Detecting confusable phoneme pairs for Swedish language learners depending on their first language

G. Ananthakrishnan, Preben Wik, Olov Engwall

Centre for Speech Technology, KTH

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. Using this method, an assessment is made regarding the typical pronunciation problems that students learning Swedish would encounter, depending on their first language.

Introduction

Computer Assisted Pronunciation Training (CAPT) is a fast growing and an important aspect of Computer Assisted Language Learning (CALL) systems. However, problems faced by student with different first language (L1) backgrounds are often very different. At the same time some of the problems are common to almost all language backgrounds. In the context of Computer Assisted Pronunciation Training (CAPT), this aspect is very relevant. Bannert (1980) pointed out what sounds and phonemes in Swedish may be confusing to students depending on their L1 and found a large variation. However, this study was made based on expert knowledge of a trained phonetician, based on students with classical pronunciation problems. In this paper, we describe an automated method that can extract such knowledge from data collected from several second language (L2) students. Such explicit knowledge can be used to increase the accuracy of detecting specific types of pronunciation errors, as well as developing customized training methods for students with a particular L1 background.

We follow the approach of Truong (2004) 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 in order to assess what kind of pronunciation errors and confusions are most probable, given the L1 of the student. 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 make assessments on every incorrect utterance, but instead make a judgement on entire sessions of utterances (around 80) of several speakers (2 to 24). We compare the performance of the classifiers on native speech (assuming the native speech to be correct). We assess only on those phoneme pairs, on which the classifiers report significantly higher error rates for L2 learners than on native speech, to be problematic.

Ensemble of Classifiers

The block diagram of the ensemble classification framework we used in this study is illustrated in Figure 1, previously described in (Picard et al., 2010). 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 Hidden Markov Model (HMM) based alignment (Sjölander and Heldner, 2004). We use the native speech for training our models and test them on non-native speech ut-

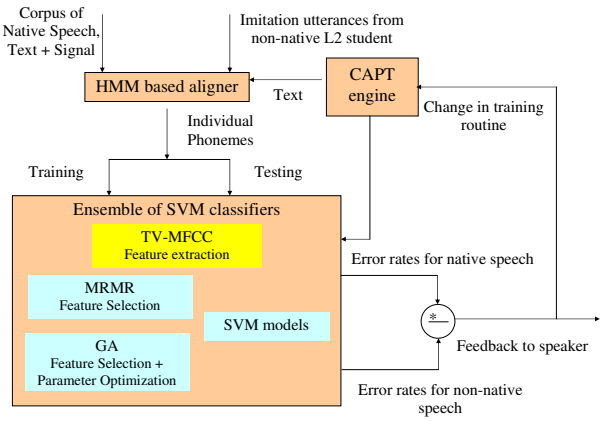

Figure 1: Block Diagram of our system using an ensemble of binary classifiers, to detect specific mispronunciation errors, by finding significant performance differences in the performance on native speech and non-native speech.

tered by the L2 language learners. The input to the classification framework are the acoustic segments of individual phonemes. The classifier system consists of 4 components, described below.

Acoustic Features

In order to model static as well as transient sounds, we used dynamic features in the form of ime-Varying Mel Frequency Cepstral Coefficients (TV-MFCC). If $A(f,t)$ is the time varying log spectrum of the audio signal, with F mel frequency sub-bands 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 (Picard et al., 2010). The duration is added at the end of the vector. Thus, the total number of features are $P * Q + 1$.

Minimum Redundancy Maximum Relevance

Since we use many different kinds of binary classifiers, the most relevant and optimum acoustic features are not always the same. We therefore use two feature selection algorithms. Minimum Redundancy Maximum Relevance (MRMR) (Peng et al., 2005) 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. This method reduces the search space by a large amount, and thus the time taken for the Genetic Algorithms (GAs) to converge.

Genetic Algorithms

In order to ensure optimal performance for the binary classifiers, the poor set of features needs to be discarded. Besides, the optimum parameters for each classifier would also be different. We, therefore, employ an implementation of GA (Goldberg, 1989) with a K-fold cross validation scheme in order to select both the feature indices and to optimize the parameters of the classifier.

Support Vector Machines

Support Vector Machine (SVM) try to find the best possible hyper-plane separating two classes, by maximizing the distance between the elements of the two classes. SVMs are known to perform well for binary classification problems even on sparse and high dimensional data. We also use the Gaussian kernel in order to allow non-linear hyper-planes. The 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.

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 reading a text displayed on the screen. This was done using a desktop microphone and a sampling frequency of 16 KHz. The material was used for training 95 binary classifiers using the process described in the above section. The non-native speech data consisted of 2 to 24 students each from 11 different L1 backgrounds, learning to speak Swedish in a Swedish course. The students used the VILLE Swedish virtual language tutor (Wik et al., 2009; Wik and Hjalmarsson, 2009) and produced the utterances while trying to repeat the words or sentences uttered by the virtual tutor. 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. 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 cross-validation, and the best features and classifier parameters were chosen. 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. The error rate of the classifiers on native speech 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. All the relevant phonemes were chosen from the entire session of each students from a particular L1 background 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 utterances by students with a said L1 background was significantly higher than on the native speech. Thus, even if the classifier error rate is quite high on native speech, it could still be useful for providing suitable assessment. In this study we illustrate only three examples per L1 background, with the highest probability of the error rate being significantly higher.

Results and Discussion

Table 1 displays the three most likely pairs of Swedish phonemes that could cause confusing pronunciations from students of each L1 background. It should be noted that this table does not provide an exhaustive list of confusions, but only the three phonemes pairs with largest significant classifier error rates compared to native speech. Again, since the sample set of students was rather small in some cases, the examples may not be indicative of the entire language group.

From a cursory glance at the Table 1, it is clear that most of the errors are made in the Swedish vowels. The distinction between /ɛ/ *vs.* /ə/ is most confusing to students from almost all language backgrounds. Some other recurrent confusable pairs are /ɑː/ *vs.* /oː/ and /ɛː/ *vs.* /e/. Most of the errors made on consonants were in voicing. Several language groups like Polish, Greek and Persian had significantly larger error rates from the classifiers than native Swedish pronunciation. Polish stop consonants can be hard or soft depending on the vocalic context. This difference in pronunciation may be the source of errors while pronouncing Swedish sounds. For Greek, the unvoiced stop consonant disappear in the context of nasal sounds. This may be the source of errors for Greek pronunciations. The high error rates for Persian voiced and unvoiced distinctions are, however, puzzling. This may, however, be due to different ranges of voicing onset for Persian and Swedish stop consonants. Students with a Spanish L1 background are known to make errors when producing the sound /oː/ and /ɔ/, which is reflected in the confusions, /øː/ *vs.* /ɔ/ and /ɑː/ *vs.* /oː/. Some language backgrounds indicate significantly larger classifier errors for lip rounding distinctions, such as ə → œː for Arabic and Polish L1 backgrounds.

This method, thus, automatically lists problem areas for students with different L1 backgrounds, which will help the CAPT system to provide a customized training material according to the apriori knowledge. The power of this paradigm is the ability to circumvent the low accuracies of certain classifiers as well as to locate specific problems. The paradigm can be extended to detecting specific cases of mispronunciations. However, not all the classifiers may be accurate enough to be used in specific detections.

Acknowledgements

We would like to thank the Swedish Research Council projects 80449001, Computer-Animated Language Teachers (CALaTea) for financial support. We would also like to acknowledge the help of Chris Koniaris for processing the data which we used in this study.

Table 1: Illustration of the three most common confusions that students with different L1 backgrounds, learning Swedish as an L2, as estimated by our algorithm. The arrow indicates in which direction the confusion is detected to occur.

L1 Background (No. of students)	Error Category	Confusable phonemes (IPA)	Examples	Diff. in Error (%) between native and non-native speech
American English (4)	VU	ɡ → k	**g**ås → **k**al	12%
	OC	æː ← ə	h**ä**r ← pojk**e**n	11%
	OC	oː → ʊ	h**å**l → b**o**tt	10%
Arabic (2)	OC	ɛ ← ə	r**ä**tt ← pojk**e**n	18%
	PF	b → v	**b**il → **v**år	16%
	UR	ə → œː	pojk**e**n → f**ö**r	15%
Mandarin Chinese (10)	VU	ɡ → k	**g**ås → **k**al	17%
	OC	ɑː ← oː	h**a**l ← h**å**l	13%
	OC	æː ← ə	h**ä**r ← pojk**e**n	12%
French (4)	OC	ɛ → ə	r**ä**tt → pojk**e**n	10%
	UR	ə → ø	pojk**e**n → f**ö**ll	9%
	OC	ɔ ← ʊ	h**å**ll ← b**o**tt	8%
German (3)	VU	ɡ → k	**g**ås → **k**al	11%
	OC	ɔ → ʊ	h**å**ll → b**o**tt	9%
	OC	ɛː ← e	r**ä**tt: ← v**e**tt	8%
Greek (5)	OC	ɛ → ə	r**ä**tt → pojk**e**n	8%
	OC	ɛ ← ə	r**ä**tt ← pojk**e**n	8%
	OC	ɔ ← ʊ	h**å**ll ← b**o**tt	7%
Persian (24)	VU	d → t	**d**al → **t**al	14%
	OC	ɑː ← oː	h**a**l ← h**å**l	11%
	VU	ɡ → k	**g**ås → **k**al	11%
Polish (4)	VU	ɡ → k	**g**ås → **k**al	30%
	VU	d → t	**d**al → **t**al	24%
	UR	ə → œː	pojk**e**n → f**ö**r	20%
Russian (7)	OC	ɑː ← oː	h**a**l ← h**å**l	11%
	OC	ɔ ← ʊ	h**å**ll ← b**o**tt	11%
	OC	ɛ ← ə	r**ä**tt ← pojk**e**n	9%
Spanish (11)	FB	øː ← ɔ	f**ö**ll: ← h**å**ll	14%
	OC	ɛ ← ə	r**ä**tt ← pojk**e**n	14%
	OC	ɑː ← oː	h**a**l ← h**å**l	12%
Turkish (7)	OC	ɑː ← oː	h**a**l ← h**å**l	20%
	FB	œː ← ɑː	f**ö**r ← h**a**l	10%
	OC	ɛ ← ə	r**ä**tt ← pojk**e**n	10%

References

Bannert R (1980). *Svårigheter med svenskt uttal: inventering och prioritering*. Inst. f ”or lingvistik, Lunds univ.

Goldberg D (1989). *Genetic algorithms in search, optimization, and machine learning*. Addison-wesley.

Peng H, Long F and Ding C (2005). 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.

Picard S, Ananthakrishnan G, Wik P, Engwall O and Abdou S (2010). Detection of Specific Mispronunciations using Audiovisual Features. In *Proc. Int. Conf. on Auditory-Visual Speech Processing*. Kanagawa, Japan.

Sjölander K and Heldner M (2004). Word level precision of the NALIGN automatic segmentation algorithm. In *Proc. of Fonetik*, 116–119.

Truong K (2004). *Automatic pronunciation error detection in Dutch as a second language, an acoustic-phonetic approach*. Master’s thesis, Utrecht University, The Netherlands.

Wik P, Hincks R and Hirschberg J (2009). Responses to Ville: A virtual language teacher for Swedish. In *Proc. SLaTE*. Wroxall Abbey Estates, UK: Citeseer.

Wik P and Hjalmarsson A (2009). Embodied conversational agents in computer assisted language learning. *Speech communication*, 51(10):1024–1037.