Robust Romanian Language Automatic Speech Recognizer Based on Multistyle Training

DORU-PETRU MUNTEANU, CONSTANTIN-IULIAN VIZITIU
Communications and Electronic Systems Department
Military Technical Academy
G.Cosbuc 81-83, 050141, Bucharest
ROMANIA
munteanud@mta.ro, vic@mta.ro http://www.mta.ro

Abstract: - This paper presents solutions for increasing environmental robustness of a Romanian language continuous speech recognizer, previously developed. All state-of-the-art automatic speech recognizers (ASR) are data-driven and rely heavily on huge speech data for estimating the model parameters. Most of the available speech corpora used for this training phase contain clean speech recorded in low noise and reverberation free environments with high quality audio equipment. However, in real-world ASR are facing various acoustic conditions, speech signal being degraded by noise, reverberations, convolution distortions, etc. The acoustic mismatches between the training conditions and testing conditions are the main cause of ASR performance degradation. For instance, the word error rate may be an order of magnitude higher in an office environment than in a clean laboratory environment. There are a lot of methods and techniques aiming to keep the ASR performances at an acceptable in various acoustic conditions. In this paper we are presenting a special strategy called multistyle training for building a robust Romanian language ASR system. The method is based on training the recognizer with degraded speech signal obtained by adding to clean speech various levels artificial noise. Experimental results presented, prove that this scheme strongly increase the system robustness to additive noise. The system architecture based on context-dependent HMM phonemes is also described in detail.

Key-Words: - continuous speech recognition, environmental robustness, multistyle training, context dependent models, hidden Markov models

1 Introduction

Automatic speech recognition is still a subject for scientific research world-wide because it can offer cheap solutions in man-machine interaction. The recognition performances were increased every year in the last decades. A big challenge that both commercial and research ASRs have to address is the recognition robustness. There are various environmental factors that lead to speech signal degradation from the time it leaves the mouth until it reaches in digital format.

Most of the speech corpora contain clean speech recorded in low-noise reverberation-free conditions [7], [8]. Speech recognition systems performances trained with clean speech are known to degrade significantly in the real world applications [9] due to several factors that affect the speech signal such as additive noise (fans, air conditioning, door slams, keyboard or mouse clicks, etc.) or channel distortions (reverberations, microphone frequency response, A/D converter input filter, etc). There are two important strategies for increasing systems robustness: speech enhancement (e.g., spectral noise subtraction, echo cancellation) and acoustical model-based methods (e.g. adaptation techniques, parallel model combination, multistyle training).

The speech recognizer proposed in this paper is based on mainly two environmental methods:

• Cepstral mean normalization (CMN) – reduces convolutive channel distortion

• Multistyle training – adapts the models to additive stationary noise

Experimental results prove that system robustness is greatly improved for a wide range of the signal to noise ratio (SNR). Although we have modeled white Gaussian noise only, the method can be applied for any type of additive noise that could corrupt speech in various acoustic environments.

In this paper, the speech recognizer architecture is described first. The Romanian language ASR uses phoneme-based hidden Markov models (HMMs) with Gaussian distribution. Also a voice activity detector is used for real-time recognition in the testing phase. Then context-dependent (CD) modeling is used for training first order CD HMMs (triphones) in order to increase the ASR performances. This CD modeling is a very important aspect to be discussed as all recognition experiments were performed using both context independent (CI) and context dependent models.

2 Speech recognizer architecture

2.1 Continuous speech recognition

In the last years, considerable progress in large vocabulary continuous speech recognition (CSR) has been made [10]. Actual laboratory systems are capable of transcribing continuous speech from any speaker with average error rates under 5%. If speaker adaptation is allowed the error rate could be under 1% after few minutes of speech. Most of these speech recognizers are based on hidden Markov models (HMM) or hybrids HMM-Artificial Neural Networks (ANN). Unfortunately, for practical systems performances are worse because of environmental conditions and the way speakers speak.

Robust spontaneous speech recognition is still an elusive goal and actual systems are from far too complex for the performances they are deliver [4].

In previously published work [1], [2], [11], [19] it has been described the Romanian language continuous speech recognizer used for experiments presented in this paper. The ASR was build using a very well known toolkit [5].

The ASR system presented here [2] is based on statistical modeling of time-varying speech sequences with a well known and effective statistical modeling technique called Hidden Markov Model (HMM). The ASR building process is shown in Fig.1.

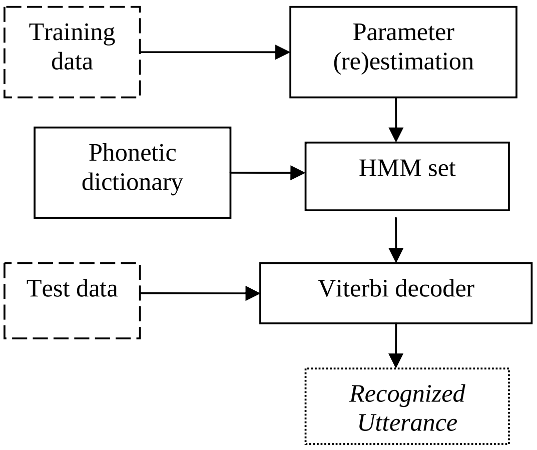

Fig. 1 Romanian language ASR

Each Romanian phoneme was modeled with a three-state HMM and a left-right topology, using multiple-mixture Gaussian continuous distribution (Fig. 2).

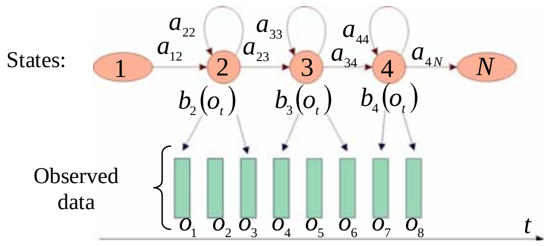

Fig. 2 Left-right topology HMM

The covariance matrices are diagonal in order to reduce the resources required for the output probability computation [19].

The ASR is based on 34 Romanian phonemes. First, context-independent HMM are trained and tested. Then, embedding expert knowledge about Romanian language in the form of phonetic questions, a set of context-dependent HMM are trained and refined.

There are few things that have to be mentioned regarding Romanian language phonetics. Although for most of the world languages, the orthographic representation is not phonetic, starting with 1880 Romanian language became mostly phonetic rather than etymologic as it was considered before. Its phonetic behavior is similar to the other Romance languages like Italian, Spanish, Portuguese, etc.

2.2 Voice activity detection

A very important characteristic of our ASR was the capability to perform in real time in order to test it in real conditions and to embed it in various applications. In order to perform such a task, the system needed a voice activity detector (VAD). It reduces the continuous speech recognition effort by separating the speech/silence parts from speech signal.

The ASR VAD uses a two level algorithm which first classifies each frame of data as either speech or silence and then applies a heuristic to determine the start and end of each utterance.

The detector classifies each frame as speech or silence based only on the log energy of the signal. A frame has a length of 20 ms (320 samples, at a 16 kHz sampling rate).

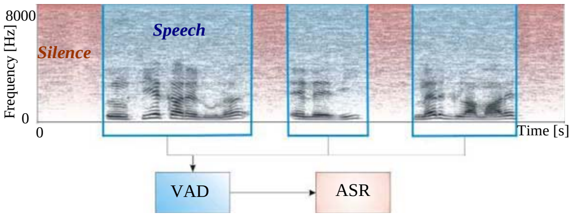

Fig. 3 Real-time voice activity detector applied to speech signal

When the energy value exceeds a threshold, the frame is marked as speech otherwise as silence. The threshold is made up of two components which are adjusted automatically within a calibrating procedure. The detector is adjusting adaptively its parameters from the current acoustic environment prior to speech recognition process itself.

2.3 Context dependent modeling

First option in building the ASR is to train context-independent models (monophones) from the training data. The transcription of the wave files is usually available at word level. The phonetic dictionary makes possible the transcription of the utterances at phoneme level. Such context independent monophones have the advantage of a good coverage of the data (each phoneme occurs few hundred times during few minutes of speech). Consequently CI models are trainable but they are not consistent.

The context dependent models ensure a better modeling accuracy, but the number of models increases heavily and there is much less training data for each model. In large vocabulary speech recognition, many contexts have only few occurrences in the training data that is insufficient for a robust parameter estimation of the corresponding models; there are also contexts that have no occurrence in the training set, the so-called “unseen contexts”. To handle these problems, first model – based tying was proposed [13]. State tying proved to be more efficient [14], [15] so that we have also adopted this strategy. The contextually equivalent sets of HMM states are determined in our approach applying the phonetic decision trees.

In context independent phoneme modeling each word results as a concatenation of the component phonemes; therefore a model is necessary for each phoneme. The Romanian language has 34 phonemes, requiring 34 different context-independent models.

In current speech, the words are not simple strings of independent phonemes: as effect of co articulation, the immediate neighbor – phonemes, for instance the preceding and the following one, affect each phoneme in the word. This immediate neighbor – phonemes are called respectively the left and the right context; a phoneme constitutes with the left and right context a triphone [4] For example in the triphone “Z – o + k”, (SAMPA- transcription [6] for the Romanian word ”joc”), the phoneme “o” has as left context “Z” and as right context “k”.

For each such a triphone a model must be trained: in Romanian that will give a number which equals 343 = 39305 acoustic models! The number of parameters to be estimated for such a huge system became prohibitive. First, for the recognition task presented in this paper we have modeled only word internal triphones, neglecting the co-articulation effect between words. Secondly, a state tying procedure conducted to a significant decrease in system parameters without loosing the modeling accuracy.

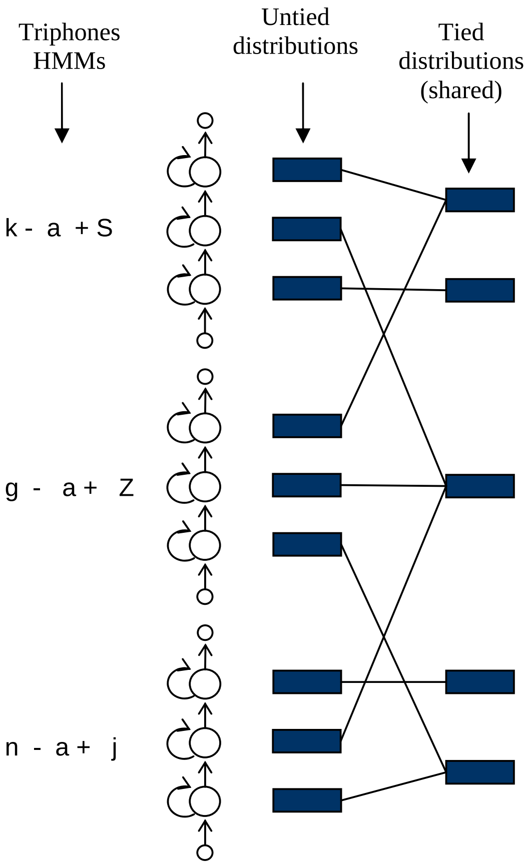

Fig.4. Triphone State tying for CI phoneme “a”

If triphones are used in place of single phonemes (monophones), the number of needed models increases and the problem of insufficient training data arise. One of the most efficient solutions for this problem consists in tying the acoustically similar states of the models built for triphones corresponding to each context.

For example, in figure 1, three models are representing two different contexts of the phoneme “a”, namely the triphones “k – a + S”, “g – a + z”, “n – a + j”. One may observe that the acoustically similar states are grouped into clusters. The models at the logical level are different, but at the physical level they partially share their distributions. In Figure 1, for untied states, there are 9 sets of distributions for the three triphones, while for the tied states, the same triphone share only 5 sets of distribution. The number of parameters has been reduced still keeping model specificity [20]. This tying process is in fact a compromise between the system complexity and system specificity.

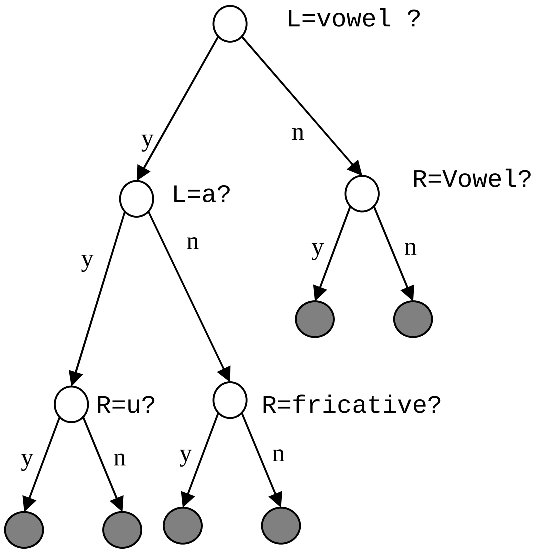

Fig. 5 Phonetic decision tree example

The choice of the states and the clustering in phonetic classes are achieved by mean of phonetic decision trees [14].

A phonetic decision tree built as a binary tree, is shown in figure 2 and has in the root node all the training frames to be tied, in other words all the contexts of a phoneme. To each node of the tree, beginning with the parent – nodes, is associated a phonetic question Q_i concerning the contexts of the phoneme. Possible questions are, for example: is the left context a vowel? (*L = vowel?*), is the right context the phoneme <u> (*R = u?*); the first question designates a class of phonemes while the second only a single phonetic element (monotonal).

Depending on the answer, yes or no, child nodes are created and the appropriate speech frames are placed in them. New questions are further applied for the child nodes, and the frames are divided again. The questions are selected in order to increase the log likelihood of the data after splitting. Splitting is stopped when increasing in log likelihood is less than an imposed threshold, leading to a leaf node. In such leaf nodes are concentrated all states having the same answer to the question made along the path from the root node and therefore states reaching the same leaf node can be tied as regarded acoustically similar. For each leaf node pair the occupancy must be calculated in order to merge insufficient occupied leaf nodes.

A decision tree is built for each state of each phoneme. The sequential top down construction of the decision trees is realized automatically, with an algorithm selecting the questions Qi from a large set Q of 130 phonetic questions based on phonetic knowledge on Romanian language.

2.4 Decoding

The main stages of the Romanian language ASR speech recognition process in the testing phase are presented in Fig. 6.

The unknown speech waveform is converted by the acoustic front-end into a sequence of acoustic vectors consisting in 12 mel-frequency cepstral coefficients (MFCC) accompanied by their first and second order derivatives.

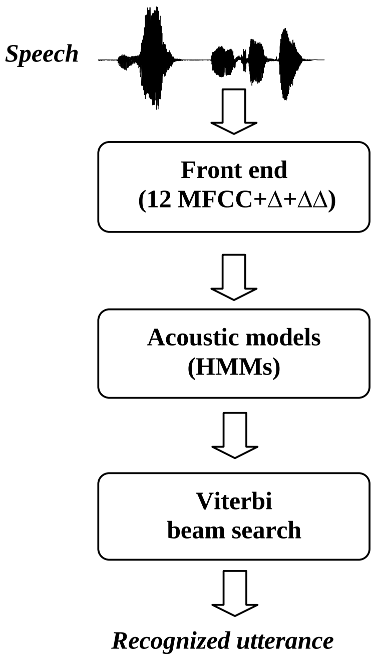

Fig. 6 Continuous Speech Recognition Stages

Phoneme-based context dependent (CD) HMMs with continuous Gaussian distribution were considered. A number of 34 Romanian language phoneme-based context independent (CI) models are trained in the first stage.

It is well known that CI models do not capture the inherent speech variability mainly due to the co articulation effect albeit they are trainable. The recognition system is furthermore refined and first order CD models (triphones) are trained.

3 The acoustical environment

In practice, real world speech differs from clean speech, being degraded by the acoustical environment, which could be defined as the transformations that affects speech from the time it leaves the mouth until it is in digital format. A recognition system is called robust if its accuracy does not significantly degrade under mismatched conditions. There are two classes of environmental factors that could corrupt speech:

- Additive noise: computer fans, air conditioning, door slams, other people speech.
- Channel distortion: reverberations, frequency response of the microphone or analog-to-digital converter (CAD).

In most cases, white noise is useful as a conceptual entity, but it seldom occurs in practice. Most of the noise captured by microphones is colored, since its spectrum is not flat (white). For example, pink noise is a particular type of colored noise that has a low-pass nature, as it has more energy at the low frequencies while rolling of at higher frequencies and it could be generated by a computer fan or an automobile engine.

Acoustical environment model is presented in Fig. 7, and the relation between corrupted speech $y[m]$ and clean speech $x[m]$ is given by:

$$y[m] = x[m] * h[m] + n[m] \qquad (1)$$

where $n[m]$ is the additive noise and $h[m]$is the impulse response of the environment.

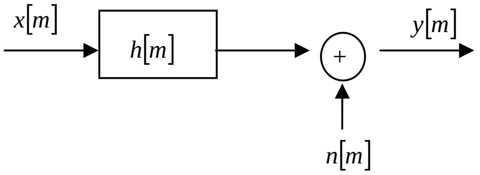

Fig. 7 Acoustical environment model

Regarding the convolution component $h[m]$, the most important factors that could affect the digital form of the speech are reverberation and microphone transfer function. Techniques such as Adaptive Echo Cancellation (AEC) have been successfully applied for reducing the reverberation.

In frequency domain, using discrete Fourier transform (DFT) in $2K$ points for digital signals, using the short term analysis [17]:

$$\begin{aligned} \left|Y(f_k)\right|^2 &= \left|X(f_k)\right|^2 \left|H(f_k)\right|^2 + \left|Z(f_k)\right|^2 \\ &+2\,\mathrm{Re}\left\{X(f_k)H(f_k)Z^*(f_k)\right\} \end{aligned} \qquad (2)$$

where Y, X, H and Z are DFT of y, x, h and z, respectively.

The last term of eq. (1) may be neglected as x and z signals may be considered statistically independent. In practical computing of the speech signal parameters, a bank of M filters is used, so that we may consider that:

$$\left|Y(f_i)\right|^2 \cong \left|X(f_i)\right|^2 \left|H(f_i)\right|^2 + \left|Z(f_i)\right|^2, i = \overline{1,M} \qquad (3)$$

where i is the bank filter index.

It is important to notice that we assumed the filter impulse response length $h[n]$ is smaller than the analysis window length $2N$. That is why ASR systems obtain poor results in rooms with long reverberation time, such as empty rooms with sound reflecting walls.

For MFCC analysis, on the logarithmic scale, it can be proved that:

$$\begin{aligned} \ln\left|Y(f_i)\right|^2 &\cong \ln\left|X(f_i)\right|^2 + \ln\left|H(f_i)\right|^2 \\ &+ \ln\Big(1 + \exp\Big(\ln\left|Z(f_i)\right|^2 - \ln\left|Y(f_i)\right|^2 - \ln\left|Y(f_i\right. \end{aligned} \qquad (4)$$

Most of the ASR systems are using cepstral coefficients, so that we can evaluate the effect of these distortions on the speech signal distorted by both noise and reverberations.

We define c as the discrete cosines transform (DCT) operator that is applied to the log-spectrum of a particular signal passed through the bank filter mentioned above. Signal vectors $M+1$ sized are given by:

$$\begin{aligned} c_x &= C\Big(\ln\left|X(f_0)\right|^2 \; \ln\left|X(f_1)\right|^2 \; \ldots \ln\left|X(f_M)\right|^2 \\ c_h &= C\Big(\ln\left|H(f_0)\right|^2 \; \ln\left|H(f_1)\right|^2 \; \ldots \ln\left|H(f_M)\right| \\ c_y &= C\Big(\ln\left|Y(f_0)\right|^2 \; \ln\left|Y(f_1)\right|^2 \; \ldots \ln\left|Y(f_M)\right|^2 \\ c_z &= C\Big(\ln\left|Z(f_0)\right|^2 \; \ln\left|Z(f_1)\right|^2 \; \ldots \ln\left|Z(f_M)\right| \end{aligned} \qquad (5)$$

It is defined the non-linear function $g(\cdot)$:

$$g(c_w) = C\ln\left(1+\exp\left(C^{-1}(c_w)\right)\right) \tag{6}$$

We can identify:

$$C\left(\ln\left(1+\exp\left(\ln\left|Z(f_i)\right|^2 - \ln\left|Y(f_i)\right|^2 - \ln\left|Y(f_i)\right|^2\right)\right)\right) = g(c_z - c_x - c_h) \tag{7}$$

From eq. (5), (6) and (7) results:

$$c_y = c_x + c_h + g(c_z - c_x - c_h) \tag{8}$$

This formula emphasize the fact that the distorted speech signal MFC coefficients (MFCC) are a combination of the MFC coefficients of the following signals: clean speech signal, environment impulse response function and noise signal [22].

It is well known the fact that MFCC of the speech signal, c_x have a Gaussian distribution.[18]

Obviously, one may observer from eq. (8) that MFCC of the speech signal distorted by the acoustic environment is not Gaussian anymore, because of non-linearity of the $g(\ )$ function.

Assuming the case of no-reverberation, we may simplify the eq. (8) :

$$c_y = c_x + \ln\left(1+\exp(c_z - c_x)\right) \tag{9}$$

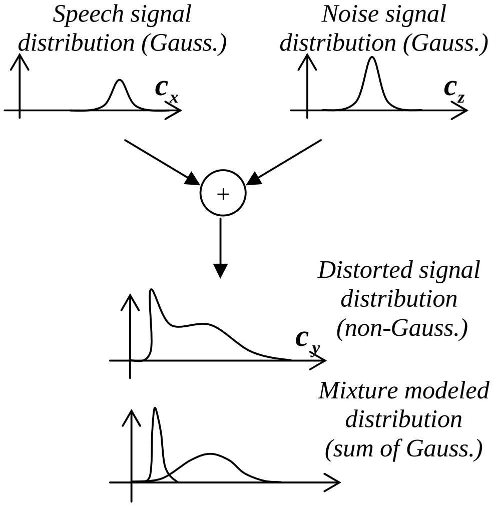
Speech signal distribution (Gauss.)
Noise signal distribution (Gauss.)
c_x
c_z
+
Distorted signal distribution (non-Gauss.)
c_y
Mixture modeled distribution (sum of Gauss.)

Fig.8 Modeling distorted signal with Gaussian mixtures.

Both c_x and c_z are assumed to have normal distribution. Monte Carlo simulation performed proved that for small variances of the speech MFCC comparing to noise MFCC, c_y has normal distribution. For state of the art ASRs this condition is met as they are working with mixtures of Gaussians. The variances of the mixtures are comparable with noise variances [17].

The emission distribution of the HMM of Romanian language ASR presented in this paper are modeled by Gaussian mixtures (Fig. 8).

The microphone is also very important for the speech acquisition. Head-mounted, close-talking microphones are recommended for most of the speech recognition system as they capture less of the surrounding noise (Fig. 9).

In order to eliminate the speech variability caused by different digital-analog converters (DAC), it could be included within the head-set and connected by USB. Another promising strategy for speech acquisition is to use array of microphones. The idea is to use more than one microphone, estimate the relative phase of the signal arriving to each of the element array and than to compute the angle of the arrival. After locating the speaker, all other perturbing signals arriving from other directions or distances are rejected. The major drawbacks of the multi-microphone systems are that they require additional computation to enhance speech and, on the other hand, they also need special hardware (multiple microphones input).

Fig. 9 Microphones used for speech recognition: array (a), USB close-talking (b), close-talking(c), desktop (d)

In order to reduce the serious mismatch between the training and test conditions that often causes dramatic degradation of the accuracy of the recognizers, three major categories of techniques are used:

- *Inherently robust parameters* for speech, such as Perceptual Linear Prediction (PLP)

- *speech enhancement* including AEC, noise spectral subtraction (NSS) [12], algorithms based on arrays of microphones
- *model based methods* for noise compensation

In this paper we are presenting experimental results for model based techniques. The major problem for the speech recognizer is the mismatch between the training data (usually, noise-free high quality speech) and test data (environmental conditions).

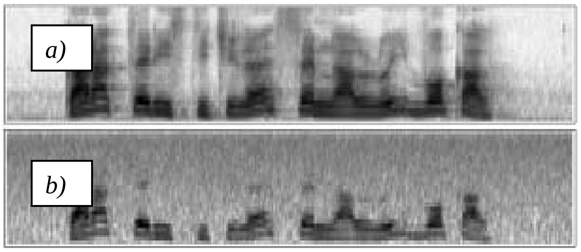

Fig. 10 MFCC vs. time for a) clean speech (SNR>40 dB) and b) corrupted speech (SNR=10 dB)

In order to simplify the problem we have referred especially to the additive noise. The simplest approach for this problem is to train the system with the same signal-to-noise ration (SNR) as in the test condition.

The training data may be easily processed by adding to the clean speech noise artificially generated with the same distribution as the noise from the test conditions. Consequently, noisy MFCCs are generated (Fig.10).

Experiments presented here prove that such a matched system performs quite well, much better than the system trained with clean speech, anyway. This simple strategy works only if the test conditions are known and stationary but fails in any other situations.

Classical adaptation techniques such as Maximum Posterior Probability Estimation (MAP) or Maximum Likelihood Linear Regression (MLLR) could be used to adapt a clean, speaker-independent recognizer to a particular speaker or to a particular environment. After few thousands of adaptation phrases, the recognition system is adapted to the new condition. Adaptation to new conditions approaches are time consuming and sometimes the applications don’t allow them.

In order to increase the environmental robustness of the Romanian language continuous speech recognizer we have adopted the so called multistyle training. Various SNR phrases are produced by adding artificial noise to the clean speech and then the system is trained with the whole collection.

4 Experimental results

4.1 Romanian language continuous speech recognizer

The speech recognizer has an architecture that is described by Fig.1. The acoustical front-end provides 12 mel-frequency cepstral coefficients (MFCC) for each frame of 25 ms, at 100 frames/s rate. Prior to signal parameterization input signal is preemphasized by a filter with the transfer function:

$$H(z) = 1 - 0.97\, z^{-1} \qquad (9)$$

Each frame is weighted by a Hamming window. Acoustic vectors are augmented by the first and second variation coefficients.

For acoustical modeling we have used phone-based HMMs with three states in a left-right topology. Continuous Gaussian output distribution with diagonal variance matrices has been adopted. CI models parameters for all 34 Romanian language phonemes were estimated.

The main steps of the training procedure for the context independent models are:

- HMM initialization - all models are identical
- Baum-Welch parameter re-estimation: 3 to 5 iterations (a threshold of 0.01 in log-likelihood was used for convergence) for composite models
- Viterbi forced alignment: when the training dictionary contain multiple pronunciations, the one with the best alignment score is selected
- Baum-Welch parameter re-estimation: 3 to 5 iterations.

Then, in order to increase the system accuracy, first-order context-dependent (CD) models, the so-called triphones, have been also trained. We used phonetic decision trees in order to cluster acoustical similar states in a top-down fashion based on data likelihood criteria. Expert knowledge from Romanian language phonetics has been used by means of over 130 phonetic questions in order to determine contextually equivalent classes of HMM states. Training stage was based on uniform model initialization with the global speech mean and variance. Models are than differentiated by the well-known embedded Baum-Welch procedure.

Time-synchronous Viterbi beam search was the strategy for decoding the unknown utterances. Pruning the search space by beam search was very useful for reducing the computation time.

4.2 Romanian language continuous speech recognizer

The HMMs have three emitting states plus two confluent non-emitting states. The entry state and the exit state are the glue in building up the composite HMM needed for parameter estimation. The output distribution for each emitting state determines the likelihood of the observation data generated by that state. In our case, we have considered single Gaussian continuous density probability function.

Within the HMM paradigm, both training and recognition procedures require estimation of the "best" state sequence. For training this is needed to form new estimates of the model parameters and for recognition the likelihood of the path is used to decide between alternative recognition hypotheses. The state sequence and its likelihood can be found in one of two ways: probabilistically, using total likelihood and deterministically using maximum likelihood.

Probabilistic state sequence estimation consists in computing the total likelihood of a single utterance. Consequently we can find the posterior probability for each observation to be generated by each state. This posterior probability for a state j to be occupied at a given time t is called *occupancy* and the well-known forward-backward algorithm [1] can compute it. It is computed with the formula:

$$\gamma_j(t) = \Pr\left(x(t) = j \middle| O_1, O_2, \ldots O_T\right) = \frac{\alpha_j(t) \cdot \beta_j(t)}{L} \quad (8)$$

where O_i is the observation vector at time i, L is the total likelihood of the data, i.e :

$$L = \Pr\left(O_1, O_2, \ldots O_T\right) \quad (9)$$

and $\alpha_i(t)$, $\beta_j(t)$ are the forward and backward probability, respectively.

Once, the γ coefficients are computed, there is a need for finding some better estimates of the model parameters. By the Baum-Welch procedure the increase in the total likelihood of the observation data, L is granted [14].

For the training process we need not only the acoustical data (vectors extracted from the speech form) but also the associated phonetic transcription. Most of the speech corpuses provide transcription for the acoustical data at the word level. Phonetic transcription can be achieved automatically using a pronunciation dictionary. The basic unit for the training data is the sentence (or phrases) uttered by one speaker, also called utterance. Regarding the phonetic transcription there are two distinct situations:

a) Training data is segmented, i.e. the phonetic transcription contains also information about the class (words or phonemes) boundaries;

b) Training data in un-segmented, i.e. class boundaries are not known.

For language modeling (LM), a loop-grammar (Fig. 11.) was adopted, as it is known to be the most difficult task. The reason for choosing this uniform unigram LM is that the system is sensible to any improvements in acoustic modeling.

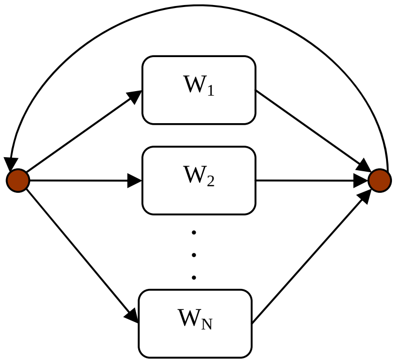

Fig. 11 N-words loop grammar

Recordings were performed with a good quality microphone in noise-free conditions, with a SNR > 40 dB. This clean system has an word error rate (WER) of 14,84 % for monophones and 10,04% for triphones.

4.3 Increasing system robustness

The clean system (trained with clean speech) WER has seriously degraded when we have tested it in mismatch conditions.

For both training and test data we have generated different SNRs phrases in a range between 0 and 25 dB. We have made three groups of experiments for both triphones and monophones:

Clean system: trained with clean speech tested for each SNR

Matched systems: trained and tested with the same SNR

Multistyle training: trained with all phrases (clean + various SNRs) and tested for each SNR

Although the first situation (a) is not met so often for speech corpuses because segmentation and labelling is a laborious time-consuming procedure, usually performed manually, in our experiments on a Romanian Language speech corpus, we have segmented a small number of utterances and used them for initialization of phoneme-based HMMs.

For most of the continuous speech recognition tasks and corpuses like Resource Management (RM), Wall Street Journal (WSJ), etc. there is a huge amount of acoustical training and testing data, consisting in thousands of sentences spoken from tens of speakers, covering hundreds of hours of speech. Segmenting speech data for such tasks is an overwhelming process, only a small part of the corpus having class boundary information.

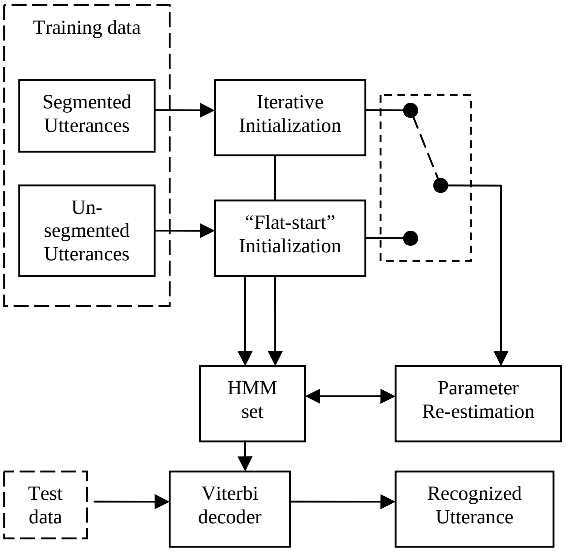

Fig. 12. Initialization scheme within continuous speech recognizer architecture

In this paper we have used phoneme boundaries information for initialization the HMM, covering a small percentage (10%) from the whole database. The experiments were performed on a Romanian Language corpus consisting in over 4000 utterances spoken by 11 speakers.

We will describe now the initialization process. We modeled each Romanian phoneme with a three-state HMM with a left-right topology, using single-mixture Gaussian continuous distribution. The covariance matrices are diagonal in order to reduce the resources required for the output probability computation.

The HMM initialization is based on the concept that the models are the generators of the speech vectors. Every training example can be viewed as the output of a HMM whose parameters are to be estimated.

The first method used for initialization of the models is based on the fact that phoneme boundaries are known, so an initial training is performed for each HMM separately. If the state that generated each vector in the training data was known, then the unknown means and variances could be estimated by averaging all the vectors associated with each state. Similarly, the transition matrix could be estimated by counting the number of time slots that each state is occupied. The main steps could be seen in Fig. 3.

The process is performed in an iterative fashion. First the training data corresponding to a single model is uniformly segmented and each successive segment is associated with successive state. Of course this makes sense only for the left-right topology, otherwise another approach has to be used (for ergodic models).

The second method for HMM initialization is based on making all the models identical initially and then performing embedded training as described in section IV. The idea is simple: the global speech (i.e. all training utterances) mean and covariance is computed and these values are used to initialize the entire set of HMMs, consequently all these being identical. Further, embedded training is used in order to differentiate models.

After the models were initialized by one of the methods presented above, the main training procedure for building the phoneme based system revolves around the concept of embedded training. Unlike the initialization, embedded training simultaneously updates all of the HMMs in a system using all of the training data. We will further describe this process. First, a complete initialized set of HMMs is loaded. Any training utterance has an associated phonetic transcription in which only the sequence of modeled is taking into account, boundary information (if any) being ignored. That means segmented data could also be used in embedded training.

Every utterance is processed in turn as fallows: the sequences of models provided by the phonetic transcription are used to construct a composite HMM by concatenating instances (Fig. 4) of the phone based HMMs. Then, the forward-backward algorithm is applied and the sums needed to form the weighted averages are accumulated.

When all the utterances have been processed, the new parameters estimates are formed from the weighted sums and the updated HMM set is now available for a new iteration. A very important remark is when the system is flat (after second initialization method) a uniform segmentation is assumed.

The mathematical details of embedded Baum-Welch re-estimation are given in [14].

For each re-estimation iteration, there is a good practice to monitor the performance of the models on the test data and stop training when no further improvements are obtained. In our experiments we have traced the average of the log-likelihood per frame, and established a stop iteration threshold of 0.1.

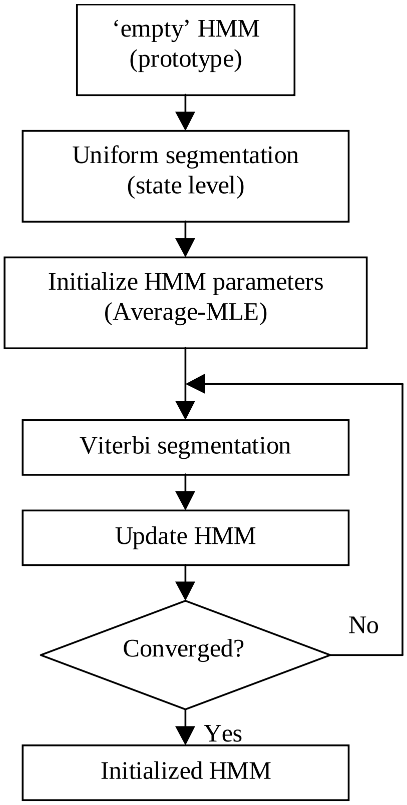

Fig. 13. Initialization HMM algorithm based on segmented training data

We have found that 2 to 5 iterations are enough when training phones models and increasing this number of iterations has some drawbacks:

- Computation time is longer
- Over-training can occur (models can become to closely matched to the training data and fail to generalize well on unseen data)

Table 1 WER for the clean system

Signal to noise ratio	Word error rate	
	CI	CD
0	94,65	97,56
3	96,30	96,20
5	96,30	95,77
12	88,00	84,23
10	83,00	77,00
13	70,14	66,50
15	66,86	53,99
17	53,33	44,20
20	39,62	29,67
22	34,44	20,33
25	22,86	19,15
30	19,02	15,66
>40	14,84	10,04

In Table 1, one may see that the clean system performances are significantly degrading as the SNR is decreasing. Obviously, such a system is not robust at all, having a 30 - 40 % WER for normal room conditions with a SNR of 20 dB.

In Table 2 are presented the experimental results for the matched systems and multistyle trained system.

Table 2 WER for matched and multistyle systems

SNR	CI-HMM		CD-HMM	
	Multi-style	Matched	Multi-style	Matched
0	72,11	46,95	71,55	42,91
5	51,08	34,93	32,77	26,29
10	32,02	26,76	15,96	19,72
15	21,97	19,91	9,01	17
20	19,53	17,93	8,36	14,08
25	20,85	15,31	8,64	12,68
>40	33,43	14,84	10,05	10,04

Comparative plots are given in Fig. 14 and 15 for all experiments conditions included both CI and CD models, respectively.

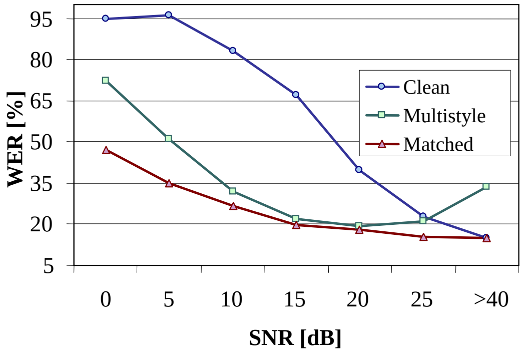

Fig.14 WER vs. SNR, for clean system, multistyle training and matched systems (monophones)

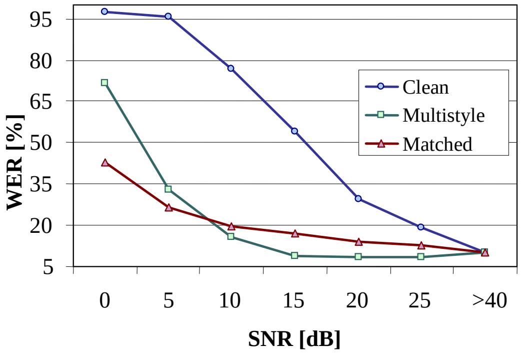

Fig.15 WER vs. SNR, for clean system, multistyle training and matched systems (triphones)

Comparative to the matched systems, multistyle training does not require knowledge of the specific noise level and thus is a viable alternative to the theoretical lower bound of matched conditions.

One could see the multistyle trained system is clearly more robust than the clean system, being almost as good as the matched system. For the monophone case (Fig.14.), the multistyle system has the best performance in 15-25 dB range as it was trained with 0,5,10,15,20,25 and >40 dB.

The same behavior has the multistyle system for the triphone case (Fig.15.) except that WER is biased below the matched system. Because of the diversity of the training data, the resulting multistyle trained system is more robust to varying noise conditions.

5 Conclusion

In this paper we have presented some solutions for increasing the robustness of a Romanian language continuous speech recognizer previously developed. ASR system is based on phoneme HMM in either context independent or context dependent form.

We have presented the architecture of the ASR system insisting on few important aspects such as: continuous speech recognition, voice activity detection for real time applications and context dependent modeling. HMM is state-of-the art technology that offers very good performances for continuous speech recognition having powerful tools for parameters estimation (such as embedded Baum-Welch procedure) and for speech decoding (Viterbi beam search).

By using context dependent models, ASR word error rate decreases significantly as they are capturing the inherent phoneme variability in speech signal (caused by co-articulation). The procedure for triphone training uses phonetic questions for building binary decision trees. This includes expert knowledge about phoneme characteristics, specific to Romanian language. The context-dependent models generated are more specific (but less trainable) but with a better accuracy in speech recognition. In our experiments the best WER was 19,53% for CI models and 8,36% for CD models, in the case of multistyle training, proving a relative decrease with more than 100%.

According to the aspects presented in the initialization stage, we may conclude that flat start initialization of HMM followed by embedded training is good enough (even better than the initialization using segmented speech) to initialize and train the phone models.

Comparing the results for the presented initialization methods flat start conveys to better performance than the hand-labelled segmentation in terms of word recognition rate (WRR), for M system the gain is 4,3 percents absolute and 6,75% relative while for F system the gain is 3,8% absolute and 8,49% relative. The results could be explained by the fact that only 10% from the training data was segmented and used for the first method. This quantity seems to be not enough for best performance, while the uniform segmentation after the second initialization method realizes a partial alignment for enough phonemes from the training utterance. Then, in the subsequent iterations, the models align as intended. Consequently, there is no need for class boundary information in order to initialize each model independently.

It is well known that several environmental factors could affect recognition performances in real world applications. In most cases they are critical and the system accuracy is degrading in mismatch conditions. In order to keep the system accuracy even for high levels of additive noise, we have adopted the so called multistyle training. The ASR system is trained with utterances affected by various levels of artificial additive white Gaussian noise. Consequently, the level of SNR of the training phrases was within a specific range encountered in real situations: 0 to 25 dB. The system was expected to have good performances in these conditions for testing phrases. The experiments proved that very good error rate was obtained at the half of that range while decreasing at the ends. Still the multistyle trained system had very good results comparing to the clean system (trained only with noise free speech signal) but slightly worse than the matched systems that offer a generally lower bound for the WER.

The main advantage of the multistyle training strategy over the matched systems is that it needs only one single ASR that is capable to keep his accuracy in a wide range of SNR. The matched systems are distinct ASRs trained for every specific condition, requiring more memory resources and an additional procedure for SNR estimation in order to select the proper recognition system.

References

[1] D. Munteanu, O. Dumitru, Romanian Language Continuous Speech Recognition by Context-Dependency Modeling, *Int. conf. DOGS 2004, Sombor, Serbia and Montenegro*, 2004, pp. 9 – 12.

[2] E. Oancea, C. Burileanu, D. Munteanu, Continuous Speech Recognition System Improvement, *The 3rd Conference on Speech*

Technology and Human – Computer Dialog (SpeD), 2005, pp. 81-91.

[3] R. P. Lippmann, E. A. Martin, D. P. Paul, Multi-Style Training for Robust Isolated-Word Speech Recognition, *Int. Conf. on Acoustics Speech and Signal Processing*, Dallas, TX, 1987, pp. 709-712.

[4] X. Huang, A. Acero, H.-W. Hon, *Spoken Language Processing – A Guide to Theory, Algorithm, and System Development*, Prentice Hall, 2001.

[5] S.J. Young et all, *The HTK Book*, Cambridge University Engineering Dept., 2002.

[6] A. Acero, L. Deng, T. Kristjansson, J. Zhang, Hmm Adaptation Using Vector Taylor Series for Noisy Speech Recognition, Proc. *of International Conference of Speech and Language Processing*, vol.3, 2000, pp. 869-872.

[7] J.L. Gauvain, L. Lamel, M. Adda-Decker, Developments in Continuous Speech Dictation using the ARPA WSJ Task, *Procedings of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, 1995, pp. 65-68.

[8] T. Robinson, J. Fransen, D. Pye, J. Foote, S. Renals, WSJCAM0: A British English Speech Corpus for Large Vocabulary Continuous Speech Recognition, Proc. *of ICASSP'95*, 1995, pp. 81-84.

[9] M. Matassoni, M. Omologo, D. Giuliani, Hands-Free Speech Recognition Using a Filtered Clean Corpus and Incremental HMM Adaptation, *Proc. ICASSP'00,* 2000, pp. 1407 – 1410.

[10] G. Garau, S. Renals, Combining Spectral Representations for Large-Vocabulary Continuous Speech Recognition, *IEEE Trans. on Audio Speech and Signal Processing*, vol. 16(3), 2008, pp. 508-518.

[11] D. Munteanu, C. Vizitiu, Robust Romanian Language Automatic Speech Recognizer, Proc. of *The 6th WSEAS International Conference on Computational Intelligence, Man-Machine Systems And Cybernetics (CIMMACS '07)*, , 2007, pp. 251-254.

[12] D. Munteanu, C. Molder, C. VIZITIU, Speech Enhancement by Noise Spectral Subtraction, *The 5th International Conference „New Challenges in the Field of Military Sciences 2007"*, November, Budapest, Hungary, 2007.

[13] S.J. Young, The General Use of Tying in Phoneme-Based HMM Speech Recognizers, *Proc. ICASSP'92*, Vol. 1, 1992, pp. 569-572, San Francisco.

[14] J.J. Odell, *The Use of Decision Trees with Context Sensitive Phoneme Modeling, MPhil Thesis*, Cambridge University Engineering Department, 1992

[15] P.C. Woodland, J.J. Odell, V. Valtchev, S.J. Young, Large Vocabulary Continuous Speech Recognition Using HTK, *Proc. ICASSP* 1994, Adelaide, April, 1994.

[16] ETSI- STQ, Aurora group: Distributed Speech Recognition, URL: http://portal.etsi.org/fixed/Quality /SpeechRecognition.asp.

[17] X.D. Huang, A. Acero, H.-W. Hon, *Spoken Language Processing. A guide to Theory, Algorithm and System Development*, Prentice Hall, 2001, pp. 477-544.

[18] J.R. Deller, J.G. Proakis, *Discrete Time Processing of Speech Signals*, PC New York, 1993.

[19] D.P. Munteanu, *Methods for Romanian Language Continuous Speech Recognition*, Ph.D. Thesis, Military Technical Academy, 2006.

[20] S.Sasikumar, S.Karthikeyan, M.Suganthi, Madheswan, *A Narrative Approach for Speech Signal Based MMSE Estimation Using QuantumParameters*,WSEAS TRANSACTIONS on SIGNAL PROCESSING Iss. 12, Vol. 3, December 2007.

[21] M. Borschbach, M. Pyka, *The Effect of Dimension on the Ability of Linguistic Feature Extraction based on Context Preprocessing by ICA*, WSEAS TRANSACTIONS on SIGNAL PROCESSING Iss. 4, Vol. 3, April 2007.