STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LII**, Number 2, 2007

TOWARD A SIMPLE PHONEME BASED SPEECH RECOGNITION SYSTEM

MARGIT ANTAL

Abstract. This paper presents a simple speech recognition system using Gaussian mixtures as phoneme models. The proposed architecture does not follow the integrated search strategy. Instead we use a modular design. We propose two modifications to the Viterbi decoding algorithm in order to be applicable to our phoneme models. Both strategies have been implemented and tested on two corpora. Experiments have proved our phoneme recognition system reliability and its good recognition performance.

1. Introduction

The purpose of this paper is to present our findings during the construction and evaluation of our phoneme based speech recognition system. Despite the fact that good software packages already exist for solving this problem, we decided to develop our own software. The main objective was to use state of the art techniques, but only those which do not contradict human speech recognition. A secondary objective was to simplify the architecture of such a system to the extent of not decreasing the system performance and its usability. Nowadays it is important to create constrained speech recognition systems, which work reasonably in a low resource environment.

State of the art automatic speech recognition (ASR) is based on modelling the phonemes with hidden Markov models (HMM), using the well known three state left to right topology for each phoneme. This model incorporates an inherent phoneme duration, modeled by the state transition probabilities. Several papers [5, 8, 10, 11] noticed the negligible effect of these state transition probabilities on the recognition rate in HMM based ASR, and hence it is usual

Received by the editors: June 2007.

2000 *Mathematics Subject Classification.* 68T10, 62H30.

1998 *CR Categories and Descriptors.* I.5.2 [**Pattern Recognition**]: Design Methodology– *Classifier Design and Evaluation*; I.5.4 [**Computer Systems Organization**]: Special-purpose and Application-based Systems – *Signal Processing Systems* .

to ignore them or use the same value for each transition probability. Due to this observation we modeled every phoneme with a one state HMM, which can be considered as a Gaussian mixture (GMM).

Phoneme duration is an important problem for speech comprehension, especially in languages like Hungarian, in which most of the phonemes has both a short and long form. These durations are so important that even the written language uses different letters for each vowel, one for the short and one for the long form of the same phoneme. In the case of consonants there are no different written forms, but the letter is doubled. Good duration modelling can therefore be a major issue in these languages, not only for speech recognition but for speech synthesis too.

As a first step we performed some phoneme classification experiments in order to evaluate our phoneme models. These GMM phoneme models performed so well that we could go further to the problem of phoneme recognition. In this step we had to modify the classic Viterbi decoding algorithm (given by formula (13)) in order to make it suitable for GMM phoneme models. We should mention that the classic Viterbi algorithm without state transition probabilities (omitting the state transition matrix a_{ij} from formula (13)) made a huge number of insertion errors. In order to overcome this, our first attempt was to introduce explicit phoneme durations computed from speech corpora into the decoding process. The idea was taken from [9], but we simplified it. Levinson used statistical models for phoneme duration modelling (Gamma distributions), we used only the minimum and maximum duration for each phoneme. We went even further in simplifying this duration modelling by using the same fixed durational minimum and maximum for each phoneme. As this increased the decoding algorithm complexity by a factor D, which is the maximum duration of phonemes, we tried to find a cheaper solution for decoding.

In the second attempt we made another adaptation of the Viterbi algorithm for monophone one-state models, which introduces an empirical constant in order to be able to control the insertion errors. As we prove experimentally, this constant incorporates the average phoneme duration implicitly. This was our conclusion after we have tuned this parameter for two corpora: TIMIT the well-known English corpus and OASIS, a small Hungarian corpus for isolated word recognition.

Several measurements were carried out in order to demonstrate the viability of this simplified strategy. Whenever it was possible, we compared our results with other results obtained on the same corpora, eventually using different phoneme modelling techniques. However, the purpose was not to outperform state of the art technologies, but the construction of such a system that is simple and efficient for some constrained speech recognition tasks.

This paper is organized as follows. Section 2 presents the architecture of our system, the feature extraction module, the acoustic-phonetic module, which is a standard GMM and the decoding module in which we propose modifications to the Viterbi algorithm. In section 3, we present the corpora used for experiments and the evaluations of the proposed decoding methods. We end with discussion and conclusions.

2. The recognition system

Our speech recognition system has a very simple modular architecture. The first module of the system is the feature extraction module. The extraction of Mel-frequency cepstral coefficients (MFCC) is presented in subsection 2.1. In this module only standard methods were used as recommended in [3, 5, 12].

The second module is the acoustic-phonetic module. We used Gaussian mixture models for training the phonemes based on phonetically segmented and annotated corpora. Once this stage is completed we can evaluate the phoneme models. As we have stated already, we used context independent phonemes modeled by Gaussian mixtures.

The third module is the phonetic decoding module containing two modified versions of the Viterbi decoding algorithm.

2.1. **Feature extraction module.**

The extraction of reliable features is a very important issue in speech recognition. There are a large number of features we can use. Among others we can use is the speech waveform itself. However this has two main shortcomings. The first one is the dimension of this feature, and the second one is that time domain features are much less accurate than frequency-domain features. In the following we present the extraction of Mel-frequency cepstrum coefficients used in our system. This was implemented based on [5].

In our system the acoustic analysis of the speech signal was done by short-time spectrum analysis with 20 ms frames and 10 ms overlap between consecutive frames. For a frame length of 20 ms it can be assumed that the speech signal is stationary, allowing the computation of short-time Fourier spectrum.

Let us denote by $x[n],\ n = 0, 1, \dots, N-1$ the samples from a frame. For this input signal we compute the Discrete Fourier Transform (DFT):

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j(2\pi/N)kn},\ k = 0, 1, \dots, N-1 \tag{1}$$

In order to reduce the dimension of the feature vector a filterbank composed of M triangular filters was used. The equation of the mth triangular filter is the following.

$$H_m[k] = \begin{cases} 0 & k < f[m-1] \\ \frac{k - f[m-1]}{f[m] - f[m-1]} & f[m-1] \leq k \leq f[m] \\ \frac{f[m+1] - k}{f[m+1] - f[m]} & f[m] \leq k \leq f[m+1] \\ 0 & k > f[m+1] \end{cases} \tag{2}$$

Such filters compute the average spectrum around each center frequency with increasing bandwidths.

Let us denote by f_l and f_h the lowest and the highest frequencies of the filterbank in Hz, F_s the sampling frequency in Hz, M the number of filters, and N the size of DFT.

The filterbank's boundary points $f[m]$ are uniformly spaced in the mel-scale:

$$f[m] = \frac{N}{F_s} B^{-1}(B(f_l) + m \frac{B(f_h) - B(f_l)}{M+1}) \tag{3}$$

where the mel-scale B is given by

$$B(f) = 1125 ln(1 + f/700) \tag{4}$$

and B^{-1} is its inverse

$$B^{-1}(b) = 700(e^{\frac{1}{1125}} - 1) \tag{5}$$

The next step is the computation of log-energy at the output of each filter

$$S[m] = ln(\sum_{k=0}^{N-1} |X[k]|^2 H_m[k]), \quad 1 \leq m \leq M \tag{6}$$

The Mel-frequency cepstrum is then the discrete cosine transform of the M filter outputs:

$$c[n] = \sum_{m=0}^{M-1} S[m] cos(\frac{\pi n}{M}(m - \frac{1}{2})) \quad 0 \leq n < M \tag{7}$$

For speech recognition applications it is typical to use a number of filters M between 24 and 40 and to evaluate only the first 13 coefficients given by equation (7). In our experiments we used $M = 28$ filters.

Temporal changes in spectra play an important role in human perception. One way to capture this information is to use delta coefficients that measure the change in coefficients over time. Delta features were obtained by evaluating the first and the second order delta cepstral coefficients given by the following equations

$$\Delta c_k = \frac{2(c_{k+2} - c_{k-2}) + (c_{k+1} - c_{k-1})}{10} \tag{8}$$

$$\Delta\Delta c_k = \frac{2(\Delta c_{k+2} - \Delta c_{k-2}) + (\Delta c_{k+1} - \Delta c_{k-1})}{10} \tag{9}$$

where c_k represents the feature vector containing the first 13 MFCC coefficients obtained using formula (7) for the kth time frame.

The combined cepstral, first and second order delta cepstral vectors form a set of 39-parameter feature vector (observation vector) $o_k = \begin{pmatrix} c_k \\ \Delta c_k \\ \Delta\Delta c_k \end{pmatrix}$, which were used in all the experiments described in this paper.

2.2. **The Acoustic-Phonetic module.** Observation densities in phonemes are modeled by mixtures of multivariate Gaussians. The proper number of Gaussians can be estimated separately for every phoneme or can be fixed the same value for every phoneme. We used the latter approach. Let us denote by M the number of Gaussian densities. In this case the observation density function for phoneme i, $b_i(\overrightarrow{o_t})$ has the form

$$b_i(\overrightarrow{o_t}) = \sum_{j=1}^{M} w_{ij} \cdot \frac{1}{(2\pi)^{D/2}|\Sigma_{ij}|^{1/2}} \cdot e^{-\frac{1}{2}(\overrightarrow{o_t} - \overrightarrow{\mu_{ij}})^T \Sigma_{ij}^{-1} (\overrightarrow{o_t} - \overrightarrow{\mu_{ij}}))} \tag{10}$$

where D represents the dimensionality of the $\overrightarrow{o_t}$ observation (feature vector), $\overrightarrow{\mu_{ij}}$ and Σ_{ij} are the mean vector and the covariance matrix for the jth mixture component. For every phoneme the mixture weights sum to unity ($\sum_{j=1}^{M} w_{ij} = 1$) in order to have a true probability function.

The complete model thus consists of the set of n phonemes, the ith phoneme being modeled by a GMM with the parameters $(w_{ij}, \overrightarrow{\mu_{ij}}, \Sigma_{ij}), 1 \leq j \leq M$. $\overrightarrow{\mu_{ij}}$ is a mean vector composed by D real numbers. We used diagonal covariance matrix, hence it can be represented by D real numbers. The complete model can be represented using $n * M * (1 + D + D) = n * M * (2D + 1)$ real numbers.

2.3. **Phonetic decoding module.** Let us denote by $O = \{\overrightarrow{o_1}, \overrightarrow{o_2}, \ldots \overrightarrow{o_T}\}$ the acoustic observation sequence, which has to be decoded into a phoneme sequence. The set of all phoneme sequences will be denoted by F. Essentially the task here is to find $\hat{f} \in F$ defined by

$$\hat{f} = \arg\max_{f \in F} P(f|O) = \arg\max_{f \in F} \frac{P(O|f) \cdot P(f)}{P(O)} \tag{11}$$

where $P(f)$ is known as the phonetic language model. Assuming that any observation sequence is equally likely, equation (11) becomes

$$\hat{f} = \arg\max_{f \in F} P(O|f) \cdot P(f) \tag{12}$$

Equation (12) expresses that we face a search problem. Phonetic transcription reduces to the task of finding the most likely phoneme sequence for the input sequence of acoustic vectors.

For HMM phoneme models Viterbi algorithm [5] solves the problem of finding the most probable state sequence.

We review the classic Viterbi algorithm, which will be adapted to our phoneme models in the following section. Let $\alpha_t(j)$ denote the maximum likelihood of $\overrightarrow{o_1}, \ \overrightarrow{o_2}, \ \ldots \overrightarrow{o_t}$ over all state sequences terminating in state j. This quantity can be evaluated recursively according to

$$\alpha_t(j) = \max_{1 \leq i \leq n} [\alpha_{t-1}(i) \cdot a_{ij}] \cdot b_j(\overrightarrow{o_t}) \tag{13}$$

where a_{ij} represents the state transition probability between state i and state j, $b_j(\overrightarrow{o_t})$ is the observation probability of $\overrightarrow{o_t}$ in state j

At every time instance we retain $B_t(j)$ =arg $\max_{1 \leq i \leq n} [\alpha_{t-1}(i) \cdot a_{ij}]$, $1 \leq j \leq n$ in order to be able to back trace the optimal path through the trellis.

The Viterbi algorithm presented previously is based on dynamic programming technique. Essentially it is a planar search algorithm through a lattice, where the lattice consists of points representing phoneme likelihoods for each

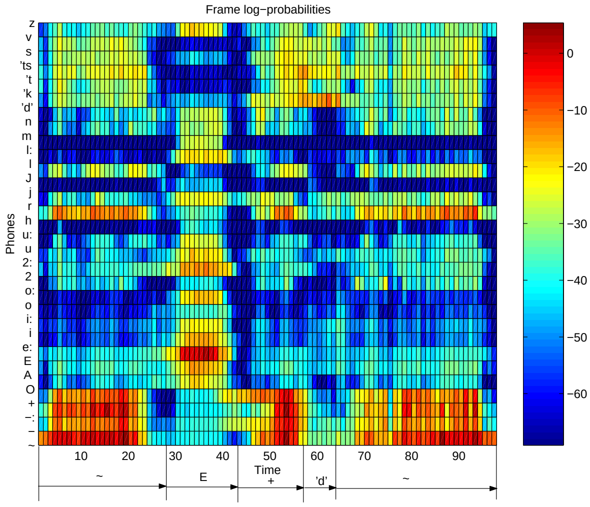

FIGURE 1. Search space

time instance. This search space is shown for the digit one (”egy” in Hungarian) in figure 1.

2.3.1. *Explicit usage of durations.* According to paper [14] it can be useful for the recognition stage to set a minimum number of frames, which can constitute a phoneme. This setting will help the decoder to decrease the number of insertion errors.

Instead of using complicated durational models we propose a simple modification of equation (13), which can be expressed as

$$\alpha_t(j) = \max_{1 \le i \le n} \left\{ \max_{\tau_{min} \le \tau \le \tau_{max}} \left\{ \alpha_{t-\tau}(i) \cdot a_{ij} \cdot \prod_{\Theta=0}^{\tau-1} b_j(\overrightarrow{o_{t-\Theta}}) \right\} \right\} \tag{14}$$

for $1 \le j \le n$, $1 \le t \le T$, where τ_{min} and τ_{max} are the minimum and maximum allowable durations for any phonetic unit. It is supposed that observations are independent then $\prod_{\Theta=0}^{\tau-1} b_j(\overrightarrow{o_{t-\Theta}})$ computes the probability of the observation sequence $\overrightarrow{o_{t-\tau+1}}, \overrightarrow{o_{t-\tau+2}}, \ldots \overrightarrow{o_t}$ in the state j. Esentially we compute this probability for every allowed length τ, where $\tau_{min} \le \tau \le \tau_{max}$. Retaining at each stage of the recursion the values i and τ that maximize (14), makes possible back tracing through $\alpha_t(j)$ in order to obtain the best state and duration sequences.

In our system every phoneme is modeled by a one-state HMM so we could not use state transition probabilities, instead we used the following formula:

$$a_{ij} = \begin{cases} 1, & phone_j\ is\ allowed\ to\ follow\ phone_i \\ 0, & otherwise \end{cases} \tag{15}$$

which can be seen as a very simple language model. Using (15) reduces substantially the search space.

We computed the minimum $\tau_{min}(j)$ and maximum duration $\tau_{max}(j)$ of every phoneme $j = 1 \dots n$, which were incorporated in formula (14) resulting in

$$\alpha_t(j) = \max_{1 \le i \le n} \left\{ \max_{\tau_{min}(j) \le \tau \le \tau_{max}(j)} \left\{ \alpha_{t-\tau}(i) \cdot a_{ij} \cdot \prod_{\Theta=0}^{\tau-1} b_j(\overrightarrow{o_{t-\Theta}}) \right\} \right\} \tag{16}$$

Section 3 presents experiments using both formulae: (14), (16).

2.3.2. *Implicit duration modelling.* Another approach to phoneme decoding is to use the Viterbi algorithm directly for the context independent phoneme models. In this case the state transition probabilities do not exist and we should omit in equation (13). Omitting state transition probabilities resulted in a huge number of insertion errors, which should be somehow overcome.

Firstly, we used the logarithmic form of Viterbi approximation as shown in the following formula:

$$\log \alpha_t(j) = \max_{1 \le i \le n} \{\log \alpha_{t-1}(i) + \log a_{ij}\} + \log b_j(\overrightarrow{o_t}) \tag{17}$$

Instead of omitting the term $\log a_{ij}$, we propose replacing it by I_{ij}, which is given as

$$I_{ij} = \begin{cases} \beta,\ if\ i = j \\ 0,\ otherwise \end{cases} \tag{18}$$

with $\beta > 0$, and the final formula for Viterbi approximation became:

$$\log \alpha_t(j) = \max_{1 \le i \le n} \{\log \alpha_{t-1}(i) + I_{ij}\} + \log b_j(\overrightarrow{o_t}) \tag{19}$$

Because larger β values will result in larger phoneme durations in the decoded phoneme sequence, this decoding process incorporates implicitly the average phoneme duration. It can be proved experimentally that the optimal value of the β parameter and the average phoneme duration for a given language are directly proportionals. We should note that our method proposed for decoding is very similar to that proposed by Robinson in [13]. Robinson used a recurrent neural network for phoneme classification and for the decoding process he used a dynamic programming approach. In the decoding formula it was introduced a transitional cost, similar to the state transition probability in HMM. He worked with distances instead of probabilities, but the ideas are very similar. Moreover, he tried to introduce duration information and bigram probabilities into the transition function and observed that these additional information did not increased significantly the recognition accuracy.

3. Experiments

For measurements we used our software written in C++ language, which has a modular design being composed by a signal processing module for MFCC feature extraction, a Gaussian mixture module and a decoder module. The signal processing and the Gaussian mixture modules were successfully used for speaker identification systems too[1].

3.1. **Evaluation.** The standard evaluation metric for phoneme recognition systems is the phoneme error rate (PER). The PER measures the difference between the phoneme string returned by the recognizer and the correct reference transcription. The distance between the two phoneme strings is computed by the classical minimum edit distance algorithm [5]. The result of computation will be the minimum number of phoneme substitutions, insertions and deletions necessary to map between the correct and hypothesized strings. This can be expressed by the formula

$$PER = 100 \cdot \frac{I + S + D}{N} \tag{20}$$

where N represents the number of phonemes in the correct transcription, I, S and D represent the number of insertions, substitutions and deletions. Recognition accuracy is computed as

$$A = 100 - PER \tag{21}$$

Another performance measure could be the number of correct phonemes returned by the recogniser, which can be computed by the minimum distance algorithm. This will be denoted by C.

3.2. **Corpora.** We used two corpora for the experiments, the first one was TIMIT, a well known American English corpus, and the second one was OASIS Numbers, a small Hungarian corpus designed for number recognition. Because TIMIT is well known, we describe shortly only the Hungarian corpus.

The OASIS corpus is a small isolated -number corpus being developed at the Research Group on Artificial Intelligence of the Hungarian Academy of Sciences. The segmented part of the corpus contains speech from 26 speakers: 1 child, 9 female and 16 male voices. Each speaker reads the same 26 words twice. Any Hungarian number can be formed by concatenation from this set of 26 numbers. Each word is manually segmented and labelled phonetically. Twenty speakers were used for training and six speakers for testing. The corpus contains 31 phonemes, annotated using SAMPA symbols. The only modification was made for the notation of stop symbols, where instead of using one symbol, the closure part and the burst part were annotated separately. For the voiceless closure part it is used the symbol $/-/$ while for the voiced closure part the symbol $/+/$. For example instead of the symbol $/t/$ it were used two symbols:$/-/$and $/'t/$. Further information on this corpus can be found in [6, 7, 15]. Table 1 presents the exact content of the corpus used for training and test.

For the TIMIT corpus the phoneme models consist of $61*32*(2*39+1) = 154208$ real numbers and for the OASIS corpus $31*16*(2*39+1) = 39184$ real numbers.

3.2.1. *Explicit usage of durations.* Our first attempt to phoneme duration modelling was a data driven approach. The minimum, maximum and the average phoneme durations were determined based on corpora. Figure 2 shows these values for the OASIS corpus.

In the first three experiments we used the same minimum and maximum duration for every phoneme and in the fourth experiment we introduced those shown in figure 2. Results are reported in table 2.

3.2.2. *Implicit duration modelling.* Using the second approach, firstly we present the phoneme recognition experiments for the TIMIT corpus. We used all the 61 phonemes of the corpus without grouping allophones. This will serve as a baseline for further system improvements. The first step was to determine the optimal value for β parameter introduced to the Viterbi decoding algorithm.

For training we used the whole training part of the corpus. For evaluation we used two sets, a smaller *timit_test_core* and a larger one *timit_test*, both had

Phoneme	Training	Test	Phoneme	Training	Test
-	519	156	i:	40	12
-:	40	12	j	80	24
'd'	118	36	J	80	24
'k	200	60	l	200	60
't	399	120	l:	40	12
'ts	200	60	m	120	36
+	120	36	n	559	168
˜	2080	624	O	240	72
2	80	24	o	160	48
:2	40	12	o:	40	12
A:	120	36	r	160	48
E	600	180	s	160	48
e:	160	48	u	80	24
h	306	96	u:	40	12
i	240	72	v	240	72
			z	240	72

Table 1. Phoneme frequencies in training and test part of the corpus

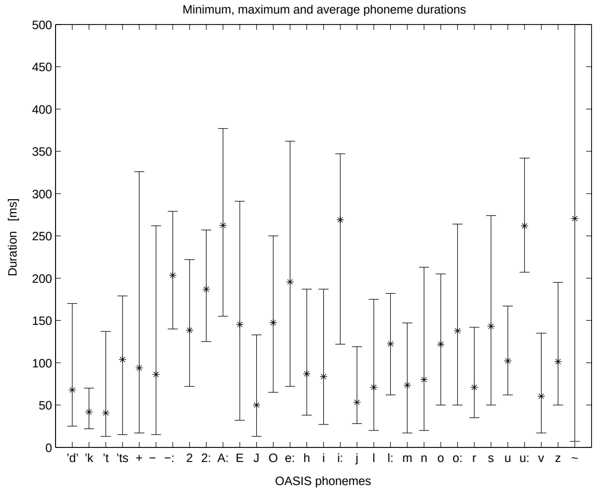

Figure 2. OASIS-Minimum, maximum and average phoneme durations

Dur.	D	I	S	A	C
1..50	0.04%	73.10%	2.54%	24.30%	97.40%
2..50	0.17%	27.72%	3.45%	68.65%	96.37%
3..50	0.69%	17.09%	4.66%	77.54%	96.64%
$\tau_{min}..\tau_{max}$	1.12%	15.71%	6.21%	76.94%	92.65%

Table 2. Phoneme recognition -OASIS - explicit phoneme durations

Evaluation	β	D	I	S	A	C
timit_test_core	11	7.41%	7.68%	37.76%	47.14%	54.81%
7525 phonemes	13	9.75%	4.81%	36.37%	49.06%	53.87%
TR: 61 models	15	12.14%	3.21%	34.79%	49.84%	53.06%
TE: 61 models	17	14.35%	2.19%	33.27%	50.17%	52.37%
	20	18%	1.26%	30.88%	49.86%	51.12%
timit_test	11	6.79%	7.39%	34.54%	51.27%	58.66%
65825 phonemes	13	9.13%	4.76%	33.12%	52.98%	57.74%
TR: 61 models	15	11.42%	3.24%	31.66%	53.67%	56.91%
TE: 61 models	17	13.7%	2.21%	30.18%	53.90%	56.11%
	20	17.16%	1.31%	28.12%	53.40%	54.71%
timit_test	11	7.03%	7.63%	25.68%	59.65%	67.28%
65825 phonemes	13	9.35%	4.97%	24.48%	61.18%	66.16%
TR: 61 models	15	11.55%	3.37%	23.38%	61.68%	65.05%
TE: 39 models	17	13.81%	2.32%	22.16%	61.70%	64.00%
	20	17.22%	1.37%	20.45%	60.95%	62.32%

Table 3. Phoneme recognition - TIMIT

been proposed by the creators of the corpus. While the smaller set contains 192 sentences, the larger one is formed by 1680 sentences. Table 3 shows the recognition accuracies together with the three type of errors for various values of the β parameter. As phoneme models 32 Gaussians models were used with diagonal covariance matrices. The first evaluation set *timit_test_core* contains 7525 phones, while *timit_test* contains 65825 phones. These results were obtained without using language model, which means that it was allowed for every phoneme to follow every other phoneme.

In the third part of the table 3 we present the results obtained by the same experiments with a modification in the interpretation of the decoding process.

Paper	Method	LM	TR	TE	ACC.	CORR.
Ostendorf[4]	SSM+CI	bigram	61	39	64.20%	70.00%
This paper	GMM+CI	0gram	61	39	61.70%	64.00%
Robinson[13]	REPN+CD	0gram	61	61	61.70%	69.10%
Robinson[13]	REPN+CD	bigram	61	61	63.50%	70.00%
Robinson[13]	REPN+CD	bigram	61	39	69.80%	76.50%

Table 4. TIMIT - Phoneme recognition. LM - Language Model, TR - Number of phoneme models trained, TE - Number of phoneme models for decoding, ACC -Accuracy, CORR -Correct, SSM - Stochastic Segment Model, REPN - Recurrent Error Propagation Network, CI - Context Independent phoneme models, CD - Context Dependent phoneme models

We used 61 phoneme models for decoding, but before applying the minimum distance algorithm, we converted the 61 phonemes to the 39 phoneme groups, as suggested by the creators of the corpus. This step reduced substantially the substitution errors, which suggests us that the phoneme grouping influences mainly the substitution errors.

Table 4 presents comparative results obtained on TIMIT. It can be seen that the best results were obtained by the neural network modelling, however this model is not fully comparable to the other two papers because this represents a context dependent modelling of the phonemes.

In the following we present results obtained for the Hungarian corpus. In this case, due to the limited amount of training data we used as phoneme models mixtures of 16 Gaussians.

Table 5 presents the recognition results obtained for various values of the β parameter. Figure 3 shows β parameter tuning for both corpora.

Our results compare favorably with those published in [15] on this corpus. They reported 82.05% recognition accuracy, using a hybrid ANN-HMM framework.

4. Discussion and conclusions

One of the most important finding of this work is that we achieved very good phoneme recognition accuracy with a very simple phoneme modelling and an even simpler phonetic decoding strategy. The second decoding strategy, in which we introduced the parameter β performed better than the first

β	D	I	S	A	C
11	0.52%	27.93%	5.18%	66.36%	94.30%
13	0.82%	21.80%	5.53%	71.85%	93.65%
15	1.42%	17.35%	5.65%	75.56%	92.91%
17	1.68%	14.35%	5.95%	78.02%	92.05%
20	2.37%	11.27%	6.00%	80.35%	91.62%
23	3.49%	9.15%	5.82%	81.51%	90.67%
25	4.06%	7.77%	5.78%	82.38%	90.15%
27	4.83%	7.08%	5.83%	82.25%	89.33%
30	6.43%	5.82%	5.70%	82.03%	87.86%

Table 5. Phoneme recognition - OASIS - implicit duration modelling

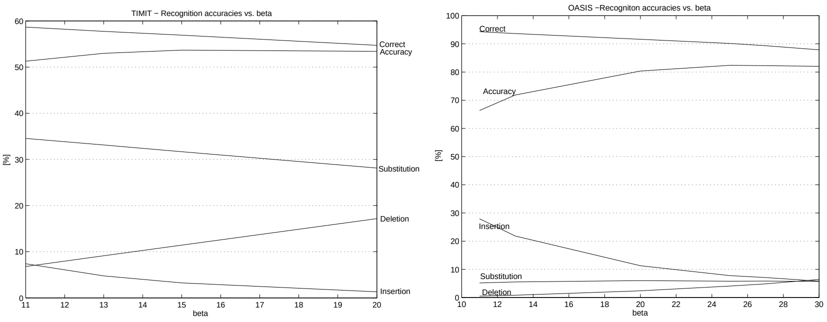

Figure 3. Phoneme recognition vs. β parameter

one, which considers different phoneme durations. Not only the recognition accuracy is better in the second approach, but the algorithm itself is a very efficient one.

Most of the papers working with the TIMIT corpus report phoneme recognition results for the reduced phoneme set. In order to produce comparable results, before computing the minimum edit distance between the recognised phoneme string and the original one, we converted the phonemes to their phoneme groups. This yields a better recognition accuracy, decreasing especially the substitution errors. In this way working with the reduced phoneme set increased approximately with 8% the recognition accuracy. For the 39 phoneme groups we obtained 61.70% recognition accuracy without using any phoneme level language model. The first decoding approach was not used for

TIMIT as this algorithm is a very inefficient one and has increased the time for decoding.

For the OASIS corpus both the proposed decoding techniques were evaluated. For the first decoding technique we have found that imposing a minimum phoneme duration (3 frames in our case) yields the same good result as using the phoneme specific minimum and maximum durations. This could be due to the limited amount of training data in this corpus. We should note that 3 frames roughly corresponds to the average of minimum durations over the whole phoneme set. The second decoding technique has shown its superiority over the first one. With the parameter β tuned for maximum accuracy we obtained 82.38% recognition accuracy, which compares favorably to 82.05% found in [15].

We believe that the most important finding is that we obtained these results by using only models and algorithms which do not contradict in their functionality human speech recognition. Despite the fact that phoneme recognition accuracy was not increased, our simple phoneme recognition system warrants stable and reliable behaviour with a good recognition performance.

References

[1] Antal, M., Toderean, G., Speaker Recognition and Broad Phonetic Groups, Proc. 24th IASTED International Multi-Conference on Signal Processing, Pattern Recognition and Applications, Febr. 15-17, Innsbruck, Austria, pp. 155-158, 2006.

[2] Bourlard, H., Hermansky, H., Morgan., N., Towards Increasing Speech Recognition Error Rates, Speech Communication, Vol. 18., pp. 205-231, 1996.

[3] Deller, J.R., Hansen, J. H. L., Proakis, J. G., Discrete-Time Signal Processing of Speech Signals, John Wiley & Sons, 2000.

[4] Digalakis V., Ostendorf M., Rohlicek, J. R., Fast Search Algorithms for Connected Phone Recognition Using the Stochastic Segment Model, IEEE Trans. on Signal Processing, December, pp. 173-178, 1992.

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

[6] Kocsor, A., Kuba, A. Jr., Toth, L., An Overview of the OASIS Speech Recognition Project, Proceedings of the 4th International Conference on Applied Informatics, August 30 - September 3, Eger-Noszvaj, Hungary, pp. 94-102, 1999.

[7] Kocsor, A., Toth, L., Kuba, A., Kovacs, K., Jelasity, M., Gyimothy, T., Csirik J., A Comparative Study of Several Feature Transformation and Learning Methods, International Journal of Speech Technology, pp. Vol. 3, Nr 3/4, pp. 253-262, 2000.

[8] Lee, K-F., Hon, H-W., Speaker-independent Phone Recognition Using Hidden Markov Models, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 37, No. 11, 1989.

[9] Levinson, S. E., Ljolje, A., Miller, L. G., Continuous Recognition from Phonetic Transcription, Proceedings of a workshop on Speech and Natural Language, Pennsylvania, U.S., pp. 190-199, 1990.

[10] Mari, J. F., Fohr, D., Junqua, J-C., A second order HMM for high performance word and phoneme-based speech recognition, IEEE Transactions on Speech and Audio Processing, vol. 23, no. 2, pp. 435-438, 1996.

[11] Pylkkonnen, J., Phone Duration Modeling Techniques in Continuous Speech Recognition, Master thesis Helsinki University of Technology, 2004.

[12] Rabiner, L., R., Juang, B.H., Fundamentals of Speech Recognition, Prentice-Hall, Englewood Cliffs, NJ, 1993.

[13] Robinson, T., Fallside, F., A Recurrent Error Propagation Network Speech Recognition System, Computer Speech and Language, vol. 5, no. 3, pp. 259-274, 1991.

[14] Toth, L., Kocsor, A.: Explicit Duration Modelling in HMM/ANN Hybrids, Matousek et al. (eds.): Proceedings of TSD 2005, LNAI 3658, pp. 310-317, Springer, 2005.

[15] Toth, L., Kocsor, A., Csirik, J., On naive Bayes in Speech Recognition, In. J. Appl. Math. Comput. Sci., Vol. 15, No. 2, pp. 287-294, 2005.

Sapientia - Hungarian University of Transylvania,Faculty of Technological and Human Sciences, 540053 Tg.-Mures,540485, Soseaua Sighisoarei 1C, Romania
E-mail address: `manyi@ms.sapientia.ro`