STRUCTURED REDEFINITION OF SOUND UNITS BY MERGING AND SPLITTING FOR IMPROVED SPEECH RECOGNITION

Rita Singh[1], Bhiksha Raj[2], and Richard M. Stern[1]

1. Department of Electrical and Computer Engineering and School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213 USA
2. Compaq Computer Corporation
Cambridge, MA 02142 USA

ABSTRACT

The performance of speech recognition systems degrades when the basic sound units used are poorly defined or inconsistently used. Several attempts have been made to improve dictionaries automatically, either by redefining pronunciations of words in terms of existing sound units, or by redefining the sound units themselves completely. The problem with these approaches is that, while the former is limited by the sound units used, the latter discards all human information that has been incorporated into an expert-designed recognition dictionary. In this paper we propose a new merging-and-splitting algorithm that attempts to redefine the basic sound units used in the dictionary, while maintaining the expert knowledge built into a manually designed dictionary. Sound units from an existing dictionary are merged based on their inherent confusability, as measured by a Monte-Carlo based metric, and subsequently split to maximize the likelihood of the training data. Experiments with the Resource Management database indicate that this approach results in an improvement in recognition accuracy when context-independent models are used for recognition. When context-dependent models are used, the improvement observed is reduced.

1. INTRODUCTION

The performance of speech recognition systems is critically dependent on the basic sound units used: ill-defined or broadly-defined units have high inherent confusability and result in poor recognition accuracies. Traditionally, the basic sound units are phonetically motivated and are designed by human experts to be minimally confusable on an average. However, these phones may not be optimal for any specific task. The optimal units for a given task would depend on several factors such as the acoustic conditions of the task, the relative frequencies of various phones, their importance in distinguishing between words within the task, and the amount of training data available to train these units.

Researchers have previously tried to optimize sound units and dictionaries to specific tasks by either redefining the pronunciations of words in terms of the existing sound units [1], or by the automatic definition of the basic sound units themselves based on acoustic examples [2]. In the former case the basic manually defined units themselves are not modified. In the latter case they are completely discarded and replaced with automatically-estimated ones, thereby ignoring all the human knowledge that has been employed in designing them. Both approaches have drawbacks. In the former approach the performance of the system is limited by the definition of the basic units themselves. For example, if the sounds CH and JH have been mer ged into a single unit, no amount of pronunciation modelling can improve the system s ability to distinguish between the words GIN and CHIN . In the latter method, the inference of pronunciations of words that are not seen in the training data becomes extremely difficult. Further, the modelling of multiple pronunciations for words becomes extremely complicated.

In this paper, in contrast to previous work, we attempt to automatically redefine the basic sound units using a maximum likelihood merging and splitting algorithm, without discarding the expert knowledge that has been built into a manually-designed dictionary. This is accomplished by merging the existing sound units into a smaller set of units and then splitting the units in the smaller set to increase its size, without modifying the basic structure of the dictionary. Sounds units are merged based on their confusability as measured by a Monte-Carlo based symmetric cross entropy metric. The sound units chosen for splitting are the units whose split would result in the greatest increase in the likelihood of the training data. The pronunciations of words that have not been seen in the training data are easily obtained based on their correlations to the pronunciations of other words in the dictionary.

Experimental results show that the proposed merging and splitting algorithm results in significant improvements in the likelihood of the training data, and small but significant improvements in recognition accuracy over the baseline for the DARPA Resource Management task.

In Section 2 we describe the Monte-Carlo based distance metric that can be used to compute confusability between sound units as represented by their HMMs. In Section 3 we describe the splitting algorithm used to split sound units. In Section 4 we describe the overall merging and splitting algorithm. In Section 5 we describe experimental results. Finally in Section 6 we present our conclusions.

2. MONTE-CARLO BASED METRIC FOR COMPUTING DISTANCE BETWEEN PHONE HMMS

Consider two random variables X and Y, with $P_X(X)$ and $P_Y(Y)$. We define the distance $D(X, Y)$, between the two random variables as:

$$D(X, Y) = E_X\left[\log\left(P_x\left(\frac{X}{Y}\right)\right)\right] + E_Y\left[\log\left(P_Y\left(\frac{Y}{X}\right)\right)\right] \qquad (1)$$

where $E_x[\ ]$ refers to the expectation operator with respect to the distribution of the random variable X. Note that this metric is similar to the Kullback-Leibler metric, but is different from it in the sense that this is a true metric. This metric has previously been used in other problems such as segmentation of acoustic data [3] and is sometimes referred to as the KL2 metric.

This distance measure is easy to compute analytically when X and Y refer to random variables with simple exponential distributions. However, when X and Y have more complicated distributions such as Gaussian mixtures, or refer to samples of a non-stationary random process with complicated distributions such as those described by an HMM, no closed form expression exists for the KL2 metric. The most commonly used approximation in the case of Gaussian mixture distributions computes the distance between every Gaussian in the distribution of X and every Gaussian in the distribution of Y, and weights these terms by the product of the *a priori* probabilities of the Gaussians. In the case of HMMs this decomposes to finding the distance between the state distribution of every state of the HMM representing X and the state distribution of the corresponding state of the HMM representing Y. However, this approximation is far from satisfactory when finding the distance between HMMs of sound units, since there is no definite correspondence between the portions of two sounds represented by the corresponding states of their HMMs.

In this paper we solve this problem by replacing the expectation operator in Equation (1) by an averaging operation. In order to compute the distance between any two sound units, we first *generate* a large number of sequences from the HMMs of each of the two units. Sequences are generated from an HMM by exciting the initial state of the HMM and letting it transition through subsequent states until it enters the terminating state. Transitions from a state are decided randomly based on the transition probabilities of the state. At each state, an observation vector is generated from the state distribution associated with it. The number of sequences generated is proportional to the expected relative frequency of that sound unit in the test data. If this information is not available, we assume that all the sound units are equally likely and generate the same number of sequences for all sound units. Let S_i^X represent the i^{th} sequence generated from the HMM of the sound *X*. Similarly, let S_j^Y represent the j^{th} sequence generated from the HMM of the sound *Y*. The distance between the sound units *X* and *Y* is now defined as

$$D(X,Y) = \sum_i \log(P_X(S_i^X)) + \sum_i \log(P_Y(S_i^Y)) - \sum_i \log(P_X(S_i^Y)) - \sum_i \log(P_Y(S_i^X)) \quad (2)$$

where $\log(P_X(S_i^Y))$ is the log likelihood of the i^{th} sequence of *Y* measured on the HMM representing X.

We note that $D(X,Y)$ also represents the confusability between the sound units *X* and *Y*. The larger $D(X,Y)$ is, the lower the likelihood of sequences belonging to X on the HMM for Y (and vice-versa), and therefore the lower the likelihood that instances of *X* will be classified as *Y*.

3. HMM-BASED CLUSTERING OF SEQUENCES

Several clustering mechanisms have been proposed in the literature to cluster vectors belonging to stationary and identical independently-distributed processes. In this section we describe a clustering algorithm that is applicable to non-stationary sequences, such as phonemes, that can be modelled by an HMM. The objective of the clustering is to partition a set of sequences into a number of clusters such that the likelihood of the sequences on the HMMs representing them is maximized.

In the clustering algorithm we first train a single HMM with all the sequences being clustered. The mean of one of the states of this HMM is then perturbed since this is the smallest perturbation that can be given to the HMM. Two HMMs are thus created by adding and subtracting a small fraction of the standard deviation to the mean of this state. The likelihood of each of the sequences on these HMMs is evaluated and the sequence is said to belong the cluster whose HMM results in the higher likelihood. Once these preliminary clusters are formed, HMMs are retrained from each of the clusters and the likelihoods of all sequences are re-evaluated on these HMMs and cluster memberships are revised. This process is iterated until the likelihoods converge. At this stage, the cluster memberships of the sequences are assumed to be final. If multiple clusters are required, the cluster with the largest number of sequences is split again using the same procedure.

It is easy to see that the above sequence of steps is guaranteed to increase the total log likelihood of the data at every step. The clustering algorithm is a hill climbing algorithm, resulting in a locally optimal set of clusters.

As an alternative to this procedure, the expectation maximization (EM) algorithm [4] could be used to obtain better clusters. However, an EM-based solution would be much more computationally intensive, for relatively small gains in likelihood.

4. THE MAXIMUM LIKELIHOOD PHONE MERGING AND SPLITTING ALGORITHM

In this section we describe the complete merging and splitting algorithm for redefinition of the sound units and the recognition dictionary. The algorithm consists of two distinct steps. In the first step the closest and therefore most confusable sound units in the data are merged. In the second step the merged sound units are split to maximize the likelihood of the training data and the pronunciation dictionary is updated to use these new sound units. The following subsections describe these procedures in detail.

4.1. Merging phones

Context-independent models are first trained for all the phones in the existing dictionary. The most confusable pairs of phones are identified from this set based on the distance metric described in Section 2. We use confusability, rather than likelihood, as a criterion for the following reason: consider two words LAD : L AE D and LED : L EH D . If the two phones AE and EH were highly confusable, it would be preferable to merge the two into a single phone AE/EH and have identical pronunciations for both LAD and LED, and let the *a priori* probabilities of the words, as defined by the language model, determine the proper choice. It is important to note that for this reason, the language model must appropriately represent the given task. Once the closest phone pairs are identified, all instances of either phone in any given pair in the dictionary are replaced by a single symbol that represents both phones. New HMMs are trained with the modified dictionary using the reduced set of phones. If the number of phones in the dictionary is greater than desired, the confusability of phones in the dictionary is reevaluated for further merging.

4.2. Phone splitting

To split sound units, we first identify all segments of the training data that correspond to each of the phones. The set of segments representing each of the phones is divided into two clusters using the procedure described in Section 3, and the resulting increase in the likelihood of the data for that unit due to separation into clusters is noted. The phone for which the increase in likelihood is greatest is chosen for splitting.

The segments corresponding to this phone are then grouped into two clusters. Here we constrain the clustering such that if the phone occurs only once in the pronunciation of any word, then all instances of this unit occurring in instances of this word are grouped together in the same cluster. The two clusters represent two new sound units that have been generated by splitting the original phone, and are represented by two new symbols in the dictionary. The instances of the original phone in words where it occurs only once are replaced by the symbol representing the cluster into which all segments from that word have been incorporated. For words where the phone occurs more than once, the likelihood of all segments of data representing that phone in all instances of the word is measured on all pronunciation variants possible for the word using the two new symbols (a word where the phone occurs N times would have 2^N possible pronunciation using the new symbols). The pronunciation for which the likelihood is greatest is chosen as the pronunciation for that word.

HMMs are trained for the updated dictionary and phone set. Following this, the likelihood of all pronunciations that can be generated by replacing the original phone in the original dictionary by the two new symbols are generated, and the most likely pronunciation chosen for each word based on the newly trained HMMs. This step can be further iterated until recognition accuracy on a heldout set of data converges. It can be shown that this algorithm is guaranteed to increase the likelihood of the training data at each step.

If the desired number of sound units has not been achieved, the phone whose likelihood increases most due to splitting can be identified and the entire procedure repeated. It may not be necessary to merge a single pair of phones at each instance, and similarly to split a single pair of phones. This can be done in bigger steps, merging several pairs of phones simultaneously in the phone merging stage, and similarly splitting several phones simultaneously in the phone splitting stage.

The final set of sound units, and the corresponding dictionary give us the lexicon for training and recognition. The pronunciations for words that have not been seen in the training data can be obtained on the basis of the statistical similarity of their pronunciations in terms of the original phones to the pronunciations of other similar sounding words that have been seen in the training data.

5. EXPERIMENTAL RESULTS

The merging-and-splitting algorithm for sound unit redefinition was evaluated on the Resource Management (RM) database [5]. The CMU SPHINX-III speech recognition system was used for acoustic modeling. All experiments were conducted using continuous 5-state HMMs with one Gaussian modeling the distribution of each state.

The training corpus consisted of 2880 utterances, comprising 2.74 hours of acoustic signals. The training set covered a vocabulary of 987 words. The heldout set used to test the recognition performance at various stages consisted of 1600 utterances, comprising 1.58 hours of acoustic signals. The vocabulary of the heldout set was 991 words, four of which were not covered by the training set.

The baseline dictionary used was the CMUdict with 50 phones, which were then merged down to 44 in two steps of 3 phones each. The 44 phones were then split upwards back to 50 phones.

Figure 1 shows the KL2 distance between a subset of the phones in the CMU dictionary. The distance is color coded

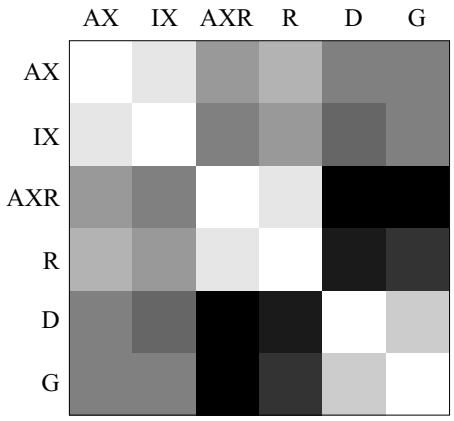

Fig. 1 Distance between sound units. The darker the color, the more distant the units

for visual clarity - the darker the color, the greater the distance. We note that the phones AX and IX are very close. The reason for this is apparent from an examination of the CMUdict, in which AX and IX represent very similar sounds. The phones AXR and R are also seen to be very close. Once again, the reason is that these are very similar sounds as they are used in the CMUdict, and the distinction between the two is sometimes unclear. The confusion between the two is also reflected in the fact that the distances between AX and AXR, and AX and R are very similar (in fact $D(\mathrm{AX}, R)$ is *less* than $D(\mathrm{AX}, \mathrm{AXR})$). The distance between AX and D or G is greater than the distance between AX and AXR or R, as expected. The phones D and G are observed to be very close. While this may appear anomalous, this is probably due to the effect of a lack of training data for the sound G, resulting in poorly trained models. Based on the observations in Figure 1, it is clear that AX and IX are good candidates for merging, as are AXR and R, and D and G. Using similar criteria, a total of six phone pairs were merged to reduce the size of the phone set to 44 phones.

We note here that some apparently dissimilar phones were also observed to be very close. For example, the KL2 distance between the phones AX and DX was relatively small. Further analysis showed that the phone AX occurred immediately after DX in 44% of all dictionary entries involving DX. Due to this correlation, the models for DX were corrupted by data from AX. We hypothesize that in such cases it may be advantageous to join all instances of DX AX into a single compound phone.

Figure 2 shows the relative increase in likelihood expected from the splitting of several phones from the reduced phone set. Sound units that have been formed by merging phones are represented by concatenating the symbols of the original phones with an underscore (*e.g.* AX_IX has been formed by merging AX and IX). Clearly, the splitting of some phones results in much greater increase in likelihoods than the splitting of others. Six phones were chosen, based on this metric, for splitting. The reconstituted dictionary, after splitting, had

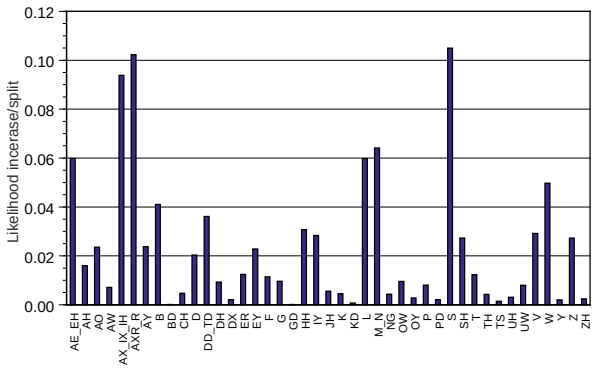

Fig. 2 Expected increase in likelihood due the splitting of several sound units

exactly as many entries as the original CMUdict.

Table 1 shows the recognition accuracies obtained with the original set of 50 phones from the CMUdict, the 44 phones that resulted after merging of phones, and the 50 phones that resulted from the splitting. The per-frame likelihoods obtained with these phone sets is also shown. All likelihoods and recognition accuracies shown were obtained with context-independent models. We note that the language weight used in these experiments was relatively small, in order to reduce any effects of the language model on the experiment.

No. of units	50 (manual)	44 (merged)	50 (Split)
WER (%)	15.4	17.6	13.8
Likelihood	-0.84	-0.92	-0.77

Table 1: Word error rates and per-frame likelihoods on the RM task, obtained with CI models for different sets of sound units

We observe that the likelihood obtained with the 50 reestimated phones is significantly higher than that obtained with the original set of phones in the CMUdict. The recognition accuracy obtained is also higher than that obtained with the phones in the CMUdict.

We trained context-dependent models with 2000 tied states and 1 Gaussian per state for both the CMUdict phones and the re-estimated sound units. Automatically generated linguistic questions [6] were used for building decision trees for state tying in both cases.

No. of units	50 (manual)	50 (Split)
WER(%)	9.2	9.0

Table 2: Word error rates obtained with CD models for the RM task with different sets of sound units

Table 2 shows the recognition accuracy obtained with context-dependent models for both the CMUdict phones and the reconstituted sound units. In the case of context-dependent models, it is observed that the difference between the CMU units and the reconstituted units is smaller than for the case of context-independent models. While this result is surprising, it can be explained. We hypothesize that the context information modeled by the context-dependent models reduces the inherent confusability between phones greatly. As a result, the improvement obtained due to the more sharply defined reconstituted sound units is not observed with context-dependent units.

6. SUMMARY AND CONCLUSIONS

In this paper we have described a new merging and splitting technique for redefining acoustic units in LVCSR systems in a structured manner. This method attempts to harness both the best features of sound-unit redefinition methods and the human knowledge built into standard lexica built by experts. The algorithm works by redefining the basic human-defined phone set in recognition systems by merging the closest units into a single unit, followed by splitting sound units to increase the size of the set of units. This method is seen to increase recognition accuracy when context-independent units are used for recognition. The improvements are greatly reduced, however, when context-dependent units are used for recognition.

The results obtained with the context-independent models, as well as the distance matrix represented in Figure 1 indicate that several phones in the CMUdict are indistinctly defined for the RM task, and rely greatly on contextual and linguistic information to improve the recognition. While the results with context-dependent models indicate that context information does indeed reduce the confusability between sound units, we expect that reducing the size of the phone set further, prior to expanding it would result in greater improvements in recognition accuracy.

ACKNOWLEDGMENTS

This research was sponsored by the Space and Naval Warfare Systems Center, San Diego, under Grant No. N66001-99-1-8905. The content of the information in this publication does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

REFERENCES

1. T. Sloboda and A. Waibel, Dictionary learning for spontaneous speech recognition, *Proc. Intl. Conference on Speech and Language Processing*, 1996, pp. 2328-233.

2. T. Holter and T. Svendsen, Combined optimization of baseforms and model parameters in speech recognition based on acoustic subword units, *Proc. IEEE Workshop on Automatic Speech Recognition*, 1997, pp.199-206.

3. M. Siegler, U. Jain, B. Raj, and R. M. Stern, Automatic Segmentation, Classification and Clustering of Broadcast News Audio, *Proc. DARPA Speech Recognition Workshop*, Feb. 1997, pp 97-99.

4. A. P. Dempster, N. Laird, and D. B. Rubin, Maximum Likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, 1977, Vol. B39, pp. 1-38.

5. P. Price, W. M. Fisher, J. Bernstein, and D. S. Pallet, The DARPA 1000-Word Resource Management database for continuous speech recognition, *Proc. IEEE Conference on Acoustics, Speech and Signal Processing*, 1988, pp. 651-654.

6. Singh, R., Raj, B., and Stern, R. Automatic clustering and generation of contextual questions for tied states in hidden Markov models , *Proc. IEEE Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, March 1999.