AN EFFICIENT SPEECH RECOGNITION ARCHITECTUREFOR HAND-HELD DEVICES USING CHANNEL AND ENERGY NORMALIZATION AND MCDCN

Dr. Himanshu N. Patel, Assistant Professor

Department of Computer Science, Dr. Babasaheb Ambedkar Open University, Ahmedabad, Gujarat, India.

himanshu.patel@baou.edu.in

Abstract—**this paper describes an efficient, small-vocabulary, grammar oriented speech recognition system using HMM (Hidden Markov Models) for mobile applications. The paper presents the issues and techniques forimproving strength and efficiency of the front-end, reducing computation through silence recognition, applying the Bayesian information criterion to build minor and superior acoustic models, minimizing finite state grammars, use of hybrid maximum likelihood and discriminative models, and generating baseforms from new spoken word.**

***Keywords*—Embedded speech recognition, low-resource speech recognition, robust speech recognition, mobile speech recognition.**

I. INTRODUCTION

The explosion and extensive use of mobile devices in everyday life has brought a need for efficient and easy to use interfaces to mobile devices. Technological advances make these mobile devices smaller, cheaper, powerful, and energy efficient. Diminishing dimensions have made the traditional keyboard/stylus interface of limited use and applicability in mobile applications. A conversational speech interface, emerging as a powerful and feasible alternativein such applications.

The increasing necessity of the conversational interface demandsimportant advances in processing power andspeech and natural language technologies. Speech recognition is significant need for a lowresourcespeech recognition system that is vigorous, precise, and effective.

This paper describes techniques for reducing the error rate,memory requirement and computational requirements of a grammar based, small vocabulary speech recognitionsystem which is intended for deployment on a portable device. Small vocabulary means about500 distinctive words or phrases within afinite-state grammar. By portable device we means system thatcan be executed by a 50 DMIPS processor, with 1 MB or less of DRAM and can be powered by a battery with an satisfactory lifetime [3].

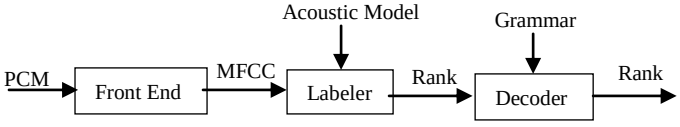
High accuracy in hostile acoustic environments is an important issue in portable device. Furthermore because they are intended for the mass market, they must be low cost. Similarly to appealto the consumer, the system must be simple and it the user must have the ability to easilypersonalize it to his/her needs.

The paper is organized as follows. We initiate with a broadoverview of the system organization in Section II. SectionIII defines various techniques to provide robustness tothe front-end. We then describe theacoustic models in Section IV. In specific we address trainingand model size in this section. Section V demonstrates techniquesfor grammar minimization. We conclude with a summary and discussion of the potentialfor low-resource speech recognition.

II. SYSTEM ORGANIZATION

The system is based onwell-known, phonetically based, HiddenMarkov Model approach. It is divided intothree primary components namely (1) front end, (2) labeler and (3) decoder as shown in Figure. 1. When processing speech, thecomputational workload is divided about equallyamong these components. The frontend may be activemore than the other modules, since it isused as well to separate speech from non-speech audio.

Fig.1 Flow chart of the system architecture



The front-end computes standard 13 dimensional MelFrequencyCepstral Coefficients from 16bit PCM sampled. The front-end also performs adaptive mean and energy normalization. The labeler computes first and second differences of the 13 dimensional cepstral vectors, and joins these with the original elements to produce a 39 dimensional feature vector.

The labeler then computes the log likelihood of each feature vector according to observation densities associated with the states of the system's Hidden Markov Model. This computation gives a ranked list of the top 100 HMM states. Possibilities are inferred based upon the rank of each HMM state by a table lookup [2]. The sequence of rank possibilities is then forwarded to the decoder. The decoder implements a synchronous Viterbi search over its vocabulary.

Words are represented as sequences of context-dependent phonemes, with each phoneme modelled as a 3 state HMM. The observation densities related with each HMM state are trained upon one phone of left context and one phone of right context only. Each observation density is modelled as a mixture of 39-dimensional diagonal Gaussians.

III. FRONT-END PROCESSING

Samples are partitioned into overlapping frames of 25 ms duration with a frame shift of 15 ms to reduce the overall computational load considerably without disturbing the recognition accurateness. Each frame of speech is windowed with a Hamming window and represented by a 13 dimensional MFCC vector. We observed by experience that noise sources have significant energy in the low frequencies and speech energy is mainly concentrated in frequencies above 200 Hz. Discarding the low frequencies therefore, improves the robustness of the system to noise.

A. Channel and Energy Normalization

To decrease the effects of channel and high changeability in the signal levels, the front-end performs adaptive mean exclusion and adaptive energy normalization, respectively. Cepstral mean removal betters channel changeability and consists of deducting from each incoming frame C_t, the presentapproximation of the mean value of the cepstra $\overline{C_t}$, for the present utterance

$$\tilde{C}_t = C_t - \overline{C}_t \qquad (1)$$

The cepstral mean is initialized with a value estimated off line on a representative collection of training data. It is then updated on a per-frame basis by interpolating the present mean estimate with each incoming frame

$$\overline{C}_t = \lambda \overline{C}_{t-1} + (1-\lambda) C_t \quad (2)$$

The interpolation weights are decided based upon the energy level of the frame.

Energy normalization consists of deducting from the zero[th]dimension of each incoming frame $C_t(0)$, an approximation of themaximum of the zero[th] cepstral coefficient, $\hat{C}_t(0)$

$$\tilde{C}_t(0) = C_t(0) - \hat{C}_t(0) \quad (3)$$

$\hat{C}_t(0)$ is larger of the maximum C_0value observed up to the t[th]frame of the n[th] utterance and the maximum C_0value observedduring the (n-1) [th] utterance.

B. Speech/Silence Detection

The front-end also separates speech from silence. By using simple Gaussian mixturemodels, front end labels each cepstral vector as speech/silence, and stores these vectors for later processing. It uses amixture of 4 Gaussians to model the distribution of silenceframes and speech frames. All Gaussianshave diagonal covariance. They are estimated on a balancedcollection of quiet and noisy data previously labeled withspeech and silence labels. When a sufficiently long sequenceof vectors labeled as speech has accumulated in the buffer,the front end determined that it is getting spoken language fordecoding, and forward the collected sequence of vectorsfor decoding. Sequences of vectors that are categorizedas silence are rejected without processing by the labeler andthe decoder, results in substantial computational savings.

C. Unwanted Signal Removal via Multichannel CDCN

Robustness in the existence of noiseand interfering signals, is animportant issue for speech recognition to work in a real world environment. To deal with a real time application constrainedto run with low computational resources, an inexpensiveandinaccurate form of adaptive filtering, a singletapdelay filter, is used to approximately align and scale the referencesignal with the noisy speech. The lineupand scaled referencesignal is then discarded from the noisy speech in the cepstral domainusing algorithm derived from CDCN [5] andcalled Multichannel Codebook Dependent CepstralNormalization. MCDCN is advantageousas:

- It allows to compensate for the loose modelingof the coupling system between the speech and the interferingsignal by taking advantage of our

knowledge of the clean speechdistribution in the cepstral domain,

- It can be adjusted to meet thedesired balance between performance and computational complexity,
- It performs on a per-frame basis, i.e., at a lowcomputationrate compared to waveform techniques,
- It does not include any iterative estimation scheme,therefore further enabling a real time use.

IV. Acoustic Models

The acoustic model contains context dependent sub-phone classes/allophones. The context for a given phone is composed of only one phone to its left and one phone to its right. A key concern is whether or not the phone context is permitted to extend across word boundaries. We have examined both approaches and settled on a system that uses within word context only since this made the search simpler and faster and had little or no effect in recognition accuracy for the tasks of interest. Except as noted in Section V all results in this paper are for such systems. The allophones are identified by growing a decision tree using the context-tagged training feature vectors and specifying the terminal nodes of the tree as the relevant instances of these classes [2].

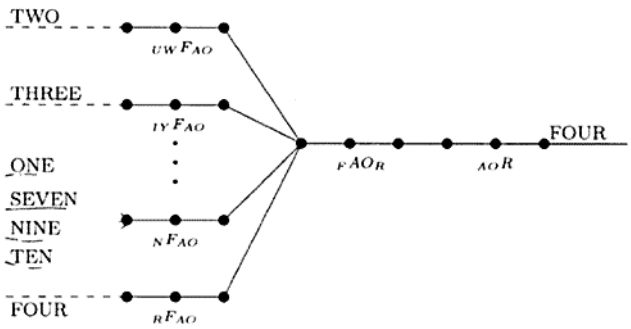
Each allophone is modeled by a single state HMM with a self-loop and an advancing transition. The training feature vectors are decanted down the decision tree and the vectors that collect at each leaf are modeled by a Gaussian Mixture Model, with diagonal covariance matrices to give a preliminary acoustic model. For the baseline, all the Gaussian Mixture Model had about equal number mixture components. Opening with these initial set of Gaussian Mixture Model several iterations of the standard Baum–Welch EM training procedure is run to obtain the final baseline model.

In our system, the output distributions on the state transitions are expressed in terms of the rank of the HMM state instead of in terms of the feature vector and the Gaussian mixture model modeling the leaf. The rank of a Hidden Markov Model state is gained by calculating the likelihood of the acoustic vector using the Gaussian Mixture Model at each state, and then ranking the states on the basis of their likelihoods.

V. Grammar Minimization

A finite state grammar can be represented as a weighted finite state automaton (FSA) on words, where all transition transmits a word and a language model probability. A word results in to a sequence of phones, each of which is modeled as a 3 state Hidden Markov Model. The phone models are context dependent. Here we only consider phones with cross word context hence; the first three states of a word model depend on the word that precedes it, as illustrated in Fig. 2

Fig.2 Sub graph of a digits grammar.



A word identifier is existing at the end of each sequence of states that models a word. Though they are not firmly necessary, these identifiers permit a fast mapping from the finest sequence of states to the recognized words when the search is finish.

Through determination and minimization, a weighted automaton with a lesser number of states may be created [1]. Though, the set of paths through the minimized graph is alike to the set of paths through the original one. Therefore, the sequence of states that best describes the acoustic observations is unaffected. Minimization essentially consists of sharing common states between paths that diverge or converge at a graph node.

The memory necessities for the storage of the graph and for the search are compact in the same amounts. As stated above, the recognition accuracy is not affected. Although the number of states to be visited throughout the search is decreased, the minimized graph is less regular than the original one. This has significant and astonishing computational penalties. The sharing of states decreases the amount of computation, since equal, parallel paths through the grammar are merged into a single path. But the merged paths must branch out again, as they eventually terminate in dissimilar words.

The unexpected result of this reduced symmetry is that a typical Reduced Instruction Set Computer (RISC) processor's core hardware, which generally includes a deeply pipelined Arithmetic and Logical Unit (ALU), cannot function at high efficiency therefore while minimization significantly reduces the total number of arithmetic operations during decoding; it results in modest increase in decoding.

VI. Summary

This paper described techniques to address recognition accuracy, robustness, system size, and computational

resource issues in phonetically based, small vocabulary speech recognition systems. We discuss methods for channel and energy normalization, speech silence detection and Multichannel Codebook Dependent CepstralNormalization.Speech recognition is already being investigated as a user interface for handheld computers [3], [4]. However technology stays to advance in shrinking size and improving performance. The technology tries to make those devices smaller, sleeker, lighter, faster and easier to use and hence ever more requiring accurate speech recognition.

REFERENCES

[1] M. Mohri, "Finite-state transducers in language and speech processing," Comput. Linguist., vol. 23, no. 3, 1997.
[2] L. R. Bahl et al., "Performance of the IBM large vocabulary continuous speech recognition system on the ARPA Wall Street Journal Task," in Proc. ICASSP 1995, vol. 1, pp. 41–44.
[3] L. Comerford, D. Frank, P. S. Gopalakrishnan, R. Gopinath, and J. Sedivy, "The IBM personal speech assistant," in Proc. ICASSP 2001, Salt Lake City, UT, May 2001.
[4] W. R. Hamburgen, D. A. Wallach, M. A. Viredaz, L. S. Brakmo, C. A. Waldspurger, J. F. Bartlett, T. Mann, and K. I. Farkas, "Itsy: Stretching the bounds of mobile computing," IEEE Computer, pp. 28–36, April 2001.
[5] A. Acero and R. M. Stern, "Environmental robustness in automaticspeech recognition," in Proc. ICASSP 1990.