USING ACOUSTIC CONDITION CLUSTERING TO IMPROVE ACOUSTIC CHANGE DETECTION ON BROADCAST NEWS

Javier Ferreiros López, Daniel P. W. Ellis*

{jfl|dpwe}@icsi.berkeley.edu

International Computer Science Institute ICSI, Berkeley, CA, USA

*currently with: Grupo de Tecnología del Habla GTH-IEL-UPM, Madrid, SPAIN

ABSTRACT

We have developed a system that breaks input speech into segments using an acoustic similarity measure. The aim is to detect the time points where the acoustic characteristics change, usually due to speaker changes but also resulting from changes in the acoustic environment. We have also developed a system to cluster the segments generated by the first system into clusters composed of homogeneous acoustic conditions. In this paper, we present a technique to improve the robustness of the acoustic change detection by feeding back the results of the segment clustering, exploiting the extra information available in the distance between the two clusters to which the segments belong. The interaction between the acoustic change detection and clustering systems gives us a substantial improvement over results previously reported on the 1997 Hub-4 Broadcast News test set that we employed [1][2]: Feedback of clustering information improved the Equal Error Rate (EER) of our acoustic change detection (ACD) system from 26.5% to 18%.

1. INTRODUCTION

We have been working with the well-know Broadcast News database [4] of TV and radio recordings. This material contains multiple speakers, a wide range of speaking styles, and varied acoustic conditions such as background music, telephone channels etc. For a number of reasons, the task of acoustic change detection (ACD) is particularly important for this data. ACD consists of finding the time points where there is a speaker change (or more generally, an acoustic environment change). We can then cluster the segments defined by these time points into clusters containing homogeneous acoustic conditions, for instance, all the speech by a particular speaker in the same acoustic environment. This could ultimately be used to label the segments with the appropriate speaker/condition tags, giving us both the acoustic change points and the recognition of the acoustic condition in each segment.

One important motivation for ACD is to obtain segments of homogeneous speech. These segments are well suited to short-term adaptation in order to improve recognition accuracy. Another possibility is for the recognition language model to be "reset" at these break points to originate a new sentence [2]. With segments clustering we will of course obtain larger amounts of homogeneous speech material to be used for model adaptation. But most significantly for the current work, the clustering information may be used to improve the robustness of subsequent decisions on acoustic environment identity and can be fed back to the ACD system to perform better segmentation decisions. Completing the tasks with speech recognition and understanding, we could use these labels as indexes for information retrieval.

This paper is structured as follows. Section 2 presents the overall segmentation system architecture. In section 3 we discuss our ACD algorithm and present the way in which clustering information is fed-back within the segmentation. Section 4 describes our experiments and results, and section 5 forms our conclusions.

2. SYSTEM ARCHITECTURE

The segmentation system is composed of the following modules: 1) Feature extraction modules calculating both PLP-smoothed cepstra [5] used by the break-point hypothesis generator (Feature Extraction 1 in figure 1) and unsmoothed cepstra for ACD and clustering (Feature Extraction 2 in figure 1); 2) A break-point hypothesis generator based on a broad-class phonetic recognizer and subsequent filtering; 3) ACD based on the Bayes Information Criterion (BIC) acoustic similarity measurement; 4) A cluster generation and optimization algorithm for the acoustic segments. This architecture is shown in figure 1.

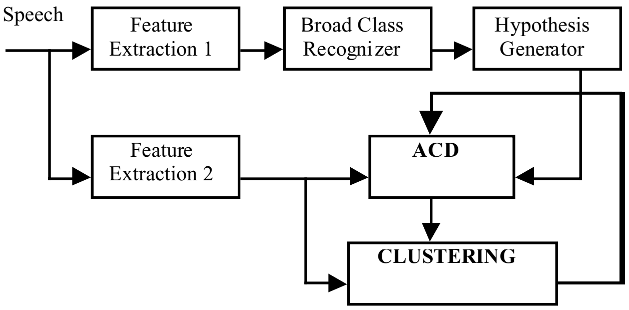

Figure 1: Block diagram of the system.

2.1 Speech Features Extraction Modules

The first blocks are speech feature extraction modules that extract a vector of features for each frame of the input waveform. We have depicted two different feature extraction modules because we found that the features best suited to our broad-class phonetic classifier were different from the features most successful in segment change detection. We use 12th order PLP-smoothed cepstra for the phonetic classifier, the features we have found best for our full speech recognition system. For ACD and clustering, we used unsmoothed cepstra. We experimented with using between 6 and 19 cepstral elements and found stable behavior between 11 and 15 parameters; table 1 shows our baseline ACD results as a function of the feature vector size, quoted as the equal-error rate (EER) over the test set described in section 4.

Number of features	%EER
6	32.0
7	30.5
8	29.5
9	29.5
10	27.5
11	26.5
12	26.5
13	26.5
14	26.0
15	26.5
19	27.0

Table 1: ACD performance evolution for different numbers of cepstral features and full covariance models.

2.2 Break-point hypothesis generator

Based on our Broadcast News speech recognizer [3], the hypothesis generator uses a feed-forward multi-layer perceptron with a single hidden layer of 1600 units. The input is 9 consecutive feature frames, and the outputs estimate the posterior probability of four broad acoustic-phonetic classes: vowels+nasals, fricatives, obstruents, and non-speech. The ACD algorithm considers placing breakpoints only at sequences of 3 or more frames labeled as non-speech. This approach means that in a test set of 686069 frames, only 5180 candidate break points (or 0.76% of the frames) need be examined.

3. THE ACOUSTIC CHANGE DETECTOR

3.1. Acoustic Similarity Test based on the Bayesian Information Criterion

For adjacent acoustic segments (delimited by candidate break points from the hypothesis generator), an actual break point is inserted by comparing the fit of a single multidimensional Gaussian model for the entire segment with separate models for each side of the break. We compare these alternatives using the Bayesian Information Criterion (BIC) [1], a likelihood measurement penalized by the complexity of the assumed model. Given a set of N vectors $X=\{x_I : i=0..N-1\}$ that we are trying to represent through a model M, the BIC score would be:

$$BIC(M) = \log[L(X,M)] - \frac{\lambda}{2} \cdot \#(M) \cdot \log(N)$$

where the penalty weight λ should theoretically be 1. $\#(M)$ measures the complexity of the model by its free parameter count, and $L(X,M)$ is the likelihood of data X under model M. The segmentation decision depends on a comparison of a hypothetical division of X into two subsequences, with separate models for each and hence more parameters, and the null hypothesis of using a single model for the entire set. Using Gaussian models, we have the following hypotheses:

$$\begin{array}{ll} H_0: & x_0 \ldots x_{N-1} \sim N(\mu, \Sigma) \\ H_1: & x_0 \ldots x_{N_1-1} \sim N(\mu_1, \Sigma_1) \\ & x_{N_1} \ldots x_{N-1} \sim N(\mu_2, \Sigma_2) \end{array}$$

Using the BIC, and keeping from the likelihood estimation only the part that is dependent on the covariance model [2] (since the feature means can be strongly affected by irrelevant changes in static channel characteristics) we end up with the following hypothesis test:

$$\log\left(\frac{|\Sigma|^{N}}{|\Sigma_1|^{N_1} \cdot |\Sigma_2|^{N_2}}\right) - \frac{\lambda}{2} \cdot \left(d + \frac{d \cdot (d+1)}{2}\right) \cdot \log(N) \overset{H_1}{\underset{H_0}{\gtrless}} 0$$

If this number is positive, we decide H_1 and break the whole segment into the two sub-segments. In this equation the complexity of H_1 is penalized via the factor $d + d \cdot (d+1)/2$ i.e. the number of free parameters in the second full covariance Gaussian model for d-dimensional feature vectors. We have found throughout our experiments that the value of λ had to be tuned in order to obtain the EER. We have found a strong dependency of its optimal value with the dimensionality of the input parameter vector (d) that is not compensated for by the model complexity factor $\#(M)$.

3.2. Acoustic Change Detection Procedure

Given this mechanism to decide whether or not to accept a possible break point, we face the problem of dynamically produce the acoustic change detection of a large amount of speech. Instead of hypothesizing a possible break point on each input frame, we consider only the nonspeech regions detected by the phonetic classifier, as described above. We found that requiring the nonspeech regions to be at least 3 frames in duration helped remove a number of spurious 'glitches'. The non-speech regions are excluded from our acoustic similarity calculations, since they presumably do not contain any speech, and speaker change is our main focus. We hypothesize the center of the non-speech region as the possible break point; thus, these initial break points hypothesis define the maximum possible partitioning of the speech material into small segments. Using these basic segments, we run the following ACD procedure [2]:

ACD Procedure

1) Add another segment into a buffer and write down a new possible break point

2) Calculate likelihood ratio via the BIC for each hypothesis in the buffer. Find the most likely break point (the one with the highest likelihood ratio)

3) Is the best likelihood ratio test positive?

   YES: Produce a break point. Delete segmented zone from buffer.

4) Go to 1)

3.3. Clustering to improve ACD

Using the same BIC criterion, we generate an initial clustering of the segments produced by the ACD using an incremental algorithm. This clustering is then further refined with fewer than 10 iterations of a re-clustering of the input acoustic segments to avoid order effects in the clustering algorithm. The basic iteration is:

Clustering Optimization Procedure

5) Pick one ACD segment

6) Remove this segment from its cluster and update cluster data

7) Find the ‘closest’ cluster (if any) to the segment

8) If there is a representative cluster, go to 6)

9) Generate a new cluster with only this segment. Go to 7)

10) Update this cluster with the segment information

11) If there are more segments to process, go to 1)

Clustering information is fed back into the ACD procedure via two approaches: a "hard" decision where a break is only allowed if the segments belong to different clusters, and a soft integration where the distance between each segment and the center of its cluster acts as a confidence measurement, weighting the contribution of the between-cluster distance in a linear combination with the BIC criterion in the ACD decisions. This last approach gave the best 18% EER performance for the test task. The implementation of the soft integration is guided by the following ideas: We were pursuing how to integrate another BIC measurement dependent on the clusters to which the two segments belong. If we think of likelihood values as if they were distances, we could define generically a distance between two acoustic segments as:

$$d(X,Y) = (N_X + N_Y)\cdot\log\left|\Sigma_{X\cup Y}\right| - N_X\cdot\log\left|\Sigma_X\right| - N_Y\cdot\log\left|\Sigma_Y\right|$$

Using this distance, the first idea is to use the function

$$\begin{aligned} G &= d(C_1, C_2) \\ &= (N_{C_1} + N_{C_2})\cdot\log\left|\Sigma_{C_1\cup C_2}\right| - N_{C_1}\cdot\log\left|\Sigma_{C_1}\right| - N_{C_2}\cdot\log\left|\Sigma_{C_2}\right| \end{aligned}$$

where

C1 is the cluster to which the segment S1 belongs

C2 is the cluster to which the segment S2 belongs

as extra information about the suitability of separating S1 from S2. Assuming that C1 and C2 are good representatives of S1 and S2 correspondingly, the "distance" from one to the other will be greater when we should separate S1 from S2 and ideally 0 (because C1=C2) when S1 and S2 come from the same acoustic conditions. We wish to integrate this class-derived information with the distance metric based only on the segments,

$$F = N\cdot\log\left|\Sigma\right| - N_1\cdot\log\left|\Sigma_1\right| - N_2\cdot\log\left|\Sigma_2\right|$$

In order to give the same relevance to the information conveyed by F and G, we decided as a first approach to equalize their dynamic range with the following formula:

$$F + \frac{\sigma_F}{\sigma_G}\cdot G - \frac{\lambda}{2}\cdot\left(d + \frac{d\cdot(d+1)}{2}\right)\cdot\log(N)$$

where σ_F and σ_G are fixed estimates of the standard deviations of *F* and *G* over all the segments. (In fact, we found first an approximate experimental value for this equalizing parameter and then realized that it was close to this ratio for the samples we had and then decided to substitute the experimental value by this more flexible one.) We stress that this is a fixed value, rather than one that is updated during the procedure. We then use the same criterion of separating the two segments if this value is positive. The value λ has to be tuned again to different values to obtain EER. To apply different importance to each source of information, we used the formula:

$$(1-\alpha)\cdot F + \alpha\cdot\frac{\sigma_F}{\sigma_G}\cdot G - \frac{\lambda}{2}\cdot\left(d + \frac{d\cdot(d+1)}{1}\right)\cdot\log(N)$$

where α can be varied from 0 (only the baseline *F* function takes control on the decisions) to 1 (only the new *G* function decides). Finally, we also tried a dynamic α with the expression:

$$\alpha = e^{-\frac{d(S_1,C_1)+d(S_2,C_2)}{factor}}$$

Here, α is 1 if d(S1,C2) and d(S2,C2) are both zero. In this case, we know that C1 and C2 are perfect representatives of the segments S1 and S2 and this is an indication that we can thoroughly rely on the robust information given by the function *G* of the distance between centers of clusters. On the other hand, α will be close to zero when these distances will be large and C1 and C2 would be in this case bad representatives of the acoustic segments. In this case we prefer not to rely so much on the clustering information, but give more relevance to the information in the old function *F*.

4. EXPERIMENTS AND RESULTS

For the evaluation of our systems we have used the same database than [1] & [2], that is, 3 hours of speech that were defined as the Hub 4 1997 evaluation data. The hand labeled transcriptions define 618 segments in this database (617 breakpoints) that are mostly speaker changes but also include changes to and from music, silence, excluded regions, etc. We will consider them all as true target break points. There are 119 different labels in the hand labeling of these segments, most of them the speaker proper names but also some others as: generic speaker labels (like "CNN_WVW_mAnnouncer1" or "female_nonnative3") and special labels (like "BEGIN", "Inter_segment_gap" or "Excluded_Region").

Following the evaluation directions in [2] for the ACD task, we have converted the break points into valid break regions that extend with the non-speech region around a certain breakpoint. This extension is produced using the labeling obtained by using the neural network trained at ICSI for recognition purposes. 6 frames of 16 msec. (about 100 msec. as in [2]) are added to both extremes of these regions as an allowance margin. With these criteria, we have defined the final target regions for our break points evaluation. Only one generated boundary can be matched to each target region. Different performances can be obtained tuning the value of λ. We can design a large variety of systems from the ones having high false acceptance with low false rejection to those performing low false acceptance with high false rejection. That is why when comparing results we have used the EER working point where the system has the same false acceptance and rejection rates. This point is also, by definition, the case in which the system generates exactly the same number of break points as the reference (617 in our case).

4.1 Results

We first made a study of the dependence of the ACD accuracy with the variation in number of features extracted. Table 1 presents the results with full-covariance Gaussian models for the specified number of plain cepstral coefficients. Table 2 shows intermediate results for several of the approaches described above. Our best result of 18% EER was achieved by the full dynamic soft integration of clustering information.

System variant	EER%
Baseline 12th order cepstral ACD	26.5
Hard feedback of cluster info	21.0
Soft feedback, equal weight	20.5
Soft feedback, optimal static α	20.0
Soft feedback, dynamic α	18.0

Table 2: ACD performance for various system configurations.

5. CONCLUSIONS

The BIC criterion performs well as an acoustic similarity criterion and allows ACD with reasonable results in accuracy and CPU time if it is accompanied by a hypothesis generator. We have improved the baseline accuracy by feeding back information from an optimized clustering of the produced segments. A dynamic soft integration has been presented that was found to be the best strategy for integrating the clustering information into the ACD module.

ACKNOWLEDGMENTS

This work has been done during a stay of the first author at ICSI with a grant from CICYT (Comisión Interministerial de Ciencia y Tecnología) Spain. We feel also indebted to the ideas and support received from Prof. Nelson Morgan and the rest of the Realization Group at ICSI.

REFERENCES

1. Scott Shaobing Chen, P.S. Gopalakrishnan, "Speaker, environment and channel change detection and clustering via the bayesian information criterion", 1998 DARPA Broadcast News Transcription & Understanding Workshop.

2. Daben Liu, Francis Kubala, "Fast speaker change detection for broadcast news transcription and indexing", Eurospeech 99.

3. Gary Cook, James Christie, Dan Ellis, Eric Fosler-Lussier, Yoshi Gotoh, Brian Kingsbury, Nelson Morgan, Steve Renals, Tony Robinson & Gethin Williams, "An overview of the SPRACH system for the transcription of Broadcast News", 1999 DARPA Broadcast News Transcription & Understanding Workshop.

4. NIST, 1997 Broadcast News Speech Corpus, CSR-V, Hub 4, Produced by the Linguistic Data Consortium, Catalog No. LDC98S71 1997. http://morph.ldc.upenn.edu/Catalog/LDC98S71.html

5. Hynek Hermansky, Perceptual linear predictive (PLP) analysis of speech, J. Acoust. Soc. Am. 87 (4), April 1990, pp. 1738-1752