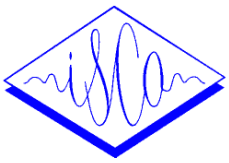
ISCA Archive
http://www.isca-speech.org/archive



3rd European Conference on
Speech Communication and Technology
EUROSPEECH'93
Berlin, Germany, September 19-23, 1993

PHONETIC FEATURES FOR SPELLED LETTER RECOGNITION WITH A TIME DELAY NEURAL NETWORK

Christoph Windheuser and Frédéric Bimbot

Télécom Paris - Dépt. Signal - C.N.R.S. - URA 820
Paris, France

ABSTRACT

In this paper we describe a TDNN based hybrid word recognition system with a novel phoneme representation based on phonetic features. This new representation is more compact than the traditional 1-out-of-N phonetic representation and leads to a smaller network. In different experiments on a spelling letter database we show that the set of phonetic features has to be chosen carefully to achieve good results. We compare the new representation against the standard phoneme representation in the same experiment and show that the phonetic feature representation leads to better recognition results and more stable learning. With the phonetic feature representation we reached a word recognition rate on an independent test set of the spelled letter task of 96.1%.

Keywords: Word recognition, time delay neural networks, phonetic features, hybrid systems.

10.21437/Eurospeech.1993-298

1. INTRODUCTION

One of the most successful speech recognition systems are *hybrid systems*. They consist of a connectionist component for phoneme recognition and a time alignment procedure to match the phonetic scores with a set of word models. A popular approach for the phoneme recognition part is the *Time Delay Neural Network (TDNN)* [6], which can achieve high recognition rates on phoneme recognition. A widely used approach for the time alignment is the *Dynamic Time Warping (DTW)* algorithm [4]. One example of such a hybrid system is the *Multi-State Time Delay Neural Network (MS-TDNN)* [3, 5].

All hybrid systems have in common that their connectionist component uses a 1-out-of-N representation with one output unit per phoneme. During the training of the network for every frame one output unit is trained to one and all other output units are trained to zero. The overwhelming amount of negative training can lead to problems and generally slows down the training speed.

In this paper we will show a novel approach to handle this problem. Instead of using a 1-out-of-N phoneme representation we train a TDNN to recognize *binary phonetic features*. Phonemes are then expressed in terms of these phonetic features by defining for every phoneme which feature is active or inactive.

2. PHONETIC FEATURES

Phonetic feature sets were designed by phoneticians as a taxonomy of the phonemes. Several phonemes can have phonetic features in common and normally a phoneme is described by several active features. So the representation of the phonemes in terms of a set of phonetic features is a *distributed representation*. Of course there exist plenty of arbitrary distributed representations of the phonemes, but in an earlier study [1] it was shown that for a TDNN a representation based on phonetic features is easier to learn than a random representation.

Training a TDNN to learn a distributed phoneme representation based on phonetic features has several advantages. First of all there normally exist fewer features than phonemes which results in a reduction of the number of output units and weights. The representation is more compact and negative and positive training is better balanced. Because different phonemes in such a representation have a smaller euclidean distance when they have some features in common, the distance can be interpreted in a phonetical way here. This phonetically motivated metric on the set of phonemes can help the network to generalize better. By forcing the network to learn a special phonetically motivated representation, we provide some a-priori knowledge to the network, which can lead to a better performance.

3. THE SYSTEM

To demonstrate the usefulness of a phonetic feature representation we ran experiments on the ALPH database from Carnegie Mellon University. It is a spelling letter database of the english alphabet. We used one speaker (jmt), speaking 1000 sentences

of phonetically balanced letters. We splitted them in 500 sentences for training and 500 sentences for testing. The high quality speech was represented to the network in frames of 16 mel-scale filter bank coefficients at a 10 ms frame rate.

Our system consists of a TDNN, which is trained on a frame-based level. The phonetic labels necessary for the training are provided by a HMM system, working in a forced-alignment mode. The TDNN consists of an input layer with 16 units, one hidden layer with 24 units and an output layer. The number of units in the output layer depends on the number of phonetic features. The connections between the input layer and the hidden layer have the delay values -1, 0 and 1 and the connections between the hidden layer and the output layer have the delay values -2, -1, 0, 1 and 2. Every unit of the hidden layer and the output layer have a learnable bias unit. We use online learning without momentum and a learning rate of 1.0. This value seems to be relatively large, but smaller values just slow down the learning. The speech input coefficients are normalized to the interval [0; 1] and the output values of the sigmoid functions of the units are also within this interval.

In the testing mode our system is very similar to the MS-TDNN. We use the DTW algorithm to find the best time alignment between the output of the TDNN and the word models. We test all words (the spelled letters) with known word boundaries. For this the output activations of the output layer are copied into the DTW and matched against all word models. We do not use any penalties, but do not allow to change a state of a model without changing the actual input frame. The local distance is computed in accordance to the euclidean metric.

4. THE EXPERIMENTS

In a first experiment we used the *distinctive feature composition set* from Chomsky and Halle [2]. It is a complete set to describe the english phonemes and contains the following 13 features:

vocalic, consonantal, high, back, low, anterior, coronal, round, tense, voice, continuant, nasal, strident.

Every phoneme is represented with some of these features active (+) and others inactive (-). The phoneme /u/ for example is represented with the active features: *vocalic, high, back* and *round* and all other features inactive. The complete phoneme representation can be found in [2].

The best result we got with this feature set was 86.6% word recognition rate on the test set. The problem with this feature set is that it was not designed for recognition purpose but for defining a system of phonological rules and the features are partly based on diachronic considerations. Figure 1 shows the errors per unit of the output layer accumulated over the whole training set for the last epoch. We see that the network had difficulties to learn the features *vocalic, high* and *tense.*

To circumvent the problem with the feature set from Chomsky and Halle, we designed a new feature set specially for spelled letter recognition. Because not all english phonemes are used for spelling, the set has not to be complete. To make the recog-

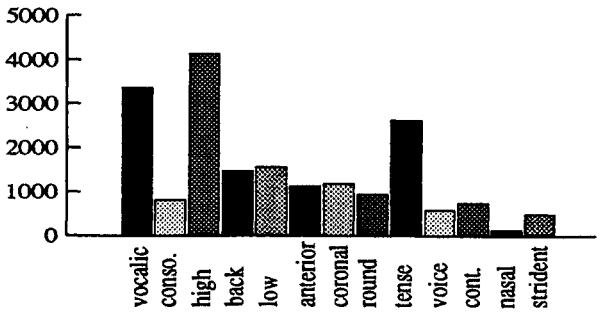


Figure 1: The error per feature accumulated over the 70.000 frames of the training set for the first feature set.

nition of abstract features, like features describing the articulatory position, easier to learn, we defined separate features for vowels and consonants. The new feature set consisted of 11 features:

vocalic, consonantal, high vowel, low vowel, back vowel, front vowel, voiced, front consonant, back consonant, fricative, nasal.

With this new feature set we reached a word recognition rate of 91.2% for the test set. Figure 2 shows the accumulated errors per feature for this feature set.

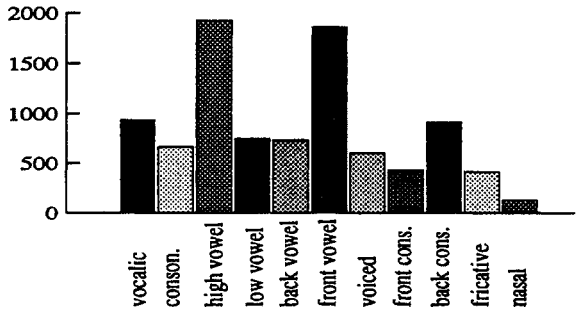


Figure 2: The error per feature accumulated over the 70.000 frames of the training set for the second feature set.

We see that the more abstract features like *high vowel* and *front vowel* cause the biggest errors because they are more difficult to detect from the acoustic signal. But the overall error is quite lower than with the first feature set. It might be confusing that the same features *vocalic* and *consonant* in both feature sets show a different accumulated error. This is due to the fact that they are defined differently in both sets.

To decrease the error per feature and to further increase the recognition rate, we defined a new feature set. This third feature set consists of a greater number of features, which are more specific. This additional redundancy should improve the word recognition performance of the network. The explicit definition of the 18 features of this new feature set is shown in table 1.

With this feature set we improved the word recognition rate of our system to 95.9% on the test set. Figure 3 shows the accu-

	aa	ah	ax	eh	y	iy	ow	uw	l	m	n	jh	r	z	v	w	d	b	s	ch	f	t	k	p
vowel	+	+	+	+	+	+	+	+	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-
closed vowel	-	-	-	-	+	+	-	+	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-
mid vowel	-	+	+	+	-	-	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
open vowel	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
back vowel	-	-	+	-	-	-	+	+	-	-	-	-	-	-	-	+	-	-	-	-	-	-	-	-
central vowel	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
front vowel	-	-	-	+	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
vocalic consonant	-	-	-	-	-	-	-	-	+	+	+	-	+	-	-	-	-	-	-	-	-	-	-	-
liquid	-	-	-	-	-	-	-	-	+	-	-	-	+	-	-	-	-	-	-	-	-	-	-	-
nasal	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	-	-	-	-	-	-	-
consonant	-	-	-	-	-	-	-	-	-	-	-	+	-	+	+	-	+	+	+	+	+	+	+	+
plosive	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	+	+	+
fricative	-	-	-	-	-	-	-	-	-	-	-	+	-	+	+	-	-	-	+	+	+	-	-	-
front consonant	-	-	-	-	-	-	-	-	-	+	-	-	-	-	+	-	-	+	-	-	+	-	-	+
central consonant	-	-	-	-	-	-	-	-	+	-	+	-	-	+	-	-	+	-	+	-	-	+	-	-
back consonant	-	-	-	-	-	-	-	-	-	-	-	+	+	-	-	-	-	-	-	+	-	-	+	-
unvoiced	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	+	+	+	+	+	+
voiced	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	-	-	-	-	-	-

Table 1: Definition of the third feature set.

mulated errors over the training set.

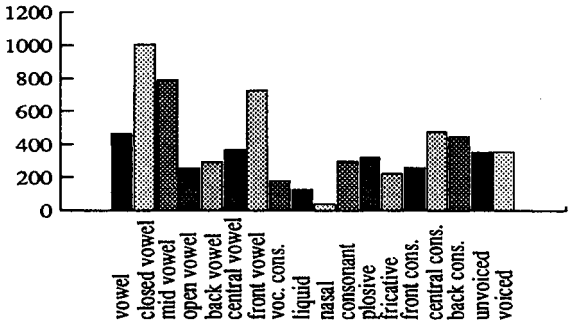


Figure 3: The error per feature accumulated over the 70.000 frames of the training set for the third feature set.

Again most errors occur with the more abstract and articulatory features like *closed vowel, mid vowel* or *front vowel*, but the overall error is much smaller as with the second feature set.

In this experiment with the third feature set we used a more detailed labeling for the diphthongs. We substituted all the labels /ey/ by /eh/ /y/, /ay/ by /aa/ /y/, /ch/ by /t/ /ch/, /jh/ by /d/ /jh/ and /ow/ by /ow/ /w/. This led to a better learning of the phonetic features. The phonetic symbols used here and in the table 1 are in accordance with the DARPA TIMIT alphabet.

5. THE LOCAL DISTANCE IN THE DTW

In the DTW the distances between the activation vector of the TDNN output layer and all the frames of a word model are computed for all input speech frames. In all the experiments reported here we used the euclidean distance to compute this distance. To get a minimal distance the activation of output units of active features should be near one and simultaneously the activation of output units of inactive features should be near zero. So the network not only has to learn to provide a high activation for a output unit when its feature is present in the input, but also to provide a low activation when the feature is not present.

Another possibility to compute the local distance in the DTW is to just take the output units of active features into account. This way the activation of output units with active features is interpreted as *evidence* of the presence of this feature in the actual speech frame. The activation of inactive features does not influence the distance and is not interpreted as evidence that these features are not present.

This new distance between an activation vector of the TDNN output unit and a frame of a word model is computed by adding the activation of output units which features are active in the model for this frame. This distance is also used in the MS-TDNN approach. To avoid normalization of the distance values, the feature set should be designed in a way that for every phoneme the same number of features are active. The third feature set has exactly this property, as we see in table 1.

With this new distance function we improved our results on the spelling letter task to 96.1% word recognition on the test set. Moreover, there is just a small difference to the result on the training set (96.7%), which shows the good generalization ability of the network.

6. COMPARISON

For comparison reasons we also trained our system with the standard 1-out-of-N phoneme representation. All parameters were kept the same except the number of units in the output layer and the targets for learning. Due to the reduced number of output units, the system with the feature representation had 20% less weights (3312 weighs for the system with feature representation and 4338 weights for the system with phoneme representation). Figure 4 shows the word recognition results on the test set for both systems after several epochs of learning. It is clearly visible, that the system with the feature representation learns faster and has less problems with overfitting, due to its reduced number of learnable parameters.

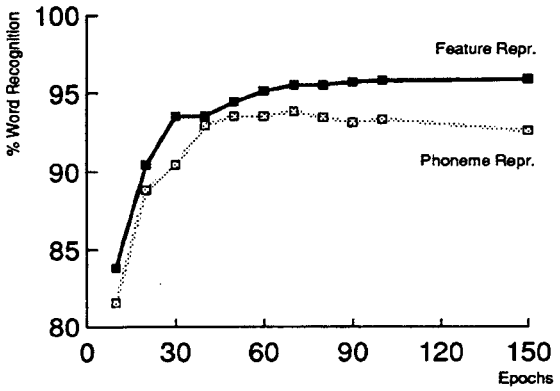


Figure 4: Results on the test set of the ALPH spelling letter database over several training epochs with the third feature representation and with a 1-out-of-N phoneme representation.

7. CONCLUSION

We have shown in the experiments reported here, that adding some a priori knowledge can help a speech recognition system to accomplish its task. We forced the network to learn a specific representation of the phonemes. This resulted in a smaller network with a more compact representation of the phonemes. We showed, that it is possible to reach very high recognition results with the feature set representation. We further showed, that in the same experiment the phonetic feature representation led to better results and a more stable training over the traditional 1-out-of-N phonetic representation. But the set of features has to be chosen carefully. Too abstract features or features which are irrelevant for recognition purpose should be avoided. Some redundancy in the feature set can lead to better recognition results.

In the following table we summarize our results:

System	Word recognition rate
1-out-of-N phoneme representation	**93.8 %**
feature set 1 representation	**86.6 %**
feature set 2 representation	**91.2 %**
feature set 3 representation	**95.9 %**
feature set 3 + non-euclidian distance in the DTW	**96.1 %**

Table 2: Summary of the results of the different systems on the ALPH spelled letter database.

ACKNOWLEDGEMENTS

The authors would like to thank Alex Waibel for providing the ALPH database, Patrick Haffner and Denis Jouvet for a lot of fruitful discussions and reviewing a first version of this paper and Serge Bommer for helping with the experiments.

The authors gratefully acknowledge financial support from the *french Ministère de l'Enseignement Supérieur et de la Recherche* and from *France Télécom CNET Lannion.*

REFERENCES

[1] F. Bimbot, G. Chollet and J.-P. Tubach. "TDNNs for Phonetic Feature Extraction: A Visual Exploration". In *Proc. of the ICASSP*, 1991.

[2] N. Chomsky and M. Halle. "The Sound Pattern of English", *Haper and Row*, 1968.

[3] P. Haffner, M. Franzini and A. Waibel. "Integrating Time Alignment and Connectionist Networks for High Performance Continuous Speech Recognition". In *Proc. of the ICASSP*, 1991.

[4] H. Sakoe and S. Chiba. "Dynamic programming algorithm optimization for spoken word recognition". In *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, pp. 43-49, 1978.

[5] J. Tebelskis. "Performance Through Consistency: Connectionist Large Vocabulary Continuous Speech Recognition". In *Proc. of the ICASSP*, 1993.

[6] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano and K. Lang. "Phoneme Recognition using Time-Delay Neural Networks". In *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 37, pp. 328-339, 1989.