Did we say the same word? Towards identifying spoken word similarity

Ossian O'Reilly (ooreilly@stanford.edu) and Andreas Mavrommatis (andreasm@stanford.edu)

December 9, 2014

1 Introduction

Robust and accurate machine learning algorithms for speech recognition and natural language processing must be able to handle numerous challenges. For example, variabilities present in human voice recordings such as gender of speaker, tempo, pitch, pronunciation, accent, noise, etc. add complexity to the problem of transcribing voice into text.

In the particular application of identifying whether two audio samples contain the same language content, using the text transcribed from speech recognition as the similarity metric is susceptible to inaccuracies during the transcription process. Therefore, it appears that using the audio signals themselves (without the intermediate step of transcribing audio to text) could offer some benefit. Applications where it might be beneficial (or even necessary) to never transcribe to text include searching for instances of spoken words in an audio file (or a video lecture), or learning the pronunciation of a new language.

As a first step to address this problem, we perform binary classification for many pairs of words of the English language. We explore various techniques to determine what features are important in classifying spoken words.

2 Data Acquisition and Processing

Our data are audio waveforms from www.forvo.com, a user-generated pronunciation database. We downloaded waveforms for 45 different words, each containing at least 10 individual pronunciations (examples). We converted each waveform into a spectrogram of 1025 frequency bands by $\sim$250 time samples (variable, depending on original duration). We also computed Mel Frequency Cepstrum Coefficients (MFCC).

3 Feature Selection

We explored the following features: (1) Raw spectrograms; (2) Spectrograms aligned in time (to avoid bias due to misalignment of each spoken word); (3) Centroids derived from K-means clustering on both raw and aligned spectrograms (to reduce dimensionality); (4) Principal components derived from PCA on aligned spectrograms (to reduce dimensionality); (5) 2D Haar wavelet transform of raw and aligned spectrograms (efficient compression); and (6) Mel Frequency Cepstrum Coefficients (MFCC).

Each features are associated with specific thresholds which we vary in order to select the best set of features. The best features were derived as the ones that gave the lowest train and test errors after classifying two words using a Support Vector Machine (`liblinear` package). Our training set consisted of 70% of the dataset; the rest being the test set. For this stage, we focus exclusively on a single pair of words – specifically, the words 'tumblr' and 'anything' (because they had the largest number of examples).

3.1 Thresholded spectrograms

We increase the signal-to-noise ratio of each spectrogram by keeping only amplitude coefficients, A, that exceed a threshold value (above some background noise level), given by

$$A_{\text{thr}} = \mu(A) + \kappa\sigma(A) \tag{1}$$

where $\mu(A)$ and $\sigma(A)$ are the sample mean and standard deviation of the amplitudes, respectively, and κ is a scalar parameter that we vary. Setting numerous amplitude coefficients to zero makes the feature representation sparse, which improves the computational performance when using a classifier such as SVM, or when applying feature extraction using K-means clustering.

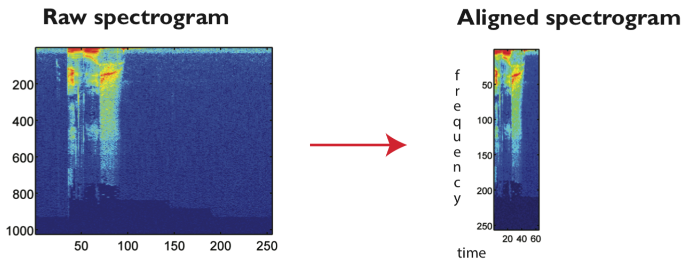

Figure 1: *Example of converting a raw spectrogram to aligned and rescaled spectrogram from a particular sample.*

3.2 Aligned and rescaled spectrograms

To avoid bias due to misalignment of each spoken word (caused by differences in the timing and tempo of speakers), we discard time periods with low signal-to-noise ratio. We assume that each spectrogram can be segmented into three segments: a silent (noise only) beginning, followed by a voice segment, and another silent segment at the end (Figure 1). To identify the voice segment, we compute and set a threshold on $||u^{(i)}||_2$, where each $u^{(i)}$ is a vector (along the frequency dimension) that contains the amplitude coefficients at time sample i. We identify the voice segment as the part of the signal that exceeds a threshold on $||u^{(i)}||_2$ for at least 4 consecutive samples (thresholds were determined empirically). In addition, to reduce computational time, we resize each spectrogram to a fixed size of 256 by 64 elements (Figure 1). As with the raw spectrograms, we keep only amplitude coefficients that exceed a specified threshold.

3.3 K-means clustering

We treat each point in the spectrograms and MFCCs as a coordinate in (t, f, A)-space; that is, (time, frequency, amplitude), and perform K-means clustering in that space. We select features by varying the number of clusters. For the raw and aligned spectrograms, we also vary the threshold on the amplitudes (Figure 2). More specifically,

- For the raw spectrograms, to avoid biases due to timing differences, we discard the time coordinate of the clusters. The feature vector in this case contains the centroid coordinates (f, A).
- For the aligned spectrograms, we retain all three dimensions in the cluster space: (t, f, A), since by aligning the spectrograms in time we have corrected for timing differences.
- For the MFCCs, we discard the time coordinate of the clusters, similar to the raw spectrograms. The feature vector in this case contains the centroid coordinates (f, A).

3.4 Principal Component Analysis

As an additional way of reducing the dimensionality of our features, we performed principal component analysis (PCA) on the aligned spectrograms. For feature selection, we varied both the amplitude threshold and the number of principal components that are kept in the PCA representation of the spectrograms.

3.5 Wavelet transform

Wavelets have been used in many signal processing and image processing applications to, for instance, compress and denoise signals. Many naturally occurring signals are suitable for compression using wavelets as only a few set of coefficients contain most of the energy in the signal. Currently, it is the preferred image compression technique in the jpeg library. Here we apply the 2D Haar wavelet transform to each spectrogram and retain the maximum modulus coefficients by thresholding.

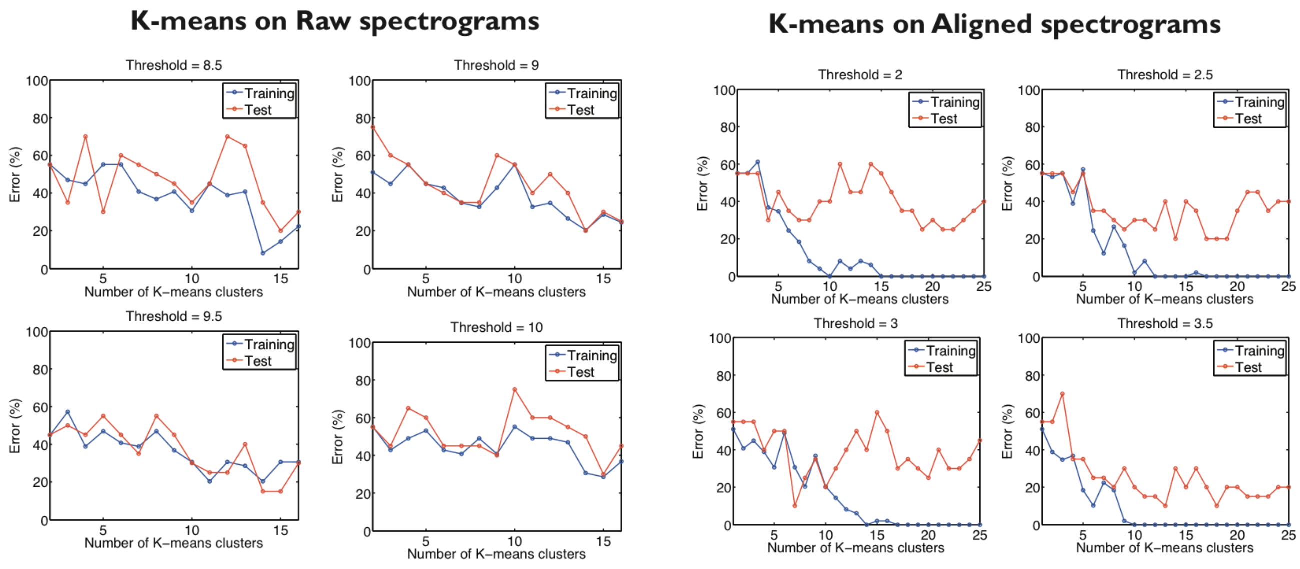

Figure 2: *Examples of feature selection process, showing the variation of training error (blue) and test error (red) as we vary the number of K-means clusters and amplitude threshold if the raw (left) and aligned (right) spectrograms. Note that aligning the spectrograms lowers the training error and in some cases the test error as well.*

4 Results and Discussion

After selecting the features that gave the best test and training errors in the word pair ‘tumblr’ vs ‘anything’ (as described in the previous section), we then performed binary classification using a Support Vector Machine (`liblinear` package) on all pair combinations of the 45 words. Our training set consisted of 70% of the dataset; the rest being the test set. We computed training and test errors for each pair of words; Table 1 lists the results of all word-pair classifications.

Aligning and resizing the time axes of the spectrograms resulted in a significantly lower test error (reduction from 36% to 24% on average; see Table 1, Figure 3), because it corrects for differences in timing between different speakers, as well as differences in tempo.

To our surprise, reducing the feature dimensionality by performing K-means clustering and PCA on the spectrograms and/or MFCC did not increase performance by any significant amount (Table 1). One disadvantage of the K-means clustering is that it produces clustered point clouds that are different for individual samples of the same word. This results in further inconsistencies in the features of examples with the same word label.

Based on the alphabetical ordering of words in the confusion matrices (Figure 3, panels 'AS' and 'WAS'), there are two distinct bands; one with low test error and one with large test error on average. The band with the lowest error corresponds to words beginning with the sound 's', which are apparently easier to distinguish from other words. We note that we did not simply have more data for words beginning with 's'. On the other hand, words that beginning with the sound 'm' are the most difficult to distinguish from other words (Figure 3). We suspect that nasal sounds such as 'm' and 'n' do not produce distunguishable auditory signals.

On average, words with more examples are more easily distinguished, regardless of the type of features (Figure 3, bottom row). Therefore, incorporating more data should increase performance. One way to expand the dataset would be to create synthetic data by manipulating our current data (i.e., changing pitch, tempo, timbre etc.). Another approach would be to extract smaller phonetic components (phonemes) from each waveform that represent individual sounds (instead of entire words). Each word would then be represented by an ordered sequence of phonemes, which would serve as a basis for words.

5 Conclusion

We applied machine learning algorithms to address the problem of classifying audio samples based on the words they contain. We find that aligned and rescaled spectrograms derived from the audio samples and wavelet transforms of the aligned spectrograms are useful features for this problem. Using a linear SVM classifier on a training set of 45 words, we find an average test error of 21.8% when we use the wavelet transform. On average, words with more examples yield lower test errors, implying that incorporating more data (i.e., more audio samples per word) should increase performance. Further development is needed for implementing practical applications, such as voice search within an audio or video record, or language learning software.

Features	Training error (%)	Test error (%)	Learning rate
Raw spectrograms	0	36.2 +/- 19.3	0.71
Aligned spectrograms	0	24.2 +/- 16.5	0.44
K-means on raw spectrograms	0	37.8 +/- 19.5	0.11
K-means on aligned spectrograms	0	36.0 +/- 21.9	0.59
MFCC	0	35.6 +/- 18.4	0.54
K-means on MFCC	0	46.3 +/- 17.2	0.40
PCA on aligned spectrograms	0	38.6 +/- 17.5	0.44
Wavelet transform on raw spectrograms	0	38.9 +/- 18.2	0.35
Wavelet transform on aligned spectrograms	0	21.8 +/- 17.0	0.46

Table 1: Error analysis for each of the feature selection methods. The learning rate is defined as the decrease in test error per additional example.

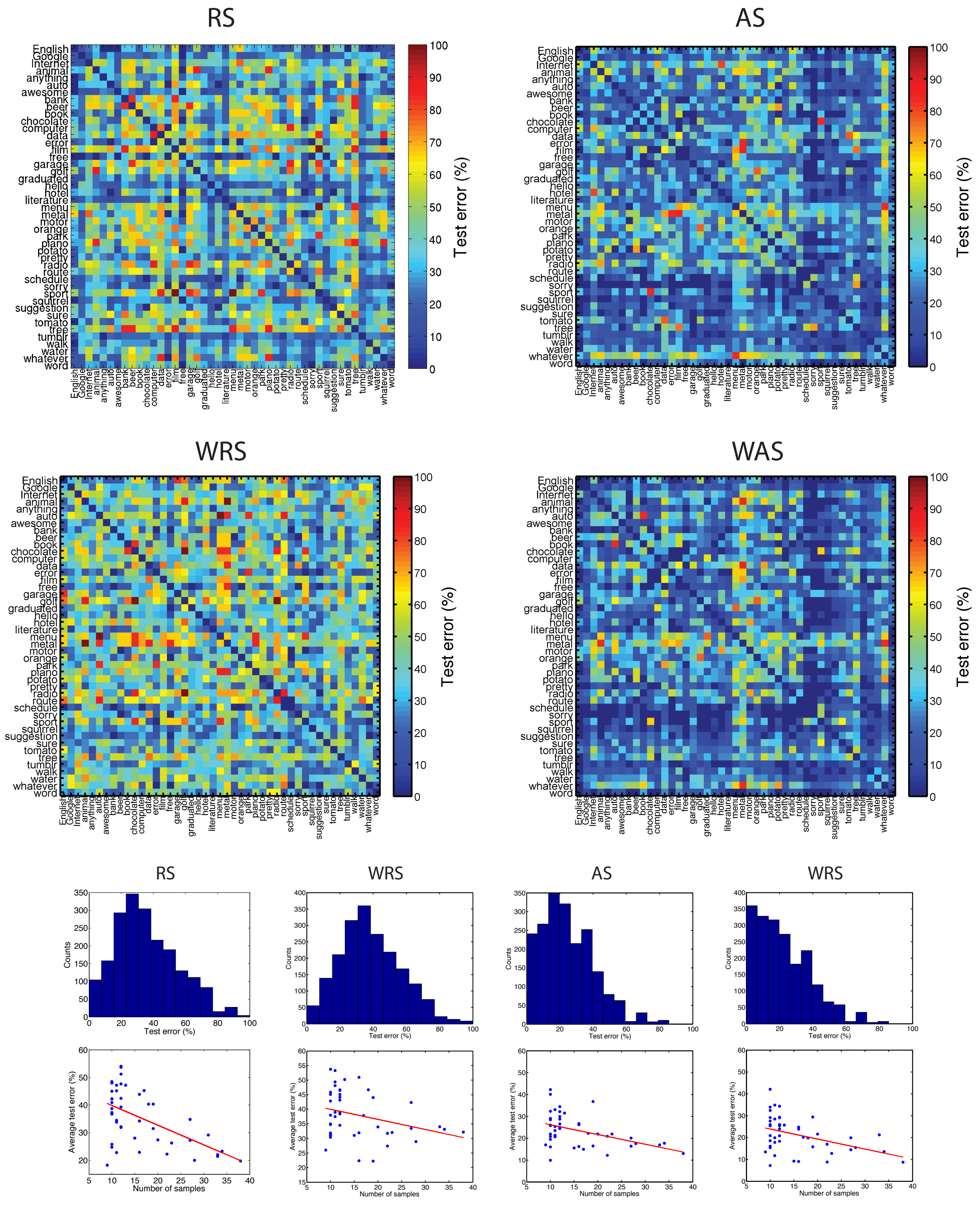

Figure 3: *Results of word-pair classification for four different features: Raw Spectrograms (RS), Aligned Spectrograms (AS), Wavelet transform on Raw Spectrograms (WRS), and Wavelet transform on Aligned Spectrograms (WAS). Colored images are confusion matrices, showing the test error for each word pair. Histograms are distributions of test errors for each feature. Scatter plots show the variation of the average test error for each word as a function of the number of samples for that word. Red lines are linear fits.*