,Fb) of a pair of feature vectors Fa,Fb is

vectorMatch(Fa,Fb) = Fa.infoWt*MAMI(Fa,Fb)

Let feature vectors Fa and Fb be indexed by i to access their m individual features. Then

$$MAMI(Fa,Fb) = \frac{SUM(i=1 \text{ to } m)\ \{ \min(Fa[i],Fb[i]) \}}{SUM(i=1 \text{ to } m)\ \{\max(Fa[i],Fb[i]) \}} * 1/m$$

Thus if the features are random uniformly distributed random variables in the range 0 to 1, the expected (null hypothesis) value of MAMI is ½.

Conclusion

Speech comparison in TRS enables students to focus on those elements of pronunciation that are deficient. Pitch and emphasis are used quite differently in most languages. For example, in English, pitch is used to mark questions, responses, and place in a list, whereas in Chinese there are very different words whose only distinguishing characteristic is pitch.

Some users of TRS who could not hear the difference between a vowel sound produced by a native speaker and their own vowel, have been helped by the visual display drawing their attention to the nature of the difference.

Ongoing research includes better automatic adaptation to different microphones' response curves without burdening the user with training sessions or stringent microphone requirements.

References

Danforth, Doug (1997), ftp downloadable from www.speech.cs.cmu.edu/comp.speech/Section6/Q6.3.html.