Recognition of Phoneme Strings using TRAP Technique

Petr Schwarz , Pavel Matějka, Jan Černocký

Brno University of Technology, Czech Republic
OGI School of Science & Engineering, OHSU, Portland, Oregon USA
{schwarzp,cernocky}@fit.vutbr.cz matejkap@feec.vutbr.cz

Abstract

We investigate and compare several techniques for automatic recognition of unconstrained context-independent phoneme strings from TIMIT and NTIMIT databases. Among the compared techniques, the technique based on TempoRAl Patterns (TRAP) achieves the best results in the clean speech, it achieves about 10% relative improovements against baseline system. Its advantage is also observed in the presence of mismatch between training and testing conditions. Issues such as the optimal length of temporal patterns in the TRAP technique and the effectiveness of mean and variance normalization of the patterns and the multi-band input the TRAP estimations, are also explored.

1. Introduction

Our goal is to design a front-end module that would deliver language and task independent posterior probabilities of sub-word units such as phonemes together with an information about their temporal extent. There should be no language model used or any other constraint so that the system may be used for a keyword spotting, speaker identification, language identification or recognition of out-of-vocabulary words.

A number of related works on the TIMIT database were done and published in the past but it is sometimes hard to compare results. The databases and their cuts may be different, or different language models are used. Some report results on a recognition task – that is the task includes determining the segment boundaries. Another works report results on a classification task – the segmentation into units is known and the task is merely to determine the class from which the known segments come from. One of the most relevant works is Lee's and Hon's [1]. They use discrete Hidden Markov Models (HMMs) and the LPC parameterization and propose collapsing of 61 TIMIT labels to 39 separate categories for testing purpose. Robinson and Fallside [2] were investigating recurrent neural nets on this task. Their results appear to be the best results reported on this task. Rathinavelu and Deng [3, 4] improve accuracy of HMM by extending this method with some additional parameters, which are derived from training data. Zahorian at al. use 2D DCT features for the phoneme classification task [5].

In this contribution, we are looking closer at the input parametrization. Our experimental system is an HMM - Neural Network (HMM/NN) hybrid. It has less parameters comparing to traditional HMM systems, and is capable of handling correlated multiple frames of features. The one-state context-independent phoneme models are used. In our preliminary experiments, this system achieved about the same results as a conventional HMM system.

The baseline setup use 13 Mel Frequency Cepstral Coefficients (MFCCs), including C_0, deltas and double deltas (referred as MFCC39). Multi-frame input [6] is also studied and applied.

Next we investigate the TRAP parameterization technique [8]. In this technique, frequency-localized posterior probabilities of sub-word units (phonemes) are estimated from temporal evolution of critical band spectral densities within a single critical band. Such estimates are then used in another class-posterior estimator which estimates the overall phoneme probability from the probabilities in the individual critical bands. This technique was demonstrated efficient in noisy environment [7, 8]. The TRAP technique is compared with MFCC and with multiple frames of MFCC. Test is performed on well-matched training/test conditions as well as in mismatch conditions.

The last part of this contribution is investigating issues such as the optimal length of temporal patterns in the TRAP technique, the effectiveness of mean and variance normalization of the patterns, and the use of more than one critical band as an input to the TRAP probability estimator.

2. TRAP system

Critical bands energies are obtained in the conventional way. Speech signal is divided into $25\ ms$ long frames with $10\ ms$ shift. The Mel filter-bank is emulated by triangular weighting of FFT-derived short-term spectrum to obtain short-term critical-band logarithmic spectral densities. TRAP feature vector describes a segment of temporal evolution of such critical band spectral densities within a single critical band. The usual size of TRAP feature vector is 101 points [8]. The central point is actual frame and there are 50 frames in past and 50 in future. That results in 1 second long time context. The mean and variance normalization can be applied to such temporal vector. Finally, the vector is weighted by Hamming window. This vector forms an input to a classifier. Outputs of the classifier are posterior probabilities of sub-word classes which we want to distinguish among. In our case, such classes are context-independent phonemes. Such classifier is applied in each critical band. The merger is another classifier and its function is to combine band classifier outputs into one. The described techniques yields phoneme probabilities for the center frame. Both band classifiers and merger are neural nets. The complete system is shown in fig 1.
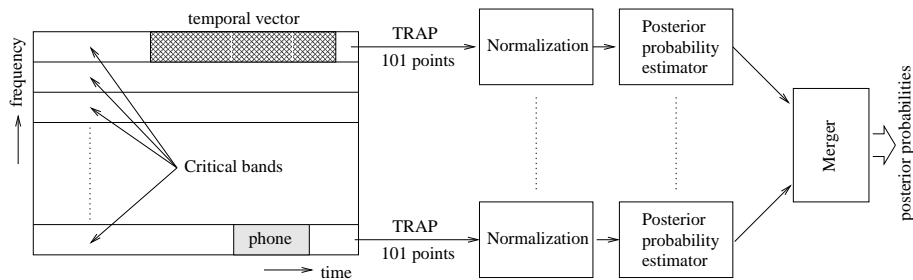
Figure 1: *TRAP system*

3. Experimental setup

3.1. Software

A Quicknet tool from the SPRACHcore package [9], employing three layer perceptron with the softmax nonlinearity at the output, was used in all experiments with neural networks. The HTK toolkit [10] was used in experiments with conventional HMM.

3.2. Phoneme set

The phoneme set consists of 39 phonemes. It is very similar to the CMU/MIT phoneme set [1], but closures were merged with burst instead of with silence (bcl b $\rightarrow$ b). We believe it is more appropriate for features which use a longer temporal context such as TRAP and multi-frame MFCC.

3.3. Databases

Two databases were used – TIMIT and NTIMIT. As known, NTIMIT was created by passing TIMIT through a fixed telephone network. Therefore, using of both databases allows for evaluating systems in both well-matched condition and in presence of mismatch between training and testing database. All SA records were removed as we felt that the phonetically identical sentences over all speakers in the database could bias the results. Databases were divided into three parts – training (412 speakers), cross-validation (50 speakers) and test (168 speakers). An original TIMIT/NTIMIT training part was split into two subsets – the training subset and the cross-validation subset. The same split was applied to both the TIMIT and the NTIMIT databases. Both databases were down-sampled to $8000Hz$.

3.4. Evaluation criteria

Classifiers were trained on the training part of the database. In case of NN, the increase in classification error on the cross-validation part during training was used as a stopping criterium to avoid over-training. There is one ad hoc parameter in the system, the word (phoneme) insertion penalty, which has to be set. This constant was tuned to the equal number of inserted and deleted phonemes on the cross-validation part of the database. The setting of this constant is very different when the testing condition does not match the training condition. Results were evaluated on the test part of database. Number of substitution, deletion and insertion errors are reported, as well as a sum of this three numbers - the phoneme error rate (PER).

As it is difficult to compare results when the number of parameters in the classifier varies, an important issue, i.e. how to deal with sizes of a classifiers, had to be addressed. One possibility was to fix the number of parameters in the classifier and always reduce the input vector size by a linear transformation to a fixed one. However, since the dimensionality reduction always implies a loss of information, a bottle-neck could be created. Therefore, in our experiments, we opted for a different solution in which the optimal size of classifier – number of neurons in the hidden layer and/or number of the Gaussian components in the mixture – was found for each experiment. A simple criterion – minimal phoneme error rate was used for this purpose.

4. Experimental results

4.1. HMM-GMM and HMM-NN with one-state model

This experiment was done to compare HMM-NN hybrid with the more conventional HMM-GMM. The TIMIT database was used in this experiment. The input comprised of MFCC39 features. The number of parameters – Gaussian components in the case of GMM and neurons in hidden layer in case of NN – was being increased until the decrease in PER was negligible (< 0.5 %). Final number of neurons in the hidden layer is 400 and final number of Gaussian mixtures is 256. There is almost no difference in minimal PER obtained from both systems (table 1). Numbers of parameters can be seen in table 2.

system	ins	sub	del	PER
GMM	10.3	22.1	9.6	42.0
NN	9.4	23.1	9.1	41.6

Table 1: *HMM-GMM and HMM-NN with one-state model*

system	parameters
GMM	788736
NN	31200

Table 2: *Numbers of parameters*

4.2. Single frame and multi-frame input with MFCC

Multiple frames of MFCC39 were joined together and formend the input to the neural net. We were looking for the minimal PER, therefore the number of subsequent frames joined together was being increased. Several configurations of the neural net were tested – 300, 400 and 500 neurons in the hidden layer. The best PERs were systematically observed for 400 neurons (table 5).

	TIMIT				NTIMIT			
	ins	sub	del	PER	ins	sub	del	PER
MFCC39	9.4	23.1	9.1	41.6	12.5	31.8	11.3	55.6
MFCC39 5 frames	9.1	21.0	7.4	**37.5**	10.9	28.0	10.2	**49.0**
TRAPS 1 sec	8.3	21.3	8.2	37.9	10.8	28.4	10.4	49.6

Table 3: *MFCC and TRAP on well-matched conditions*

	TIMIT / NTIMIT				NTIMIT / TIMIT			
	ins	sub	del	PER	ins	sub	del	PER
MFCC39	17.0	47.5	16.5	80.9	15.0	38.8	11.6	63.4
MFCC39 5 frames	16.1	49.5	14.5	80.1	14.9	46.4	14.4	75.7
TRAPS 1 sec	15.8	45.0	14.1	**75.0**	11.9	33.0	11.7	**56.6**

Table 4: *MFCC and TRAP with mismatch condition*

frames	1	3	5	9	15
PER [%]	41.6	38.1	**37.5**	37.9	39.5

Table 5: *Effect of using multiframe with MFCC*

4.3. MFCC and TRAP on well-matched conditions

A neural network for MFCC39 has 400 neurons in the hidden layer. In case of TRAPs, all nets have 300 neurons in the hidden layer. Systems were trained and tested on both the clean speech (TIMIT) and on the telephone speech (NTIMIT). As can be seen in table 3, no benefit from using 1 s long TRAP on well-match training and testing condition was found. As described later, an improvement was obtained when the length of TRAP was optimized.

4.4. MFCC and TRAP with mismatch condition

Configurations of all networks are the same as in the previous experiment. The system was trained on TIMIT and tested on NTIMIT at first and then the training and the testing databases were swapped. The TRAP-based system in this case yielded better results in both mismatched conditions (table 4). This experiment also indicates that, when dealing with mismatched data, it may be better to train the system on corrupted speech rather than on the clean one. The degradation in PER resulting from mismatch conditions is the least when using the TRAP technique and the system is trained on NTIMIT and tested on TIMIT. The degradation is 7 % (tables 3 , 4).

4.5. Effect of length of TRAP

The TRAP system, as originally proposed, extracts information from a long temporal context. The length of the context was set to be 1 s in the original system. But this length may depend on the task, on the size of classifiers, and on the amount of the training data. This experiment therefore evaluates the optimal length of the input vector for this task. The numbers of neurons in hidden layers of neural nets are constant – all had 300 neurons, and the TIMIT database is used, therefore the amount of training data is limited. The length of TRAP is being increased from 200 ms to 1 s and the PER is evaluated. As can be seen in picture 2 or in table 6, the optimal length is about 300 ms. It means using 150 ms to the future and 150 ms to the past. Finding such an optimum could mean insufficient training data

and the issue deserves further investigation. However, the fact that shorter input is effective here may have implications in applications where the minimal algorithmic delay is required.

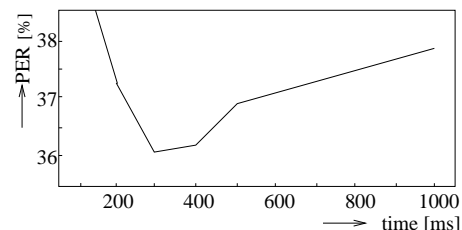| length [ms]	100	200	300	400	600	1000
PER [%]	40.9	37.3	**36.1**	36.2	37.1	37.9

Table 6: *Effect of length of TRAP*

Figure 2: *Effect of length of TRAP*

4.6. Effect of mean and variance normalization

It is known that mean and variance normalization helps when there is a mismatch between training and test condition. However, in well-matched case, the benefits of such normalization are less obvious. To evaluate the effect of the normalization in our system, experiments were performed on both well-matched and on both mismatch conditions (table 7). Significant degradation caused by both normalizations condition can be seen in well-matched condition. In case of mismatch conditions, the mean normalization always helps. The benefit from variance normalization is less clear.

normalization	TIMIT	NTIMIT	T/N	N/T
none	**37.9**	**49.6**	75.0	56.6
mean	40.5	51.8	**73.5**	54.7
mean & variance	42.6	53.2	74.8	**54.1**

Table 7: *Effect of mean and variance normalization on PER*

4.7. TRAP with more than one critical band

Recent results [11, 12] suggest advantage of use of up to three critical band trajectories in individual TRAP probability estimators. In our case this is done by concatenating Hamming windowed 310 ms (31 point) long temporal trajectories from the three adjacent critical bands to form a 93-dimensional input vector to each TRAP probability estimator. The individual three-band time-frequency patches overlap in frequency by two critical bands, thus combining the 1-3,2-4,3-5,...,(N-3)-(N-1),(N-2)-N critical bands. The number of individual TRAP probability estimators in the system is reduced by two since the inputs to the first and the last estimators overlap with their neighbors only at one critical band.

The resulting PER from the three-band TRAP system is **33.7 %**. This is the best result obtained in our experiments so far on the clean TIMIT data and represents more than 10 % relative improvement in PER comparing to the best (i.e. multiframe) baseline system.

5. Potential for merging

When the GMM and neural network classifiers as posterior probability estimators were compared, phoneme confusion matrices were also studied. We were interested if errors caused by each classifier differ so that the PER can be improved by merging of outputs of these two classifiers. We have observed that the amount of complementary information depends mainly on number of parameters in each classifiers and on the training. If the classifiers are not trained well or have fewer parameters (smaller number of mixture components or neurons in the hidden layer), merging can bring great improvement. But there is only a marginal improvement if classifiers have enough parameters and everything is trained well. The confusion matrices are very similar in this case. We observed only about 2 % absolute improvement after merging of two systems, done by summing posteriori probabilities in the logarithmic domain.

6. Conclusion

TRAP perform better on this task than multi-frame MFCC, especially when there is a mismatch between training and testing conditions. The optimal length of temporal pattern in our task is about 300 ms. The mean and variance normalization of temporal patterns degrades the system in well matched conditions. The benefit of mean normalization in mismatch conditions was demonstrated. The variance normalization helped in one mismatch experiment slightly but in the second the phoneme error rate increased, therefore its benefit is questionable.

7. Acknowledgments

8. References

[1] K. Lee and H. Hon, "Speaker-independent phone recognition using hidden Markov models", IEEE Transactions on Acoustics, Speech, and Signal Processing, 37(11):1641-1648, November 1989.

[2] A. Robinson," An application of recurrent nets to phone probability estimation", IEEE Transactions on Neural Networks, vol. 5, No. 3, 1994

[3] Rathinavelu Chengalvarayan and Li Deng, "HMM-Based Speech Recognition Using State-Dependent, Discriminatively Derived Transforms on Mel-Warped DFT Features", IEEE Transactions on Speech and Audio Processing, vol. 5, No. 3, 1997.

[4] Rathinavelu Chengalvarayan and Li Deng, "Use of Generalized Dynamic Feature Parameters for Speech Recognition", IEEE Transactions on Speech and Audio Processing, vol. 5, No. 3, 1997.

[5] S. A. Zahorian, P. L. Silsbee and X. Wang, "Phone Classification with Segmental Features and a Binary-Pair Partitioned Neural Network Classifier", Proc. ICASSP 97, pp. 1011-1014, Munich, Germany, April 1997.

[6] H. Bourlard and N. Morgan. "Connectionist speech recognition: A hybrid approach." Kluwer Academic Publishers, Boston, USA, 1994.

[7] S. Sharma, D. Ellis, S. Karajekar, P. Jain and H. Hermansky, "Feature extraction using non-linear transformation for robust speech recognition on the Aurora database", in Proc. ICASSP 2000, Turkey, 2000.

[8] H. Hermansky and S. Sharma, "Temporal Patterns (TRAPS) in ASR of Noisy Speech", in Proc. ICASSP'99, Phoenix, Arizona, USA, Mar, 1999

[9] The SPRACHcore software packages, *www.icsi.berkeley.edu/ dpwe/projects/sprach/*

[10] HTK toolkit, *htk.eng.cam.ac.uk/*

[11] P. Jain and H. Hermansky, "Beyond a single critical-band in TRAP based ASR", submitted to Eurospeech 2003.

[12] P. Jain : personal comunication.