ABSTRACT

Obtaining a total contextual coverage and a smoothed training of phonetic models are the most relevant targets in acoustic modeling for continuous speech recognition systems. Traditionally, these objectives have been addressed using clustering techniques: either bottom-up or top-down ones. These two approaches yield complementary benefits. Because of their unrestricted optimization nature, bottom-up algorithms can reach a cluster configuration with a better homogeneity than top-down clustering can do. However, the phonetic guidance used by the latter gives a complete context generalization, allowing the provision of a model to contexts not found during the training step.

In this paper a hybrid algorithm that gets the best properties of both approaches is reported. This algorithm is applied to the hidden Markov models of demiphones, a new phonetic unit introduced by the authors recently. The junction of the demiphone and this hybrid algorithm provides a noticeable saving in the size of the set of phonetic units without degrading the performance.

I. INTRODUCTION

Recently [1], the authors have introduced the demiphone, a contextual phonetic unit that models a half of a phoneme. A left demiphone describes the beginning part of a phoneme and takes into account the coarticulatory effect produced by the previous sound. Accordingly, a right demiphone models the rest of the phoneme and depends on the next phoneme. For instance, the phoneme /s/ between the vowels /o/ and /a/ is modeled by the concatenation of two units o-s s+a, being o-s a left demiphone and s+a a right one.

The demiphone is closely related to the triphone and the diphone. In the previous example, we can observe that the demiphones o-s s+a represent the triphone o-s+a, being the main difference that the former model independently both the left and the right side coarticulation. On the other hand, a diphone is exactly reproduced by the concatenation of a right demiphone and a left demiphone. For instance, the diphone describing the sound from the center of the vowel /o/ to the middle of the fricative /s/ is represented by the pair of demiphones o+s o-s, that always appear together.

The demiphone has showed to be a very efficient sublexical unit for continuous speech recognition. Its dependence on only one context allows a simpler manipulation of units during the training and recognition procedures. Furthermore, the number of different demiphones is relatively reduced (less than one thousand for Spanish) if compared with the number of triphones (several thousands), providing an implicit smoothing for their hidden Markov models. And, more relevant, the demiphone outperforms the triphone, at least when moderate size databases are used for speaker and task independent training [2]. This result can be explained by the better statistical coverage of contexts that the demiphone provides (a reduced number of demiphones covers a great percentage of speech) and the actual weak dependence between right and left context for the most of coarticulation situations.

However, despite the good coverage of contexts that the demiphone provides, the problem of unseen units during the training or the lack of smoothness in the estimation of Hidden Markov Models (HMM) for some units can be also present. Clustering of models (or state of models) is used to overcome these two drawbacks. Decision-tree based (top-down) clustering gives a convenient solution since it provides smoothing and it assigns a model to units that are not present in the training speech material but appear in the target vocabulary. However, it has been shown in task-dependent training designs that agglomerative (bottom-up) clustering outperforms the top-down clustering. This better behavior of bottom-up clustering may be explained by the unrestricted way this algorithm can gather the models when looks for the best composition of clusters. Conversely, top-down algorithms are limited by the binary structure of the decision-trees. Unfortunately, agglomerative clustering does not provide generalization of contexts and it is not able to assign a model (out of an incontextual one) to an unseen unit. Thus, this situation gives rise to the following question: could top-down and bottom-up clustering algorithms be combined in such a way that the best of them can be obtained?
For each phoneme:
• Estimate a model for every left (right) demiphone whose number of appearances in the training material exceeds a threshold $T$.
• Clusters of left (right) demiphones are created:
  • Initially, a cluster contains one model.
  • Merge the pair of most similar clusters.
  • From the new cluster, consider the movement of every element to each of the other clusters:
    • Move the element if it improves the cluster configuration.
  • Iterate until a convergence criterion is met.

Figure 1. Algorithm for agglomerative clustering.

II. THE ALGORITHMS

In this section, details of the clustering algorithms we have experimented with are outlined. In all of the cases, we have used two-states hidden Markov models for the demiphones.

II.1 Agglomerative clustering

Figure 1 summarizes the bottom-up algorithm. It is a version of the algorithm firstly proposed by Kai-Fu Lee [3]. It starts estimating a hidden Markov model for every demiphone that appears at least $T$ times in the training material (in our experimentation $T$ was set to 25). For each phoneme, the demiphones without specific model are provided with a general incontextual model. We have two incontextual models for every phoneme: one for the right demiphones and other for the left ones. These generic models are estimated from the overall training speech and they are used to initialize the specific models too. The estimation of these specific models is accomplished by one iteration of the Baum-Welch training procedure. Thus, the segmentation supplied by the incontextual units is preserved and only the probabilities of symbol emission are recalculated. This strategy tries to maintain the temporal correspondence among the states of the different demiphones corresponding to the same phoneme.

Once these starting models are obtained, the agglomerative clustering itself begins and follows the steps described in Figure 1. Left and right demiphones are clustered separately. The criterion for algorithm finalization will be described bellow.

II.2 Decision-tree algorithm

Starting with the same set of models that the agglomerative approach does, the top-down clustering is conducted by a binary decision tree-based algorithm (as sketched in Figure 2). Independent trees are learnt for left and right demiphones. The questions used to split the nodes are concerned about the phonetic attributes of the context. Table I includes the attributes of the 25 phonemes we have considered for Spanish in our work.

For each phoneme:
• Estimate a model for every left (right) demiphone whose number of appearances in the training material exceeds a threshold $T$.
• Create a tree with one root node, including all the left (right) demiphones.
• Find the best question for every node.
• Split the nodes by using their best question.
• Iterate and stop when a prefixed criterion is met.

Figure 2. Binary-decision tree-based algorithm.

<table>
<thead>
<tr>
<th>phoneme</th>
<th>attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>vowel, central, open, unrounded.</td>
</tr>
<tr>
<td>B</td>
<td>consonant, fricative, bilabial, voiced.</td>
</tr>
<tr>
<td>D</td>
<td>consonant, fricative, dental, voiced.</td>
</tr>
<tr>
<td>e</td>
<td>vowel, front, open-mid, unrounded.</td>
</tr>
<tr>
<td>f</td>
<td>consonant, fricative, labiodental, unvoiced.</td>
</tr>
<tr>
<td>G</td>
<td>consonant, fricative, velar, voiced.</td>
</tr>
<tr>
<td>i</td>
<td>vowel, front, close, unrounded.</td>
</tr>
<tr>
<td>j</td>
<td>semivowel, palatal.</td>
</tr>
<tr>
<td>jj</td>
<td>consonant, fricative, palatal, voiced.</td>
</tr>
<tr>
<td>J</td>
<td>consonant, nasal, palatal, voiced.</td>
</tr>
<tr>
<td>k</td>
<td>consonant, stop, velar, unvoiced.</td>
</tr>
<tr>
<td>l</td>
<td>consonant, lateral, alveolar, voiced.</td>
</tr>
<tr>
<td>m</td>
<td>consonant, nasal, bilabial, voiced.</td>
</tr>
<tr>
<td>n</td>
<td>consonant, nasal, alveolar, voiced.</td>
</tr>
<tr>
<td>o</td>
<td>vowel, back, open-mid, rounded.</td>
</tr>
<tr>
<td>p</td>
<td>consonant, stop, bilabial, unvoiced.</td>
</tr>
<tr>
<td>r</td>
<td>consonant, trill, single, alveolar, voiced.</td>
</tr>
<tr>
<td>rr</td>
<td>consonant, trill, multiple, alveolar, voiced.</td>
</tr>
<tr>
<td>s</td>
<td>consonant, fricative, alveolar, voiced.</td>
</tr>
<tr>
<td>t</td>
<td>consonant, stop, dental, unvoiced.</td>
</tr>
<tr>
<td>tS</td>
<td>consonant, affricate, palatal, unvoiced.</td>
</tr>
<tr>
<td>T</td>
<td>consonant, fricative, dental, unvoiced.</td>
</tr>
<tr>
<td>u</td>
<td>vowel, back, close, rounded.</td>
</tr>
<tr>
<td>w</td>
<td>semivowel, velar.</td>
</tr>
<tr>
<td>x</td>
<td>consonant, fricative, velar, unvoiced.</td>
</tr>
</tbody>
</table>

Table I. Attributes for the Spanish phonemes.

The nomenclature used is X-SAMPA [4]. Because interword contexts are not considered in our recognition system, we have taken the beginning (and the end) of a word as a new context for a phoneme. We have assigned all the attributes to this additional context.

Similarly to the agglomerative algorithm, the finalization criterion will be discussed bellow.

II.3 Homogeneity function

Both bottom-up and top-down algorithms are conducted by the same measure of cluster homogeneity. Each cluster is represented by an average model obtained by the statistical mean of the models that populate the cluster. So, its homogeneity is assessed by the entropy of this average model. In fact, the homogeneity itself is the
average entropy of the probability distributions corresponding to the states of the hidden Markov model and to the different spectral vectors used to parameterize the signal.

In the implementation of the agglomerative clustering, two cluster A and B are merged if the pair (A, B) provokes the minimum increment in entropy:

$$\Delta H = (N_a + N_b) (H(A+B)) - N_a H(A) - N_b H(B)$$  \hspace{1cm} (1)

being $H(A)$, $H(B)$, $H(A+B)$ the entropy of cluster A, B and the junction A+B, respectively, and $N_i$ the number of available samples in the training corpus of the demiphones gathered in the cluster $X$.

Accordingly, the top-down algorithm splits the tree node A using the question $Q$ that leads to the maximum decrement of entropy:

$$\Delta H = N_{Q=Y} H(A_{Q=Y}) + N_{Q=N} H(A_{Q=N}) - N_a H(A)$$  \hspace{1cm} (2)

where $A_{Q=Y}$ denotes the part of A corresponding to the “yes” answer to question $Q$ and $N_{Q=N}$ is the number of tokens in $A_{Q=N}$. Similarly, $A_{Q=N}$ and $N_{Q=N}$ for the “no” answer. In both expressions (1) and (2), entropy is weighted by the population $N$. In such a way, the final clustering configurations respond to the actual similarity between contexts and attempt to associate the infrequent contexts [5].

II.4 Selection of units

Although the clustering procedures are implemented separately for left and right demiphones of every phoneme, the final definition of clusters is determined altogether for the overall set of demiphones. This operation is called “selection of units”. Two parameters steer this selection: the maximum number $M$ of units (models) to obtain and the minimum count $N$ of occurrences that is allowed to estimate a hidden Markov model (trainability criterion).

The agglomerative algorithm is performed until the left (or right) demiphones of a phoneme are merged into a unique cluster. Thus, we get designs with one, two, three (etc.) clusters for both the left and the right demiphones of every phoneme. The selection procedure starts with the configuration where each phoneme has one cluster for the left or right demiphones. The cluster constellation is explored looking for the cluster of demiphones that, being split back into two clusters, provides the maximum decrement of entropy. The algorithm continues going back along with the agglomerative process following the maximum decrease of weighted entropy. In every step the number of samples available to estimate the model for the new clusters is checked. The procedure ends when the maximum number $M$ of clusters is reached or when increasing the number of units leads to a violation of the trainability criterion.

The top-down clustering algorithm allows a straightforward implementation of the procedure above. The nodes are selected according to the decrement of entropy that has led to them.

II.5 The hybrid algorithm

The authors have implemented a hybrid algorithm to cluster the demiphones of a given phoneme that works in two steps. Firstly, for every phoneme a decision tree is learnt to classify the left (right) demiphones (selected by threshold) according the homogeneity function, providing (if necessary) an initial weak clustering. During the recognition phase, this tree will be used to group the unseen units along with the units available during the training. Thus, a model can be supplied to every unseen unit. The clustering procedure goes on by the agglomerative algorithm that gives the final cluster configuration. This hybrid algorithm was already suggested in [6].

III. EXPERIMENTAL FRAMEWORK

III.1 Data bases

We have experimented with two different data bases of Spanish material.

III.1.a SENGLAR data base

This data base has been recorded in our laboratory in a quiet room. The signal was sampled a 16 kHz, y quantified with 16 bits linearly. The training material stems from 171 speakers which uttered a total of 4349 phonetically balanced sentences, summing up more than 4 hours of speech and 175,000 phonemes. The test corpus is formed with oral queries into a geographic information data base: 1200 sentences (with an average number of words greater than 9) from 132 train-independent speakers. The vocabulary has 1070 words, 595 of them being names of places. The test set perplexity of the task assessed by bigrams is 12. Table II illustrates the coverage provided to this test by the train. It shows the number of demiphones in the task that either do not appear in the train corpus or are out of the set of demiphones (T=25) used to initialize the clustering algorithms (enclosed by parenthesis the actual number of unseen demiphones in the test speech are included).

III.1.b SpeechDat data base [7]

Other speech material used in our experimentation comes from the Spanish corpus of the SpeechDat project. The utterances were recorded through the public telephone fixed network, sampled at 8 kHz and quantified by the A-law at 8 bits per sample. As training material we have used 5,292 phonetically balanced sentences uttered by 2500 speakers, including more than 250,000 phones. It exceeds 17 hours of speech.

<table>
<thead>
<tr>
<th></th>
<th>SENGULAR</th>
<th>TIME</th>
<th>WORDS</th>
</tr>
</thead>
<tbody>
<tr>
<td>unseen</td>
<td>29 (5)</td>
<td>0</td>
<td>53</td>
</tr>
<tr>
<td>$T = 25$</td>
<td>95 (39)</td>
<td>0</td>
<td>160</td>
</tr>
</tbody>
</table>

Table II. Test coverage by the demiphones in the train.
We have designed two different tests with utterances from 500 training-independent speakers. The first one is composed by 380 time (TIME) sentences with an inventory of 60 words. 1360 isolated utterances corresponding to 1270 different phonetically rich words (WORDS) form the second test. As can be seen in Table II, the first test has not unseen demiphones. On the contrary, the second test includes a high percentage of not modeled demiphones (160 out of a total of 805).

III.2 System overview
The speech was parameterized with mel-cepstrum coefficients. CMS (cepstral mean substraction) was used. First and second order differential parameters plus the differential energy were employed.

The recognition system models the phonetic units by gaussian SCHMM with quantization to the 6 (2 for the energy) closest codewords. The size of the codebooks was 128 (32 for the differential energy).

The languages of SENGLAR and TIME tasks were modeled by an X-gram [8].

IV. EXPERIMENTAL RESULTS
Table III shows the word accuracy obtained in the SENGLAR experiment. The “no-gram” column refers to the recognition, allowing any sequence of words, of 464 task utterances whose vocabulary is limited to 310 words (310 perplexity). Results with 300 and 400 units are provided. The word accuracy corresponding to the recognition of the full task using an X-gram to model the language is under the “X-gram” title.

Table IV reports the results reached with the SpeechDat experiment using 350 and 450 models. In this case a bigger number of models was used accordingly with the greater size of the training corpus and the greater phonetic variability of the “words” task.

V. CONCLUSIONS
These experimental results confirm the hypothesis that the hybrid algorithm can share the best of the simple alternatives: the bottom-up and the top-down approaches. This algorithm, even with a more reduced number of models, outperforms the basic options.

When the test does not contain unseen demiphones (as in the “time” test), the three algorithms exhibit the same behavior.

Contrary to the general reported experience for speaker and task independent training, the bottom-up algorithm provides a better performance that the top-down one, mainly when the big set of units and the “words” test are considered. In the authors’ opinion, this result may be explained by the singular characteristics of the test.

The number of demiphones with counts in the training material exceeding the threshold 25 is 600 and 650 for SENGLAR and SpeechDat, respectively. Except for the “words” test (artificially rich in phonetic variability), little increase in performance can be obtained using a number of units exceeding the half of those figures. This saving in units is a benefit provided by the demiphone itself: a reduced set of units is able to cope with a great percentage of actual speech.

ACKNOWLEDGEMENTS
The authors want to thank Ruth Ontoso, David Conejero and Rosa Hernández for programming the algorithms and carrying out the preliminary experimental work.

REFERENCES