EFFICIENT SPEAKER DETECTION VIA TARGET DEPENDENT DATA REDUCTION

Upendra Chaudhari, Olivier Verscheure, Juan Huerta, Xiang Li, Ganesh Ramaswamy, and Lisa Amini

IBM T.J. Watson Research Center
Rt. 134, Yorktown Heights, NY 10598
uvc@us.ibm.com

ABSTRACT

Systems designed to extract time-critical information from large volumes of unstructured data must include the ability, both from an architectural and algorithmic point of view, to filter out unimportant data that might otherwise overwhelm the available resources. This paper presents an approach for data filtering to reduce computation in the context of a distributed speech processing architecture designed to detect or identify speakers. Here, filtering means either dropping and ignoring data or passing it on for further processing. The goal of the paper is to show that when the filter is designed to select and pass on a subset of the input data that best preserves the ability to recognize a specific desired speaker, or group of speakers, a large percentage of the data can be ignored while being able to preserve most of the accuracy.

1. INTRODUCTION

Contemporary speech analysis systems operate over a range of operating points, each with a characteristic resource usage, accuracy, and throughput. This paper addresses the task of filtering out data to increase the throughput (processing capacity) while trying to minimize the impact on accuracy and maintaining or reducing the resource usage for a speech based detection task, in particular, speaker recognition. Filtering in this context means selectively dropping data and preventing it from undergoing further analysis. The purpose is to improve the capability to scale to analyzing higher volumes of data with a limited set of resources. The problem formulation is based on the fact that speaker recognition can be realized as computationally isolated algorithmic components arranged in an analysis pipeline, wherein a sequence of speech frames, each processed individually, constitute the data stream flowing through the pipeline. For example, when the input is compressed audio data, the various components might include audio waveform decompression, speech feature extraction, intermediate feature processing, and speaker detection. These algorithmic components could in fact be used for multiple tasks in this distributed architecture. Also, any component can act as a data filter for downstream components, eliminating their computation time for the filtered frame. As an example, if the waveform decompression component detects a problem with a portion of the input bit stream, it could choose to ignore that data. Success with respect to the goal of reducing computation and preserving accuracy depends on the quality of the filters.

2. PROBLEM FORMULATION AND CONTEXT

For the experiments in this paper, it is assumed that phonetic labels are associated with each data frame, but the technique could apply generally to any set of labels. Two components in the distributed architecture are studied, the filter, whose input is the set of labeled frames, and the identification/detection component (see figure 1). Data reduction is measured in terms of the percentage of frames dropped by the filter and accuracy is measured by identification and verification performance. A methodology is developed to determine target dependent filters based on the special properties of the recognition task at handwith the goal of reducing computation and preserving accuracy. Thus, the speaker, or group of speakers, to be detected will determine the filter, which is characterized by a subset of the labels and which operates by looking at the label associated with an input frame and passes it on for further processing only if that label is within its subset. The determination of this subset is a central component of this work. Note that phone based speaker models and phone sequence modeling [1] [2] have been used for speaker recognition. However, here our focus is on filtering (i.e. passing or dropping) data in a speaker or target specific manner to increase processing capacity in a distributed framework while trying to maintain accuracy. We do not address the issue of labeling the data, but rather the proper construction of filters (choosing the pass subset) given a particular set of labels. Also, it is important to point out that speaker models are built within a text-independent system and no knowledge of phone labels are used in the process. The filters are designed given these models. We remark that in designing the overall system, the complexity of the labeling should be taken into account and that in this distributed context, it is reasonable to assume that scenarios exist where such labels are generated early on, perhaps for a different task. Experimental support is provided as evidence of the efficacy of the proposed methodology. The rest of the paper is organized as

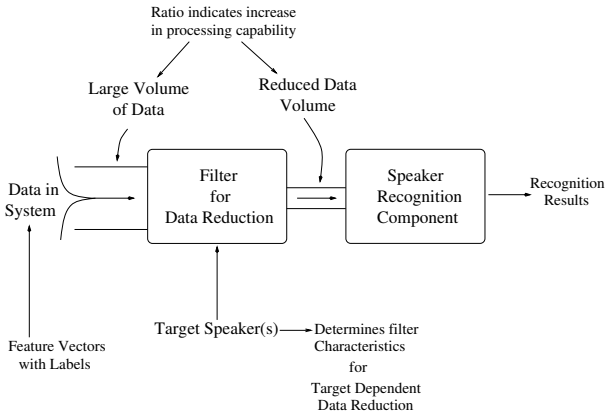

Fig. 1. *Diagram of the system subcomponent for speaker recognition.*

follows. In section 3 the recognition model is presented describing how speaker models are built and evaluated. Section 4 then describes the method for designing the target dependent filters and describes how the discriminants are affected. In section 5, experimental evaluation of the methods are presented. And finally section 6 presents discussion and conclusions.

3. RECOGNITION MODEL

The experiments are carried out on a state of the art speaker recognition system, where modeling is based on transformation enhanced models [3] in the GMM-UBM (Gaussian Mixture Model - Universal Background Model) framework [4] [5]. Data is represented as a sequence of vectors $\{\mathbf{x_i}\}$, where $i = 1, 2, 3, \ldots$, each element corresponding to an observed data frame. The UBM Model M_{UBM} is parameterized by

$$\{\mathbf{m}_i^{\text{UBM}}, \mathbf{\Sigma}_i^{\text{UBM}}, p_i^{\text{UBM}}\}_{i\,=\,1,\ldots,N^{\text{UBM}}} \text{ and } \mathbf{T}^{\text{UBM}},$$

consisting of the estimates of the mean, diagonal covariance, and mixture weight parameters for each of the N^{UBM} Gaussian components in the transformed space specified by an MLLT transformation, $\mathbf{T}^{\text{UBM}}$, which is chosen to give the optimal space for restriction to diagonal covariance models [3]. That is

$$\mathbf{m}_i^{\text{UBM}} = \mathbf{T}^{\text{UBM}}\mathbf{m}_i^{\text{UBM},o}$$

and

$$\mathbf{\Sigma}_i^{\text{UBM}} = diag(\mathbf{T}^{\text{UBM}}\mathbf{\Sigma}_i^{\text{UBM},o}\mathbf{T}^{\text{UBM},\top}),$$

where the *o* in the superscript indicates parameters derived from the original untransformed data through EM iterations. After EM, $\mathbf{T}^{\text{UBM}}$ is estimated based on the resultant parameters and is subsequently applied to them to construct the final model. This UBM model represents the background population and is trained with data from a large number of speakers so as to create a model without idiosyncratic characteristics. Based on this reference, each speaker M_j is parameterized by

$$\{\mathbf{m}_i^j, \mathbf{\Sigma}_i^j, p_i^j\}_{i\,=\,1,\ldots,N^{\text{UBM}}}.$$

The speaker dependent MLLT, $\mathbf{T}^j$, is identical to $\mathbf{T}^{\text{UBM}}$, whereas more generally it could be different. These parameters are derived via MAP adaptation from the UBM parameters in the transformed space [4] [5], based on speaker specific training data. Note that the number of Gaussian components is the same as that for the UBM. Thus the observed speaker training data $\{\mathbf{x_i}\}$ is transformed into the new space $\{\mathbf{T}^{\text{UBM}}\mathbf{x_i}\}$ before the MAP adaptation.

3.1. Discriminants

To evaluate a speaker model with respect to test data we use a likelihood ratio based discriminant function that takes into account the added feature transformation. Given a set of vectors $\mathbf{X} = \{\mathbf{x}_t\},\ t = 1 \ldots N_{\text{test}}$, in R^n, the frame based discriminant function for any individual target model M^j is

$$\begin{aligned} d(\mathbf{x}_t|M^j) &= \log p(\mathbf{T}^{\text{UBM}}\mathbf{x}_t|\mathbf{m}_{i^*}^j, \mathbf{\Sigma}_{i^*}^j, p_{i^*}^j) \\ &- \max_i \left[\log p(\mathbf{T}^{\text{UBM}}\mathbf{x}_t|\mathbf{m}_i^{\text{UBM}}, \mathbf{\Sigma}_i^{\text{UBM}}, p_i^{\text{UBM}})\right] \quad (1) \end{aligned}$$

where the index i runs through the mixture components in the model M^{UBM}, i^* is the maximizing index, and $p(\cdot)$ is a multi-variate Gaussian density. Extending to the entire test data, gives

$$d(\mathbf{X}|M^j) = \frac{1}{N_{\text{test}}} \sum_{t=1}^{N_{\text{test}}} d(\mathbf{x}_t|M^j). \quad (2)$$

When used for verification, the result is compared to a threshold. For identification, the function is computed for all speakers j to find the maximizing speaker score. We motivate the use of a filter as a computationally significant mechanism to control resource usage by noting that the above computation is required for each frame analyzed. There is an additional final score sorting cost for identification, but for practical purposes, the number of frames will vastly outnumber the speakers, maintaining the significance of frame reduction.

4. FILTERING DATA

We present an approach to the task of data reduction (thereby increasing throughput) while trying to maintain a high level of accuracy based on careful application of data filtering, where the filters that are designed are especially suited to the detection task. Such a reduction of data by filtering allows, for example, more audio data to be processed in a fixed amount of time on fixed resources, i.e. an increase in processing capacity.

The architecture for speaker recognition that is studied is a subset of a larger system that includes a feature extraction component, a feature labeling component, a filter, and a speaker detection component. Recall again that the operation of labeling is not addressed here. The recognition task itself encompasses two sub-tasks, that of identification and verification (detection). The general approach developed applies to both cases, however greater benefits are realized for the case of detection owing to the specificity of the task.

The sequence of test data frames is denoted by $\{\mathbf{x}_t\},\ t = 1 \ldots N_{\text{test}}$. Each element of the sequence has a label, such that labels and frames are in one to one correspondence. The labeling is assumed to produce for each frame, and element $\mathbf{l}$ from an alphabet of labels $\mathcal{L}$. Thus,

$$\mathbf{X}' = \{(\mathbf{x}_t, \mathbf{l}_t)\},\ t = 1 \ldots N_{\text{test}},$$

where the prime is used to indicate the set of label augmented feature vectors.

The speaker models are represented by the set $\mathcal{M} = \{M_j\}$. Let $\{\mathbf{x}_{dev}^j\}$ be development data for model M_j and $\{\mathbf{x}_{dev}\}$ be their union over j. Define

$$F_{\text{SI}} = \{\mathbf{l}_k\}, k = 1 \ldots N_{\text{SI}}^L$$

to be the set of labels defining the filter independent of the speaker being detected (The SI indicates speaker independent). N_{SI}^L is the total number of labels in the filter. Let $\{\mathbf{x}_{dev,\mathbf{l}}\}$ be the subset of the development data labeled $\mathbf{l}$. Then

$$\mathbf{l}_1 = \text{argmax}_{\mathbf{l}_i \in \mathcal{L}} \textit{perf}(\{\mathbf{x}_{dev,\mathbf{l}_i}\}),$$

where *perf* is the performance measure of interest. For the speaker independent filter, this measure is the aggregate identification rate computed over all target models using the development data. The particular experiment used for optimization is a closed set identification task among all target speakers. Continuing,

$$\mathbf{l}_n = \text{argmax}_{\mathbf{l}_i \in \mathcal{L} - \{\mathbf{l}_1 \ldots \mathbf{l}_{n-1}\}} \textit{perf}(\{\mathbf{x}_{dev,\mathbf{l}_i}\}).$$

Similarly,

$$F_j = \{\mathbf{l}_k\}, k = 1 \ldots N_j^L$$

which is the set of labels defining the filter for speaker j, defined as above with $\{\mathbf{x}_{dev,\mathbf{l}_i}^j\}$ replacing $\{\mathbf{x}_{dev,\mathbf{l}_i}\}$ (i.e. use data only from speaker j and label $\mathbf{l}_i$) and the individual identification rate of speaker j replacing the aggregate rate for the performance measure.

The discriminant with speaker independent filtering becomes

$$d(\mathbf{X}'|M^j) = \frac{1}{\sum_{i=1}^{N_{\text{test}}} I(\mathbf{l}_i \in F_{\text{SI}})} \sum_{t=1}^{N_{\text{test}}} I(\mathbf{l}_t \in F_{\text{SI}}) d(\mathbf{x}_t|M^j), \quad (3)$$

and with speaker dependent filtering,

$$d(\mathbf{X}'|M^j) = \frac{1}{\sum_{i=1}^{N_{\text{test}}} I(\mathbf{l}_i \in F_j)} \sum_{t=1}^{N_{\text{test}}} I(\mathbf{l}_t \in F_j) d(\mathbf{x}_t|M^j), \quad (4)$$

where $I(\cdot)$ is an indicator function of the validity of its argument, taken to mean that the vector's label is passed by the filter.

Thus target models are associated with an optimal set of labels with respect to recognition performance. In cases where the identity of the speaker sought is known, the data in the system can be filtered to pass only the optimal labels, F_j, for that speaker. On the other hand, if there is a set of speakers of interest, say the enrolled population, then an aggregate best list of labels, F_{SI}, can be chosen. Silence removal, a common practice, would be a degenerate form of this type of filtering.

The nature of the labels, e.g. the characteristics of the alphabet, will determine the granularity with which the data can be filtered to achieve various operating points on the performance vs. complexity curve. In the experiments, phonetic labels are studied.

5. EXPERIMENTS

5.1. Setup

The data consisted of the audio portion of the HUB4 Broadcast News Database. A subset of 64 speakers were selected as the target speakers. The waveforms were mono 16kHz PCM. The analysis configuration was 19 dimensional MFCC + 1st derivative (38 dim vector) with feature warping[6]. A rate of 100 frames per second, with 50% overlap was used and the MFCC were computed over a 20 millisecond window. For each speaker, 2 minutes of data were set aside and used for training the final speaker models. The UBM, trained on independent broadcast news data, contained 256 Gaussian components. The speaker models, being MAP adapted from the UBM, also had 256 components. The remaining data was partitioned and used for system evaluation. No labeled data was used in training the speaker models. Thus, we are not trying to build phone dependent speaker models, but rather, given well trained speaker models, our goal is to filter (or drop) test data in a speaker dependent way. There were 683 testing cells, ranging from 306 to 2034 frames (3.06 to 20.34 seconds). These were determined by the segments of contiguous speech available for each speaker in the data set.

5.2. Label Ranking

The data was labeled with an HMM based ASR system that generated alignments to available transcripts. As such, the labels are relatively high in quality.

A set of 41 phonetic units were used:
S TS UW T N K Y Z AO AY SH W NG EY B
CH OY AX JH D G UH F V ER AA IH M DH
L AH P OW AW HH AE TH R IY EH ZH

	phone rank		
	1	2	3
name	N	IH	T
% data	6.21	5.74	5.69
% accuracy	72.62	68.67	64.42
top 2 % accuracy	82.43		
top 3 % accuracy	85.21		

Table 1. Identification performance for various filtering configurations.

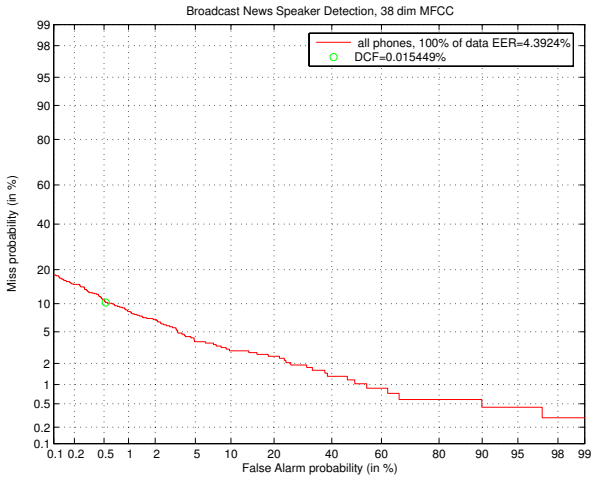

Fig. 2. Baseline verification performance using all data.

The first experiment details the per phone based recognition performance obtained based on the above labels using the 38 dimensional MFCC based features. Note again that the final speaker models did not use any label information. Table 1 summarizes the identification performance for filtering configurations based on phonesets determined by aggregate performance, i.e. based on the top labels in F_{SI}. Using all of the data, the overall identification performance was 92.53% correct. A further breakdown of the results shows, for the top 3 phones individually, the results were: 72.62% for "N", 68.67% for "IH", and 64.42% for "T" representing respectively, 6.21, 5.74, and 5.69 percent of the total data. The top phones were determined based on ranking of aggregate performance on all speakers. Scoring data from the top 2 phones combined, results in 82.43% accuracy on 11.96 percent of the data. The top 3 phones together give 85.21% on 17.65 percent of the data.

5.3. Detection

In the case of speaker detection, the results can be broken down with respect to speaker independent (aggregate best) phonesets and speaker dependent phonesets, which were determined as those for which the recognition rates were individually maximized. The baseline performance is given in figure 2. The equal error rate for this case where all of the data is used is 4.39%. DCF (Detection Cost Function: a weighted combination of false accept and false reject rates, defined for the NIST Speaker Recognition Evaluations [7]) values are also given in the plots.

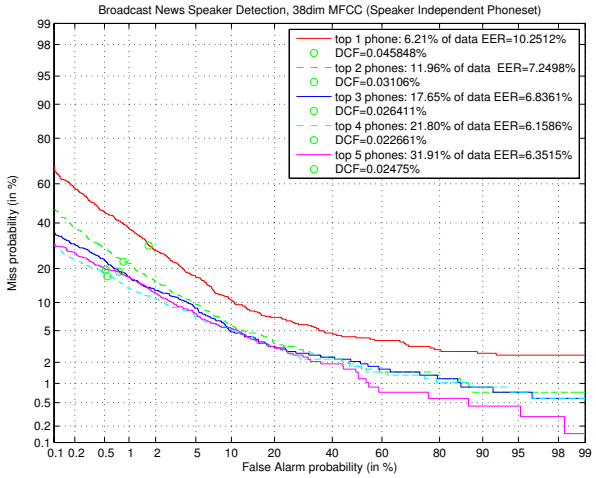

Fig. 3. Verification performance for aggregate best phonesets.

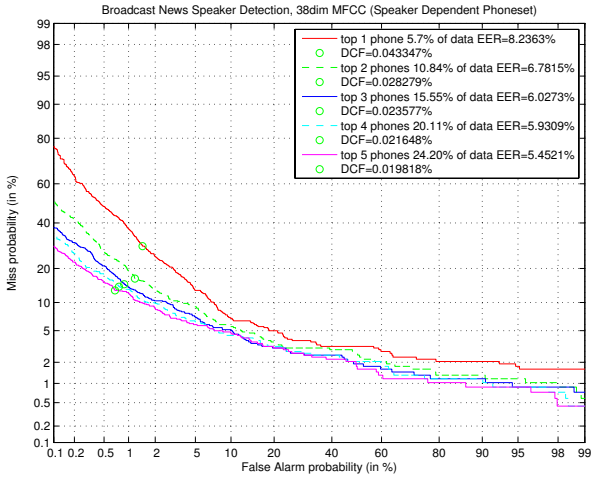

Fig. 4. Verification performance for speaker dependent phonesets.

5.3.1. Speaker Independent Filtering

Given this, the relevant question is "Can comparable performance be achieved as data is dropped and throughput is increased assuming constant resources?" The following results suggest that the question can be answered in the affirmative and provide an indication of the amount of throughput increase that can be achieved. Figure 3 shows the verification performance for 5 phonesets, ranging from the set with only the top performing (aggregate best) phone, to the set with the top 5 phones. Each of these sets represents a data filter, and the amount of data passing the filter naturally increases with the number of elements in the set. The performance improves as well. Filtering based on the top phone leaves 6.21% of the data with a corresponding equal error rate (EER) of 10.25%. Filtering based on the top 5 phones leaves 31.91% with an EER 0f 6.35%. A similar trend can be observed in the DCF values. We point out, that for the speaker independent case, the performance of the top 4 is better than that of the top 5. However, as will be seen in the next experiments, the top 5 set does indeed perform better than the top 4 set for the speaker dependent filters.

5.3.2. Speaker Dependent Filtering

In the case of a detection problem, of which verification is an example, the amount of data can be further reduced and performance improved by tailoring the filters to the entity or object being detected, in the present case a speaker. The results for this case are shown in figure 4. In this case, filtering based on the top phone, which depends on the speaker, passes only 5.7% of the data resulting in an EER of 8.24%, while the top 5 phones select 24.20% of the data for an EER of 5.45%. A similar trend is observed for the DCF. By having speaker dependent filters (in general, filters tailored to the entity being detected), less data is passed through the filters and better performance is achieved, as compared to the speaker independent case. For the present configuration, a throughput increase of 400% can be achieved with a 1% increase in EER, as compared to the all data case.

6. CONCLUSION

This paper presented a technique to reduce data flow, and thereby increase processing capacity, while preserving a high level of accuracy in a distributed speech processing environment. The task considered was speaker recognition and the data flow in the system was reduced by filtering out (dropping) data frames based on a target speaker specific subset of labels. The tradeoffs between the loss in accuracy and data reduction were investigated with experimental results verifying that data filters can be designed to preserve accuracy and pass only a fraction of the data by optimizing target (specific) performance measures.

7. REFERENCES

[1] A. O. Hatch, B. Peskin, and A. Stolcke, "Improved phonetic speaker recognition using lattice decoding," in *ICASSP*, March 2005, Philadelphia.

[2] W.M. Campbell, J.P. Campbell, D.A. Reynolds, D.A. Jones, and T.R. Leek, "Phonetic Speaker Recognition with Support Vector Machines," in *Proc. Neural Information Processing Systems Conference*, pp. 1377-1384, 8-13 December 2003, Vancouver, British Columbia.

[3] U.V. Chaudhari, J. Navrátil, and S.H. Maes, "Transformation enhanced multi-grained modeling for text-independent speaker recognition," in *International Conference on Spoken Language Processing (ICSLP)*, October 2000, Beijing.

[4] D.A. Reynolds, T.F. Quatieri, and R.B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital Signal Processing*, vol. 10, pp. 19–41, 2000.

[5] G.N. Ramaswamy, J. Navrátil, U.V. Chaudhari, and R.D. Zilca, "The ibm system for the nist 2002 cellular speaker verification evaluation," in *ICASSP*, April 2003, Hong Kong.

[6] J. Pelecanos and S. Sridharan, "Feature warping for robust speaker verification," in *Speaker Odyssey*, June 2001, Crete.

[7] NIST, "The nist year 2002 speaker recognition evaluation plan," 2002.