AUTOMATIC SUBWORD UNIT REFINEMENT FOR SPONTANEOUS SPEECH RECOGNITION VIA PHONE SPLITTING

Jon P. Nedel, Rita Singh, and Richard M. Stern
Department of Electrical and Computer Engineering and School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{jnedel, rsingh, rms}@cs.cmu.edu, http://www.cs.cmu.edu/~robust/

ABSTRACT

Spontaneous speech is highly variable and rarely conforms to conventional assumptions and linguistically defined pronunciation rules. Specifically, there may be many different continuous speech realizations for each expertly defined phonetic unit in the dictionary. The phones may be realized in a clean and complete fashion as in read speech, or they may be realized in a sloppy and incomplete fashion as in highly spontaneous speech. For spontaneous speech, therefore, it may be beneficial to model incompletely realized variants of any phonetic unit as separate units. In this paper we test this hypothesis by introducing two possible modeling classes for the phones AA and IY in the standard English CMU recognition dictionary. We propose three different automatic methods of segregating the training data properly in order to identify and label the appropriate variants. Each of these methods results in improved recognition performance over the baseline, leading to the conclusion that finer modeling frameworks can be helpful to parameterize properly and recognize spontaneous speech.

1. INTRODUCTION

Acoustic modeling for speech recognition is accomplished by identifying a set of basic sound units and learning their relevant statistical parameters. In most systems human experts define a set of basic sound units that capture the characteristics of the training data and that generalize well to data outside the training corpus. Experts also design a dictionary of words and pronunciations using these basic units.

Spontaneous speech, however, is highly variable and rarely conforms to expertly defined pronunciation rules. We hypothesize that each phoneme has canonical or "target" feature values. When speech is carefully read, features transition from one target to the next and almost always reach the targets. When speaking casually, however, canonical feature values are not always attained. Particularly, when speech is very rapid we may only see transitions that head toward targets that are never reached.

Clearly, there are multiple possible continuous speech realizations for each phoneme in the dictionary. Other researchers have focused on learning alternate pronunciations and their corresponding probabilities from training data [1–3], and on adapting underlying HMM states and topologies to effectively capture these variations [4–5].

In this paper we attempt to improve the recognition performance for spontaneous speech by finer acoustic modeling in which variations in pronunciation of a particular phone are treated as separate phonetic units. For simplicity we confine our experiments to include the pronunciation variants of two common vowels AA and IY in the English language. The sounds represented by these phones are identical to those in the standard CMU recognition dictionary [6]. For each of these phones, we consider two possible phone realizations: a "pure" rendering as in more careful speech, and a "casual" rendering as in more spontaneous speech. The practical implementation of this conjecture poses a further problem — that of automatically partitioning all instances of the phones AA and IY in the training corpus into two separate acoustic modeling classes. In this paper, we propose three methods to accomplish this variant-based partitioning. Experimental results show that the resulting context-independent (CI) models consistently outperform standard models.

To simplify the description of the three partitioning techniques mentioned above, we first present our experimental framework in the following section. Sections 3, 4 and 5 then separately describe the partitioning techniques and their individual performance, which are collectively assessed in Section 6.

2. EXPERIMENTAL FRAMEWORK

2.1 The Multiple Register Speech Corpus

We used the NIST Multiple Register Speech Corpus (MULT_REG), a parallel corpus for comparison of spontaneous and read speech recorded at SRI. The database contains fifteen spontaneous conversations on assigned topics and re-read versions of the same conversations. For our experiments, we selected data solely from the spontaneous register. We used approximately 2 hours of spontaneous speech to train our acoustic models, and 0.5 hours of spontaneous speech to test our models.

2.2 Speech Recognizer and HMM Configuration

The CMU SPHINX-III recognition system was used for all experiments. The data were modeled using 3-state left-to-right HMMs with no state skipping. Due to the limited amount of data in our training set, we used semi-continuous

HMMs (codebook size 256). In order to focus on the effects of multiple models for each phone, we trained and tested context-independent (CI) models.

2.3 The Pronunciation Dictionary

We generated two different decoding dictionaries. The first was an expansion of the CMU pronunciation dictionary to include all possible pronunciations after the phone was partitioned. A word with N instances of a split phone had 2^N alternate pronunciations in the expanded dictionary. Alternatively, we trimmed this dictionary to contain only words and pronunciation variants seen in the training data. This enabled us to focus on the effects of phone splitting.

3. GAUSSIAN-BASED PHONE SPLITTING

In this technique we use a simple probabilistic model to segregate the members of a phonetic class into two separate classes. We assume that a single Gaussian distribution can be used to model all the observed feature vectors corresponding to the phone we want to split, and we compute the overall mean and variance of these features to specify the distribution. We then apply a small linear perturbation to the mean vector computed by adding a small value to its components. For each phone instance in the training set, we then calculate the sum of the log-likelihood of all the vectors in the segment with respect to each of the two Gaussians resulting from the perturbation. Each phone segment in the training set is then assigned to its new class depending on which summed log-likelihood score is higher. This process is illustrated in Figure 1.

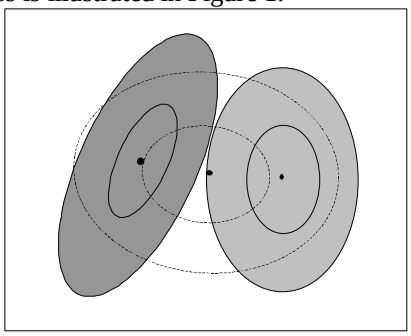

Figure 1. Illustration of Gaussian-based phone splitting. The dotted ovals represent the original Gaussian distribution of the features underlying the original phone class. The shaded areas represent the finer distributions corresponding to the resulting segregated classes.

Clearly the resulting distributions of the split phonetic classes must be sharper and more defined than the original distribution. Experiments performed to test the validity of this technique are described below.

To establish a baseline, we decoded our test set using the same CI models used to segregate the data. We tested using both the complete CMU dictionary, and a trimmed version of the CMU dictionary which contained only words present in the training corpus.

Following this, we first used the Gaussian-based phone splitting technique to separate only the phone AA into two separate classes, AA_0 and AA_1. New CI models were then trained for these phone classes, and were used to align the training data to the expanded version of the CMU dictionary that contained all possible pronunciations after the phone split, using the Viterbi algorithm. We then retrained the CI models for AA_0 and AA_1 using these aligned transcripts as a guide, and decoded the test set to evaluate the resulting models. Two different dictionaries were used for decoding; the first contained all possible pronunciations, and the second contained only words and pronunciation variants seen in the training set. We then repeated the process for only the phone IY. The models were trained, the data was Viterbi-aligned, the models were retrained and then evaluated as above.

Finally, we split AA and IY consecutively via the same process. The resulting models contained separate phone units AA_0, AA_1, IY_0, and IY_1. The WER results for all the experiments are shown in Table 1.

	Full dict	Trimmed dict
Baseline	51.1%	63.8%
Split AA	49.6%	62.0%
Split IY	49.3%	62.2%
Split AA then IY	50.2%	62.7%

Table 1. WER after phones AA and IY are segregated by Gaussian-based phone splitting. Full dict contains all possible pronunciations for decoding. Trimmed dict contains only words and pronunciations seen in the training set

Incorporating separate models for the phone classes derived with the Gaussian-based phone splitting technique resulted in a slight improvement over the standard CI baseline models. For the dictionary with all possible pronunciations, the best results occurred when IY was split and AA was left as a single class. However, with the trimmed dictionary which contained only words and pronunciations seen in the training set, the best results occurred when AA was split and IY was not. This could possibly be due to a large number of out-of-vocabulary words involving the variants of IY. In fact, we observe that the WERs in the case of the trimmed dictionary are consistently higher than with the full dictionary due to the absence of appropriate variant pronunciations in the trimmed dictionary.

4. HMM LIKELIHOOD-BASED PHONE SPLITTING

Although the simple Gaussian-based phone splitting results in better recognition performance, the method makes some general assumptions which are not quite valid. First, we

consider each speech feature vector independently of the sequence in which it occurs. The phone class is essentially modeled as a "bag of vectors" rather than a collection of sequential segments of speech. Second, the phones are segregated into separate classes without regard to the underlying acoustic models that we seek to improve by this technique. The splitting method that we propose in this section attempts to counter these shortcomings by appropriately use of the underlying phone HMMs and the likelihood scores associated with them.

The HMM itself provides a mechanism that can be used to segregate vector sequences into clusters. Each state of the HMM has an output probability distribution representing the vectors in the segments of speech it models. It is possible, therefore, to perturb the parameters of these distributions in such a way that the resulting likelihood scores can be used to segregate the observed sequences into two classes. Our second method of phone splitting adopts this procedure.

To test this procedure we trained fully-continuous CI models with one Gaussian per state with the assumption, as before, that the output distributions of these HMM states were broad enough to capture all of the data in a given phonetic class. We then perturbed the underlying mean vectors corresponding to the phone we wanted to split. In the case of the phone AA, we decided to perturb the mean vectors so that one of the models had means that were slightly closer to the means of the schwa (AX) model, while the other had means that were slightly further away from the means of the schwa model. The incorporation of this specific direction allowed us to further test our hypothesis and ensure that one resulting subclass was modeling more spontaneous renderings of the phone while the other was modeling more clean renderings of the phone.

We then Viterbi-aligned our training data using each of the CI CHMM models with perturbed mean vectors. The likelihood score corresponding to each phone segment was recorded. We then used the likelihood scores of the perturbed models to decide the new class membership of each segment. This process is illustrated in Figure 2.

We only used the fully-continuous HMM models to segregate phone instances into separate classes. Once the segregation was complete, we trained semi-continuous HMMs as before, Viterbi-aligning and retraining in each case before we evaluated the resulting models. We repeated the same experimental sequence as in the previous method, first splitting AA alone, then splitting IY alone, and finally splitting AA and IY. The WER results are shown in Table 1.

Again we see that the models with the split phonetic units perform better than baseline when evaluated on the spontaneous test set. We also observe that it is slightly better to split IY by itself than to split AA and IY together when using the large decoding dictionary.

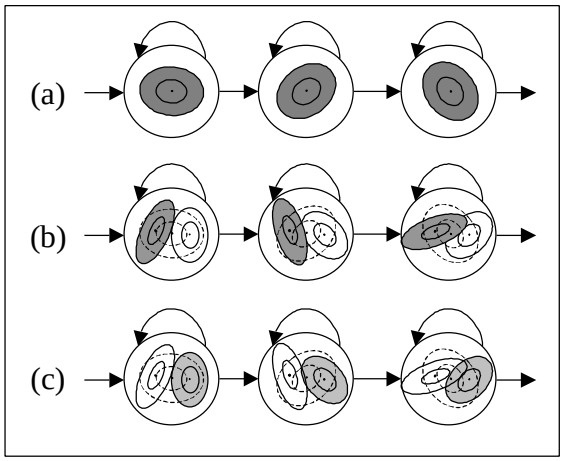

Figure 2. Illustration of HMM likelihood-based phone splitting. The means of the output densities of the CHMMs are perturbed to create two separate HMM models for the original phone class. (a) represents the original HMM. (b) represents the HMM with the means perturbed one direction. (c) represents the HMM with the means perturbed the other direction. Segregation depends on which underlying model, (b) or (c), yields the higher score.

	Full dict	Trimmed dict
Baseline	51.1%	63.8%
Split AA	50.1%	62.5%
Split IY	49.6%	61.9%
Split AA then IY	50.2%	62.7%

Table 2. WER after phones AA and IY are segregated based on the likelihood scores of 1Gau/state CHMMs with perturbed output density means. Full dict contains all possible pronunciations for decoding. Trimmed dict contains only words and pronunciations seen in the training set

In this case, however, it is also best to split IY alone when using the trimmed decoding dictionary.This contrasts with the observation in the previous section that splitting the phone AA results in better performance. This contrast may be attributable to the fact that Gaussians are more suited to modeling stationary sounds, whereas HMMs are more suited to modeling time-varying sounds. The phone AA is relatively more stationary than the phone IY, whose realization is different at the beginning and at the end of the phone. For example, the word INDIA consists of the phones IX N D IY AA. The phone IY here obviously captures the transition from the sound IY to the sound AA. Hence HMM-based splitting may be expected to give better results with the phone IY while simple Gaussian-based splitting gives better results with the phone AA.

In a follow-up experiment, we split all the vowels in the phone set via the HMM-based technique and achieved a WER of 50.4% using the full dictionary. This result is better than baseline, but we believe that the merits of the phone splitting technique cannot be seen due to the large number of alternate pronunciations in the dictionary.

5. DURATION-BASED PHONE SPLITTING

The final method that we propose in this paper for phone splitting is the simplest of the three methods. We studied the training set and found that the average duration of the schwa (AX) phone was 10 frames (where 1 frame = 10ms). We used this duration as a threshold and separated the phone into two classes based purely on the duration of the phone renderings. Each instance was assigned to one class if its duration was less than or equal to the duration of the average schwa and to the other class if its duration was greater than that of the average schwa. This method also attempted to segregate the phone into one class for more spontaneous renderings and another for more clean renderings. This was based on the assumption that the longer the duration of the vowel, the more likely it was to be a clean rendering of the phone. Duration-based phone splitting is illustrated in Figure 3.

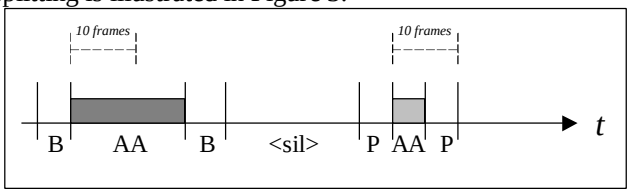

Figure 3. Illustration of duration-based phone splitting. Any instance of the phone with duration greater than 10 frames is placed in one class, and any instance with duration less than or equal to 10 frames is placed in the other class. 10 frames is the average duration of a schwa in the training set.

Again, the same sequence of experiments was performed as before, using semi-continuous CI models trained on phones segregated using the duration-based approach. The resulting models were evaluated. The results are reported in Table 3.

	Full dict	Trimmed dict
Baseline	51.1%	63.8%
Split AA	49.8%	62.1%
Split IY	49.6%	63.0%
Split AA then IY	49.9%	62.5%

Table 3. WER after phones AA and IY are segregated based on duration. Full dict contains all possible pronunciations for decoding. Trimmed dict contains only words and pronunciations seen in the training set.

The results again show an increase in performance over the baseline in each case. The best results with the full dictionary occurred when IY was split, and the best results with the trimmed dictionary occurred when AA was split.

6. DISCUSSION

All proposed methods yielded an improvement when the phone is divided into two classes, confirming our hypothesis that improved modeling of particularly different phone renderings should result in improved performance for spontaneous speech. However, no one method for phone splitting proved to be clearly superior to the other methods. Rather, we note that the splitting of AA and IY together using any given method did not outperform the splitting of AA alone or IY alone in most cases. Together with the observation that a given method of splitting worked better with one phone than the other, this suggests that phones must first be separated into "stationary" and "non-stationary" classes, and separate splitting methods must be applied to the phones in each of these classes.

An additional problem in increasing the number of pronunciation variants in the dictionary is that they increase the likelihood of confusing these pronunciation variants with the pronunciations of other words. Ideally, only the most frequently occurring pronunciation variants should be included in the dictionary. However, identification of these frequent variants would require large amounts of training data. Had more training data been available from the MULT_REG corpus, we would have attempted to do this. Alternatively, correlation between the pronunciations of various words could be applied to constrain the list of pronunciation variants. Words with similar pronunciations may be expected to have similar pronunciation variants for spontaneous speech.

7. ACKNOWLEDGEMENTS

The authors thank Dr. Bhiksha Raj for many fruitful discussions on the subject of this paper. This research was sponsored by the Department of the Navy, Naval Research Laboratory under Grant No. N00014-93-1-2005. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government.

8. REFERENCES

[1] W. Byrne, M. Finke, S. Khudanpur, J. McDonough, H. Nock, M. Riley, M. Saraclar, C. Wooters, and G. Zavaliagkos, "Pronunciation Modelling Using a Hand-Labelled Corpus for Conversational Speech Recognition", *1997 LVCSR Summer Workshop Technical Reports*.

[2] M. Finke and A. Waibel, "Speaker Mode Dependent Pronunciation Modeling in Large Vocabulary Conversational Speech Recognition", *EUROSPEECH 97*, p. 2379-2382.

[3] E. Fosler-Lussier, "Multi-Level Decision Trees for Static and Dynamic Pronunciation Models", *EUROSPEECH 99*, p. 459-462.

[4] E. Eide, "Automatic Modeling of Pronunciation Variations", *EUROSPEECH 99*, p. 451-454.

[5] M. Saraclar, H. Nock, S. Khudanpur, "Pronunciation Modeling by Sharing Gaussian Densities Across Phonetic Models", *EUROSPEECH 99*, p. 515-518.

[6] The CMU Pronouncing Dictionary, http://www.speech.cs.cmu.edu/cgi-bin/cmudict