

# Pronunciation Modeling in Speaker Detection

David Klusáček<sup>†</sup>  
Charles University  
11. 12. 2003

## Abstract

Herein is a proposal to extend research started at the 2002 Workshop in Speaker Detection at the Johns Hopkins University Center for Speech and Language Processing. My aim is to further explore pronunciation modeling approach discovered during the workshop. More specifically I will focus on robustness in noisy conditions and faster implementation which will allow to test on SWB2 after it will be developed on SWB1. I will use the standard NIST extended data evaluation, so the results will be comparable with other methods.

## 1 Introduction

Conditional pronunciation modeling is a method for open-set speaker detection task that does not rely on acoustic vectors. Instead it tries to use higher-level information carried by the speech signal. This gives us hope that it will be more robust to signal distortion than conventional GMM-UBM acoustic approach.

In the following, there will be described what has been done during the CLSP Summer Workshop 2002 in pronunciation modeling. By "pronunciation model" is meant an ambiguous "mapping" from phonemes that the speaker intends to pronounce to phones he actually pronounced. Since we don't know speaker's intention we use ASR to decode words spoken followed by lookup of phonemes in the lexicon (one word can have multiple lexicon pronunciations). Then these lexicon pronunciations are force-aligned with audio and the one with the highest score is selected as the phonetic transcription of the word. This is how time aligned phoneme stream is computed. Phones actually pronounced are taken from open-loop phone recognizer. We could have studied open-loop stream only, as other phonetic methods do, but studying relation of what has been said (phonemes) versus how it has been said (phones) seems to have more discriminative power. This is because phones alone are quite ambiguous so it can happen that two different phonemes maps into single phone. Conditioning phones on phonemes avoids this situation and gives us a "statistic footprint" of the phoneme to phone mapping. Individual pronunciation of phones as well as some coarticulation habits are encoded in this footprint. Example of these streams (for utterance fragment "to you") is shown in the following table:

---

<sup>†</sup> klusacek@atrey.karlin.mff.cuni.cz

FRAME	ASR	EG	WORD
24964	t	n	TO
24965	t	s	
24966	t	s	
24967	t	s	
24968	t	s	
24969	t	s	
24970	t	s	
24971	ax	I	
24972	ax	I	
24973	ax	I	
24974	y	j	YOU
24975	y	j	
24976	y	j	
24977	uw	j	
24978	uw	u	
24979	uw	u	
24980	uw	u	
24981	uw	u	
24982	uw	silx	
24983	uw	silx	

Where FRAME is a frame number, WORD is a recognized word, ASR is a phoneme stream and EG is open-loop phone stream.

Pronunciation modeling estimates conditional probabilities of EG given ASR (that is "reality" given the "intention") on per frame basis. Then, standard likelihood ratio detector is applied to these probabilities, as will be described in the next two sections.

## 2 Training

Since we are using likelihood ratio detector we have two models: the background model and the speaker model. Both consist of conditional probabilities of EG given ASR and are trained in the same way:

$$P(\text{EG} = e | \text{ASR} = a) = \frac{\#((e, a) \text{ appears in the INPUT})}{\#((- , a) \text{ appears in the INPUT})}$$

Where "-" means any EG phone. INPUT is a stream of (ASR,EG) pairs as shown in the introduction. We further assume that INPUT (both in training and testing) has been filtered by removing frames ASR marks as a silence and frames containing crosstalk. No thresholding nor smoothing is being used in the current implementation.

## 3 Testing

Score of the test utterance is computed as follows:

$$\text{score} = \sum_{(e, a) \text{ from INPUT st both } P_{\text{SP}}(e, a) \text{ and } P_{\text{BG}}(e, a) \text{ are defined}} (\log(P_{\text{SP}}(e|a)) - \log(P_{\text{BG}}(e|a)))$$

This means that only those pairs  $(e, a)$  that has been seen during the training of both speaker and background model are counted. Also, we treat INPUT as an array, so unlike if it was a set, if there is a pair  $(e, a)$  in the INPUT occurring  $N$ -times it will be counted  $N$ -times in the sum.

## 4 Experiments and Results

All experiments during the Workshop were conducted on SWB1 according to NIST extended data task. Previously described method has been independently run using open-loop recognizers of 5 different languages (EG GE SP JP MA – the PPRLM recognizers) resulting in 5 score values. Total score was computed as a sum of those values. Let's look on some of the experiments (all were evaluated on on SWB1, splits 1–6, trained on 8 conversations):

### Experiment 1: **Conditioning works**

This experiment should show that conditioning on ASR is really helpful. For that purpose we run the same algorithm but with ASR stream modified such that on non-silence frames it returned always the same value (different from "sil") instead of phoneme transcription. This unconditioned approach (summing scores from streams GE SP JA MA) had 7.0% EER, whereas GE SP JA MA given ASR achieved 2.7% EER.

### Experiment 2: **ASR may be replaced by open-loop**

Main disadvantage of the method is that it requires the use of the ASR. The purpose of this experiment was to see how much we loose if something easier than ASR is used. The most simple case is another open-loop recognizer. GE SP JA MA given EG achieved 4.3% EER compared to 2.7% of the original system.

### Experiment 3: **Best results**

Best results were achieved with slightly different setup. Instead of conditioning on ASR phoneme it would make more sense to condition to HMM state of the phoneme. Unfortunately we were using precomputed database of force alignment not containing this information. So we used ad-hoc solution: assigning 3 states (head/body/tail) to the phoneme according to its length. Although rough, it gave a little improvement to 2.3% EER (in GE SP JA MA given ASR setup). When using all five streams of open-loop phones, this approach reached 2.1% EER.

## 5 Further Work

Future work will cover two areas. First there will be search for faster implementation not requiring ASR. Experiment 2 suggests that this could be possible. Second I will try to examine and improve robustness to signal distortion by use of something else than open-loop phones. All testing will be obeying the NIST extended data task evaluation so it will be comparable with other systems. I will develop on SWB1 and finally will run the test on SWB2. Besides this, I will try to use smoothing which I think will improve the performance in case with 1 and 2 training conversations which is rather poor now. At last I would like to test this method on foreign language if I will find suitable corpus.

In search for a faster predictor stream, I will begin with phone recognizer with bigram (bi-phone) language model, having an open-loop phone recognizer as a baseline (4.3% EER). Moreover I will use conditioning to HMM states instead of phones here. After this will be done, there are several ways I can take. The most promising seems to be external predictor approach. By external predictor I denote other variable which serves as a predictor in conjunction with the phonemes. Let's use an example to demonstrate what I mean. Let's assume that we can detect speaker's "mood" for example by the rate of speech. It is admissible to assume that pronouciation model is different when person speaks slowly than when he/she speaks fast. So we can have binary variable *mood* having values 0 meaning slow and 1 to represent fast, coming out of the detector and we will condition phones on both mood and the phonemes (or HMM states). Another thing that can be done is a "soft" conditioning. It can be naturally combined with external "fuzzy" predictors. Soft conditioning calculates the probably during testing as follows:

$$P(EG = e | ASR = a) = P(EG = e | ASR = a, slow) * (1 - mood) + P(EG = e | ASR = a, fast) * mood$$

where  $mood \in [0, 1]$  is computed on per frame basis by the "mood recognizer".  $P(EG = e|ASR = a, slow)$  and  $P(EG = e|ASR = a, fast)$  are probabilities of the model gathered during its training. If the recognizer is absolutely sure about mood then it returns 0 or 1 and equation turns into conventional "hard" conditioning. If the recognizer is unsure it returns prior probability of  $mood$  being *fast* (given ASR) what is equivalent as if there was no conditioning on mood at all.

During training, probabilities are estimated as follows:

$$\begin{aligned}
 P(EG = e, fast|ASR = a) &= \frac{\sum_{INPUT(k)=(e,a)} mood(k)}{\#\{k|\exists f : INPUT(k) = (f, a)\}} \\
 P(EG = e, slow|ASR = a) &= \frac{\sum_{INPUT(k)=(e,a)} (1 - mood(k))}{\#\{k|\exists f : INPUT(k) = (f, a)\}} \\
 P(EG = e|ASR = a, fast) &= P(EG = e, fast|ASR = a)/P(fast|ASR = a) \\
 P(EG = e|ASR = a, slow) &= P(EG = e, slow|ASR = a)/(1 - P(fast|ASR = a)) \\
 P(fast|ASR = a) &= \frac{\sum_{\exists f:INPUT(k)=(f,a)} mood(k)}{\#\{k|\exists f : INPUT(k) = (f, a)\}}
 \end{aligned}$$

Where  $INPUT(k)$  means  $k$ -th input frame.

Variation of this "soft" method can be applied for more-valued variables as well. It would be nice if we could do it for phonemes, but unfortunately this would be computationally too intensive requiring to sum over the trellis. But we can always approximate by assuming the boundaries of the phonemes to be overlapping and probability of the phone increasing towards its center by some ad-hoc function.

Examining robustness is more connected with the predicted streams since predictors are usually sufficiently robust on its own, because they use some model (language, mood or whatever) which constrains audio data leaving enough room for noise and corruption of the signal. On the other hand predicted stream must react on the real sound more or less directly what makes the problem difficult, because increased precision means increased noise/distortion sensitivity. There are two approaches to this problem reflecting two different kinds of distortions.

(1) Frames with short noises or distortions, such as noise on the line, background noise or possibly other persons speaking on background may be detected and filtered-out before speaker detection, or in a "soft" approach, they can be given less weight than other frames depending on the rate of corruption.

(2) Constant channel distortion. For example echo, nonlinear distortion or stationary background noise. In this case some kind of quantization may be helpful. Some phones may be more robust than others. This has to be explored. For this purpose I want to simulate distortion of the signal by adding noise, non-linearly distorting the signal and using convolution to produce echo or bandstop filter. Finally I'd like to use some realistic combination of those to find sets of confusable phones as a basis for a quantization. This quantization will surely degrade the performance on the clean data but performance drop on distorted data should be so low then, that this method outperforms non-quantized one in distorted data case. In case where the detection of the constant signal distortion is not so hard (additive noise) I could quantize hierarchically thus allowing to be more precise in absence of noise and less confusing in case of its presence. Searching for predicted stream need not to stop on phones. I would like to examine articulatory features maybe resulting in construction of new features combining discriminative phones with discriminative articulatory features. For comparison I will also use raw and quantized MFCC values.

This work will result in at least two papers. One is currently being written by me, Jiří Navrátil and Doug Reynolds and it will be published on ICASSP 2003 Conference. It

summarizes what has been done on the Workshop. The other paper will cover the results of this extended research concentrating on predictor and predicted streams, their construction and performance analysis. Possibly, there could be yet another paper dedicated to analysis of noise/distortion sensitivity of phones and articulatory features.

## 6 Supervision

The supervisor of the project will be Jan Hajič of Charles University. Doug Reynolds of MIT Lincoln Lab and Fred Jelinek of JHU/CLSP have agreed to act as my advisors for this project.

## 7 Acknowledgements

I would like to thank my team I have been a member of, during the CLSP Summer Workshop, for their input and encouragement: team leader Doug Reynolds, Barbara Peskin, Joe Campbell, Walt Andrews, Jiří Navrátil, Andre Adami, Qin Jin, Radu Michaesku, Joy Abramson. Also I would like to thank to Fred Jelinek and CLSP staff for a creative atmosphere that they allowed to evolve, and for their support.

## 8 References

- [0] Doug Reynolds: “SuperSID Final Presentation”,  
<http://www.clsp.jhu.edu/ws2002/groups/supersid/supersid-final.pdf>
- [1] Doug Reynolds, Barbara Peskin, Joe Campbell, Walt Andrews, Jiří Navrátil, Andre Adami, Qin Jin, David Klusacek, Radu Michaesku, Joy Abramson: “SuperSID Team Final Report”, *Summer Workshop 2002, CLSP/JHU* – in preparation
- [2] Joe Campbell: “Speaker Recognition: A Tutorial”, *Proceedings IEEE*, 1997