ISCA Archive
http://www.isca-speech.org/archive

Speech and Language Technology in Education (SLaTE2011)
Venice, Italy
August 24-26, 2011

Using an Ensemble of Classifiers for Mispronunciation Feedback

G. Ananthakrishnan[1], *Preben Wik*[1], *Olov Engwall*[1], *Sherif Abdou*[2]

[1]Centre for Speech Technology, KTH (Royal Institute of Technology), Stockholm, Sweden
[2]Faculty of Computers & Information, Cairo University, Egypt
{agopal, preben, engwall}@kth.se, s.abdou@fci-cu.edu.eg

Abstract

This paper proposes a paradigm where commonly made segmental pronunciation errors are modeled as pair-wise confusions between two or more phonemes in the language that is being learnt. The method uses an ensemble of support vector machine classifiers with time varying Mel frequency cepstral features to distinguish between several pairs of phonemes. These classifiers are then applied to classify the phonemes uttered by second language learners. Instead of providing feedback at every mispronounced phoneme, the method attempts to provide feedback about typical mispronunciations by a certain student, over an entire session of several utterances. Two case studies that demonstrate how the paradigm is applied to provide suitable feedback to two students is also described in this paper.
Index Terms: Support Vector Machines, Time Varying-MFCC, CAPT

1. Introduction

Computer Assisted Pronunciation Training (CAPT) is a fast growing and an important aspect of Computer Assisted Language Learning (CALL) systems. While many CAPT systems have focussed on providing intelligent training methods to help improve pronunciation, some other systems rely on providing feedback to the users [1]. A few methods use Automatic Speech Recognition (ASR) to give implicit or explicit feedback regarding the quality of the pronunciation. Implicit feedback is provided when the ASR is not able to recognize the pronounced word correctly, unless it is pronounced with a native accent. Explicit feedback can be in the form of a score, for example, Goodness of Pronunciation (GOP) [2, 3] which gives the students a reasonable feedback on their performance. While many such CAPT systems provide scores that are highly correlated to scores provided by humans [4], this sort of feedback does not tell the student what the specific mistake is or what he or she should change in the pronunciation in order to get a higher score [5]. Other systems like the ISLE project [6] try to provide explicit feedback based on the specific first language (L1) and second langauge (L2) pair, by highlighting the locus of the error in the word. Using such explicit knowledge, specific and typical pronunciation errors may be accurately detected, but idiosyncratic errors are often ignored.

Several machine learning approcahes like Hidden Markov Models (HMM) [7], Linear Discriminant Analysis (LDA) [8] and Support Vector Machines [9] have been used to detect errors as well as to provide feedback. While reported performances of such methods in CAPT systems have been improving rapidly, the low accuracy of such feedback systems may discourage students from utilizing them [4].

Another approach commonly used in CAPT systems is to provide feedback on the fluency of the speaker on a global scale (e.g., [10]). The score can be given for a sentence, or group of utterances, based on segmental, or prosodic features. Some of the studies use known texts, while others make use of the students L1 to make sound judgements about the overall fluency. These systems, while useful in automatic language testing systems, may not be particularly useful as feedback to the students in helping them improve.

We follow the approach used by Truong [8] who used a set of binary classifiers, to help classify often confused phonemes. The above study required careful selection and construction of the acoustic parameters in order to make reliable classifications and claimed detection accuracies of somewhere between 70 and 90 %. They also tried to train their classifiers on native as well as non-native speech, and found that the performance, as expected, was better on native speech, which in general showed lower variance.

In our approach, we extend this methodology to include a very large number of classifiers. This requires a method in which the same classification system should in principle hold for classifying several classes of pairs of phonemes. Since different kinds of acoustic features are useful for classifying different types of phonemes, including static as well as dynamic sounds, our approach requires a common platform to select the suitable features automatically. Secondly, the accuracy when classifying different types of phonemes would also be largely different. To side-step this problem, we do not endeavor to provide feedback on every incorrect utterance, but instead make a judgement on an entire session of utterances (typically around 80). We compare the performance of the classifiers on native speech (assuming the native speech to be correct). We provide specific feedback only on those phonemes on which the classifiers report significantly higher error rates than on native speech. Thus, even though the feedback is on a global level, the feedback is specific enough to highlight the problem areas for the student to work on. This also provides a means for personalizing the CALL system to provide increased training in specific sounds deemed confusing for the particular student.

2. Ensemble of Classifiers

The block diagram of the ensemble classification framework we used in this study is illustrated in Figure 1, previously described in [11]. Given the acoustic signal and the text of what the subject is supposed to have uttered, the acoustic signal is segmented into the sequence of phonemes using an HMM based alignment [12]. We use the native speech for training our models and test them on non-native speech uttered by the L2 language learner. The input to the classification framework are the acoustic segments of individual phonemes. Since the same framework should be applicable to both static as well as transient sounds, we use acoustic features that also capture the dynamics of the phoneme, namely Time-Varying Mel Frequency Cepstral Coefficients (TV-MFCC). Different pairs of phonemes are optimally distinguished by different sets of acoustic parameters. One, therefore, needs to select the right combination of features.

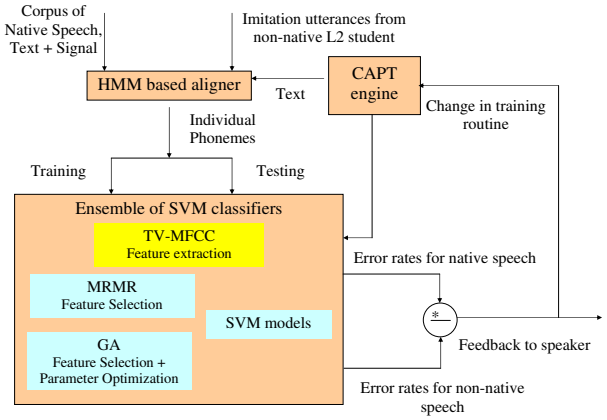

Figure 1: Block Diagram of our system using an ensemble of binary classifiers, to detect specific mispronunciation errors, by finding significant performance differences in the performance on native speech and non-native speech.

We use one filter based algorithm, namely Minimum Redundancy Maximum Relevance (MRMR) [13], and one wrapper based, Genetic Algorithms (GA) [14] to achieve this. These features are then used to train SVM models with Gaussian kernels, one of the better performing machine learning algorithms for binary class problems. The classifier system, thus, consists of 4 components as described below.

2.1. Acoustic Features

Selecting suitable acoustic features in order to classify a wide range of segmental errors is a difficult problem. The often used Mel Frequency Cepstral coefficients (MFCC), represent the frequency spectrum and its perception by humans rather accurately. However, MFCCs consider each frame of the acoustics to be stationary, which may not be the case for certain transient sounds like plosives. We, therefore, used dynamic features in the Time-varying MFCC. If $A(f,t)$ is the time varying log spectrum of the audio signal, with F mel frequency subbands and T time samples, then $\tilde{A}(p,q)$ are the 2D-DCT coefficients obtained by performing DCT along the dimensions f (frequency) and t (time). The dimensions $p:\{1 \leq p \leq P\}$ and $q:\{1 \leq Q\}$ are called 'quefrency' and 'meti' respectively.

$$\tilde{A}(p,q) = \frac{1}{T}\sum_{f=1}^{F}\sum_{t=1}^{T} A(f,t) \quad cos\left(\frac{\pi(f-(1/2))(p-1)}{F}\right) \quad cos\left(\frac{\pi(t-(1/2))(q-1)}{T}\right) \quad (1)$$

$\tilde{A}$ is then converted into a vector, which is the acoustic feature for a phoneme. This formulation is time-invariant, in the sense that stretching or contraction of the signal does not change $\tilde{A}$. However, since the duration of the segment is also an important parameter, the duration is added at the end of the vector. Thus, the total number of features are $P * Q + 1$.

2.2. Minimum Redundancy Maximum Relevance

MRMR, being a filter based algorithm, the feature selection process is not connected to classification accuracy. The features are sorted according to the inherent relationships between the features and their discriminative abilities. It relies on estimating feature redundancy (selecting features that are dissimilar to each other) and relevance (maximizing the contribution of the features towards classification) using mutual entropy, through a greedy search. Processing time varies linearly with the number of features to be retained. Even though it does not tie the selection procedure to improvements in the classification, a hypothesis about the error rate is required for calculating the mutual information. This is provided by the SVM using the whole feature set. We select N features which best satisfy the minimum redundancy maximum relevance criteria. This method reduces the search space by a large amount, and thus the time taken for the GAs to converge.

2.3. Genetic Algorithms

While MRMR returns a compact set of N best features, they may not be the best combination of features for the task. In order to ensure optimal performance for the SVMs, the poor set of features needs to be discarded. For this we use a wrapper based feature selection algorithm, which ties a particular combination of selected features to the performance of the classifier. We employ an implementation of GA [14] with a K-fold cross validation scheme in order to select both the feature indices and to optimize the parameters of the SVM. Without good initial estimates, GAs are known for their long convergence times. With an initial number of, for example, 55 features, the search space has roughly 2^{55} combinations. By forcing the initial feature population of the GA to correspond to the first few features selected by MRMR, we can constrain the search space, provide a good initial guess and thereby help the GA converge faster. We also perform a search of the SVM parameters along with the search for optimal features within the GA algorithms, i.e. the variance of the gaussian kernels and the penalty for tolerating misclassified instances during the training phase.

2.4. Support Vector Machines

SVM models are trained on native speech, using a K-fold cross-validation different folds, applying MRMR to each fold, then applying the genetic algorithm to find the optimal features and the classifier parameters over all the folds. The optimum features and parameters are then used to train the SVM binary classifiers over each fold. The error rates over each fold is calculated using the optimal set of selected features and parameters.

3. Data and Experiments

The corpus consisted of 78 phonetically rich short sentences and words uttered by 11 native Swedish speaking subjects who recorded the utterances from a text displayed on the screen. This was done using a desktop microphone and a sampling frequency of 16 KHz. This was used for training 95 binary classifiers using the process described in Section 2.4. The data for the non-native speech consisted of 2 to 7 students each from 11 different L1 backgrounds, learning to speak Swedish in a Swedish course. The students used the VILLE Swedish virtual language tutor and produced the utterances while trying to mimic the words or sentences uttered by the virtual tutor. More details on the test subjects and their performances before and after the training using VILLE are described in [15]. The data was cleaned up to remove instances of hesitations or completely incorrect utterances in the data. In this experiment, the native and non-native speakers recorded the same set of utterances, but in principle, they can be completely different.

In total, 95 pairwise classifiers under 6 categories were created. The 6 categories were

1. Plosive *vs.* Fricative (PF) (6 pairs)
2. Voiced *vs.* Unvoiced consonants (VU) (5 pairs)
3. Front *vs.* Back vowels (FB) (23 pairs)

4. Short *vs.* Long vowels (SL) (11 pairs)
5. Unrounded *vs.* Rounded Lips (UR) (23 pairs)
6. Open *vs.* Closed vowels (OC) (27 pairs)

Under each category, all possible confusable pairs of phonemes were considered. For each phoneme, TV-MFCCs, with the number of quefrency coefficients, $P = 18$, and meti coefficients, $Q = 3$, were extracted. The total number of features were 55 initially, including the duration of the phoneme. Since further feature selection was performed on these features, we did not experiment with changing the number of initial features. At the first stage, MRMR was performed to select the best 20 features with respect to the particular binary classification task. Optimization was then performed using GA with a 4-fold stratified cross-validation (in such a way that we have different speakers in the training and testing, even though the number of speakers in each fold may be different), and the best features and classifier parameters were chosen. For vowel and vowel like sounds, most of the features that were chosen were static features (i.e., along the quefrency dimension rather than the meti dimension), while more dynamic features were selected for transients and other consonants. The duration parameters was selected in most of classifications in the Long *vs.* Short category. The number of parameters selected for the specific classification task ranged from 3 to 20. The time taken to build each classifiers ranged from less than a second to 26 minutes, depending on the number of samples available for the respective phonemes, in a MATLAB^{TM} implementation of the algorithms. After the best set of features were picked, a 4-fold cross-validation was again performed to gauge the performance of the classifiers on native speech. The error rate of the classifiers ranged from 0 to 34%, assuming that the natives had a perfect pronunciation. The worst performing category was the Short *vs.* Long vowel category. For every L2 learner (student), all the phoneme boundaries were extracted using the HMM based alignment using the phonemic transcription of what they were supposed to have uttered. For every classifier, all the relevant phonemes were chosen from the entire session of each student using VILLE, and classified using the ensemble of classifiers.

Classifiers performing with an accuracy of around 70% on native speech would normally not be useful for providing pronunciation feedback on non-native speech. Therefore, we adopted a method to side-step the problem. A binomial significance test was conducted to see if the error rate estimated by each classifier on the said student's speech was significantly higher than on the native speech. It was clear that the student would have problems with those phoneme pairs for which the estimated classifier error was significantly higher than that for native speech. Thus, even if the classifier error rate is quite high on native speech, it could still be useful at providing suitable feedback to the L2 learner. In this study we only took those examples for which the significance levels were greater than 0.99. We, however, expect that with training, the significance levels can be lowered for each student. Another parameter that can be chosen to make assessments is the difference between the error rate on native and non-native speech. Based on the different combinations of phoneme pairs that are significantly worse than for native speech, several inferences can be drawn. This is illustrated in the following case studies.

3.1. Case Study 1

This is a case study for a male speaker 'J2' with an L1 background of American English. The list of phoneme classes for which the classifiers estimated a significantly higher error rate, along with their respective categories, are displayed in Table 1. There are several observations that reveal systematic information. The student appears to have 4 main problems, namely a problem of voicing difference between the phonemes /k/ and /g/, confusions between phonemes /ɑː/ and /oː/, /ɛ/ and /e/ and between /oː/ and /ʉ/. These pairs have the largest difference in error rates between native and non-native speech. However, some interesting inferences can also be made by looking at other pairs showing significantly higher error. For example, voicing may be a general problem for a few other pairs, where the amount of voicing is not sufficient to distinguish it from the unvoiced phoneme. Since the error difference is smaller, this problem may have a lower priority. On the other hand, there seem to be several confusions of the back vowel /oː/ with several similar sounding front vowels. Thus, the focus should be on training the pronunciations of front rounded vowels, which the student seems to have a problem pronouncing, rather than on the specific pairs denoted in the table. The problem with phonemes /ɑː/ and /oː/ indicates that it is more likely that /oː/ is pronounced with a more opened jaw than the other way round. The problems with /ɛ, ɛː, e, ə/ and /eː/ seem to be a systematic confusion, since almost all pairs between these phonemes have significantly higher error rates. Although many of these phonemes do occur in the student's L1, the problem he faces could be because the rules that determine which of these sounds are to be produced, in a certain phonetic context are different for Swedish and American English. The same is the case between phonemes /oː, o/ and /u, uː/, where confusions occur both ways.

Table 1: *The classifiers that show significantly higher errors for the student J2 (male with American English as L1) than on native speech.*

Category	Phoneme pair (IPA)	Phoneme tiny from the pair with significant error diff.	Difference in Error %
VU	b, p	b	7
VU	v, f	v	5
VU	g, k	g	13
VU	g, k	k	4
FB	ʉː, u	ʉː	4
FB	yː, oː	oː	5
FB	ø, oː	oː	5
FB	œ, oː	oː	6
FB	øː, oː	oː	3
UR	ə, œː	œː	5
UR	ɛː, œː	œː	3
OC	ɑː, oː	oː	12
OC	ɛ, e	e	11
OC	ɛː, eː	ɛː	8
OC	ɛː, eː	eː	7
OC	ɛː, ə	ɛː	8
OC	ɛː, ə	ə	5
OC	oː, u	u	13
OC	o, uː	uː	5

3.2. Case Study 2

The second case study is for a female speaker 'I2' with an L1 background of Spanish. As shown in Table 2, there are some problems areas that are similar to the previous case, for example with phonemes /ɛ, ɛː, e, ə/ and /eː/. In fact, the error rate difference between native and student I2's speech is higher than for student J2. We can also observe certain additional problem areas, for example with the plosive /b/ and fricative /v/. This is known to be a problem for Spanish speakers, who do not make this distinction. A similar problem, but maybe to slightly lesser extent occurs in distinguishing between the plosive /t/ and the fricative /ɕ/. Then again, the voicing timing for I2 (observed also for other Spanish speakers in our database) appears to be

Table 2: *The classifiers that show significantly higher errors for the student I2 (female with Spanish as L1) than on native speech.*

Category	Phoneme pair (IPA)	Phoneme from the pair with significant error diff.	Difference in Error %
PF	b, v	b	13
PF	t, ɕ	t	4
VU	b, p	b	15
VU	g, k	g	17
FB	ʉː, u	ʉː	8
FB	ʉː, u	u	9
FB	yː, oː	oː	11
FB	ø, ɑː	ɑː	9
FB	œ, ɑː	ɑː	19
FB	øː, oː	oː	8
FB	øː, oː	øː	9
FB	œː, oː	oː	11
FB	œː, oː	œː	16
UR	ə, œː	ə	14
UR	ɪː, yː	yː	8
UR	ɪː, yː	ʉː	16
OC	ɑː,oː	oː	12
OC	ɛ, e	e	25
OC	ɛ, e	ɛ	5
OC	ɛː, eː	eː	7
OC	ɛː, eː	ɛː	6
OC	ɛː, e	e	8
OC	ɛ, eː	ɛ	5
OC	oː,uː	uː	13

even more different from that of native Swedish speakers than that of subject J2. This student also seems to have a problem with lip rounding for the vowels /ɪː, y/ and /ʉː/ as well as distinguishing between /oː and /uː/.

4. Conclusions and Future Work

The paper describes a framework for detecting segmental mispronunciation errors and providing suitable feedback to both the language learner as well as the CAPT system. The system does not make any assumptions on the L1 of the user, nor is it constrained to detecting the errors on a limited set of sentences or words. The only constraint is that the text of what the L2 learner tried to pronounce is available to the system. For this reason, we used data where the student repeats an utterance made by the CAPT system.

The detection as well as feedback is provided over several utterances, rather than over the immediate instance of a mispronunciation. In order to counter problems related to detection errors in recognizers or classifiers, the framework makes a statistical comparison between the error rates over native Swedish speech and the L2 Swedish speech. As a threshold, only phonemic pairs with a significantly higher error rate than that for native speech are considered for providing mispronunciation feedback. This not only balances the varying performances of the classifiers over different phoneme classes, but also reduces the likelihood of erroneous detections. However, there is no guarantee that all errors are captured at a certain significance level. It may, thus, be necessary to reduce the statistical significance level in order to capture pronunciation errors that do not occur very frequently.

The framework, in principle, can also be applied for providing real-time feedback to students. This, however, would be restricted to only the classifiers with a sufficiently low error rate on native speech. The output of the framework can be used in several ways. One direct use would be to increase the number of CAPT examples from the most confused pair of phonemes, or by employing minimal-pair words to elucidate the difference. However, the results could be used more intelligently, to provide a deeper insight into the problems the student faces in the L2 pronunciations. Examples of such insight were discussed in Sections 3.1 and 3.2. This may, however, not be trivial to accomplish automatically. Our future research is directed towards developing a reasonable logic to provide this insight automatically and deploying it within the VILLE framework to see if this paradigm helps students to improve their L2 pronunciations.

5. Acknowledgements

We would like to thank the Swedish Research Council projects 80449001, Computer-Animated Language Teachers (CALaTea) and 348-2005-6161, Audiovisual Detection of Errors in Pronunciation Training (ADEPT) for financial support. We would also like to acknowledge the help of Chris Koniaris for processing the data which we used in this study.

6. References

[1] Eskenazi, M., "An overview of spoken language technology for education," Speech Communication, 51(10):832–844, 2009.

[2] Witt, S. and Young, S., "Phone-level pronunciation scoring and assessment for interactive language learning," Speech communication, 30(2-3):95–108, 2000.

[3] Kanters, S., Cucchiarini, C., and Strik, H., "The Goodness of Pronunciation algorithm: a detailed performance study," Proc. SLaTe, 2009.

[4] Hincks, R., "Speech technologies for pronunciation feedback and evaluation," ReCALL, 15(01):3–20, 2003.

[5] Neri, A., Cucchiarini, C., and Strik, H., "Feedback in computer assisted pronunciation training: technology push or demand pull?," in Proceedings of ICSLP. Citeseer, 1209–1212, 2002.

[6] Menzel, W., Herron, D., Bonaventura, P., and Morton, R., "Automatic detection and correction of non-native English pronunciations," in Proc. InSTILL, 49–56, 2000.

[7] Bunnell, H., Yarrington, D., and Polikoff, J., "STAR: articulation training for young children," in Proc. ICSLP. Citeseer, 4:85–88, 2000.

[8] Truong, K., "Automatic pronunciation error detection in Dutch as a second language: an acousticphonetic approach," M.S. thesis, Utrecht University, The Netherlands, 2004.

[9] Wei, S., Hu, G., Hu, Y., and Weng, R.-H., "A new method for mispronunciation detection using Support Vector Machine based on Pronunciation Space Models," Speech Communication, 51:896–905, 2009.

[10] Moustroufas, N. and Digalakis, V., "Automatic pronunciation evaluation of foreign speakers using unknown text," Computer Speech & Language, 21(1):219–230, 2007.

[11] Picard, S., Ananthakrishnan, G., Wik, P., Engwall, O., and Abdou, S., "Detection of Specific Mispronunciations using Audiovisual Features," in Proc. Int. Conf. on Auditory-Visual Speech Processing, Kanagawa, Japan, 2010.

[12] Sjölander, K. and Heldner, M., "Word level precision of the NALIGN automatic segmentation algorithm," in Proc. of Fonetik, 116–119, 2004.

[13] Peng, H., Long, F., and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," IEEE Transactions on pattern analysis and machine intelligence, 1226–1238, 2005.

[14] Goldberg, D., *Genetic algorithms in search, optimization, and machine learning*, Addison-wesley, 1989.

[15] Wik, P., *The Virtual Language Teacher: Models and applications for language learning using embodied conversational agents*, Ph.D. thesis, Centre for Speech Technology, KTH (Royal Institute of Technology), 2011.