3rd European Conference on Speech, Communication and Technology (EUROSPEECH 93), Berlin, Germany, September 1993, Volume 2, pp. 1481-1484

SPEAKER-INDEPENDENT CONNECTED LETTER RECOGNITION WITH A MULTI-STATE TIME DELAY NEURAL NETWORK

Hermann Hild and Alex Waibel

Universität Karlsruhe, Germany
Carnegie Mellon University, Pittsburgh, USA

ABSTRACT

We present a Multi-State Time Delay Neural Network (MS-TDNN) for speaker-independent, connected letter recognition. Our MS-TDNN achieves 98.5/92.0% word accuracy on speaker dependent/independent English letter tasks[7, 8]. In this paper we will summarize several techniques to improve (a) continuous recognition performance, such as sentence level training, and (b) phonetic modeling, such as network architectures with "internal speaker models", allowing for "tuning-in" to new speakers. We also present results on our large and still growing new German Letter data base, containing over 40.000 letters continuously spelled by 55 speakers.

Keywords: Spelled Letter Recognition, Speaker-Independence, MS-TDNN

1. INTRODUCTION

The recognition of spelled strings of letters is essential for all application involving special vocabularies, such as names or addresses. Despite its small vocabulary, the task is quite difficult because the English or German letters are easily confused. Even humans often need further inquiry to distinguish between the similar sounds of (for example) the letters **M** and **N**, or **D** and **T**. Throughout this text, we will use the terms "letter" and "word" interchangeable. The term "sentence" refers to a string of letters.

The Baseline MS-TDNN [5, 8] integrates the time-shift invariant architecture of a TDNN [12] and a nonlinear time alignment procedure (DTW) into a word-level classifier. Figure 1 shows the MS-TDNN in the process of recognizing the excerpted word **B**, represented by 16 melscale FFT coefficients at a 10 msec frame rate. The first three layers constitute a standard TDNN, which uses sliding windows with time delayed connections to compute a score for each phoneme for every frame, these are the activations in the "Phoneme Layer". Each word to be recognized is modeled by a sequence of phonemes. In the "DTW (Dynamic Time Warping) Layer", an optimal alignment path, i.e. the path with the highest accumulative phoneme scores is found for each word, the activations along these paths are then collected in the word output units. 15 (25, 50) rows of hidden units were used for speaker-dependent (-independent) experiments, corresponding to ca. 6000 (10000, 20000) trainable parameters, i.e. network weights.

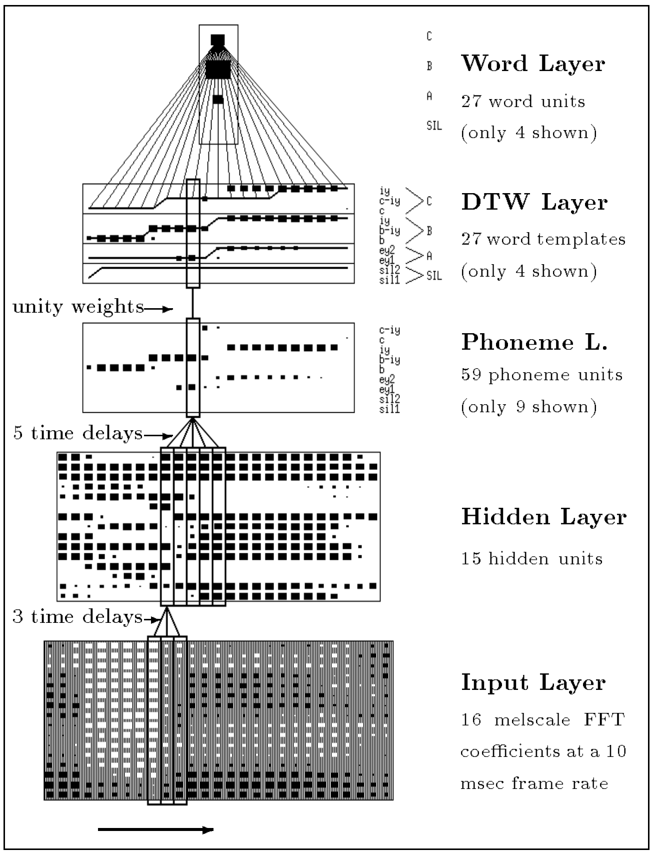


Figure 1: The MS-TDNN recognizing the excerpted word **B**. Only the activations for the words **SIL**, **A**, **B**, and **C** are shown.

Training starts with "bootstrapping", during which only the front-end TDNN is trained as a frame-by-frame phoneme classifier, with phoneme boundaries fixed as given in the training data. In a second phase, training is extended to the word level, where phoneme boundaries within the given word boundaries are freely aligned in the

DTW Layer. Instead of phonemes, the output are now words, and error derivatives are backpropagated from the word units through the alignment paths and the front-end TDNN.

The choice of sensible objective functions is of great importance. For training on the phoneme level, there is an output vector $Y = (y_1, \ldots, y_n)$ and a corresponding target vector $T = (t_1, \ldots, t_n)$ for each frame in time. T represents the correct phoneme j in a "1-out-of-n" coding, i.e. $t_i = \delta_{ij}$. Standard *Mean Square Error* ($MSE = \sum_{i=1}^{n} (y_i - t_i)^2$) is problematic for "1-out-of-n" codings for large n ($n > 50$ in our case); consider for example that for a target $(1, 0, \ldots, 0)$, the output $(0.0, \ldots, 0.0)$ has only half the error than the more desirable output $(1.0, 0.2, \ldots, 0.2)$. This problem is avoided by

$$E_{McClelland}(T, Y) = \sum_{i=1}^{n} log(1 - (y_i - t_i)^2)$$

which (like cross entropy) punishes "outliers" with an error approaching infinity for $|t_i - y_i|$ approaching 1.0.

For the word level training, we have achieved best results with an objective function similar to the "Classification Figure of Merit" (CFM) [4], which tries to maximize the distance $d = y_c - y_{hi}$ between the correct score y_c and the highest incorrect score y_{hi} instead of using absolute targets 1.0 and 0.0 for correct and incorrect word units:

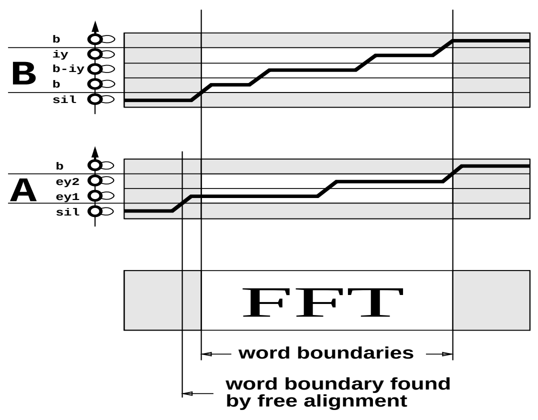
$$E_{CFM}(T, Y) = f(y_c - y_{hi}) = f(d) = (1 - d)^2$$

The philosophy here is not to "touch" any output unit not directly related to correct classification. We found it even useful to backpropagate error only in the case of a wrong or too narrow classification, i.e. if

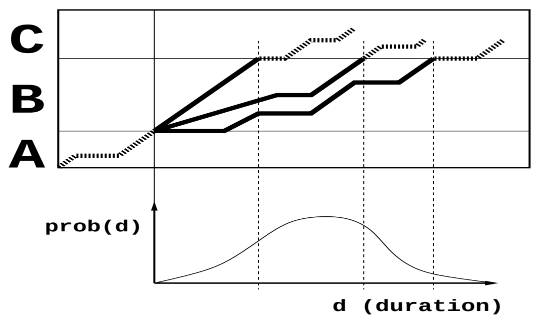
$$y_c - y_{hi} < \delta_{safety\_margin}$$

2. IMPROVING CONTINUOUS RECOGNITION

Training Across Word Boundaries. A proper treatment of word boundaries is especially important for a short word vocabulary, since most phones are at word boundaries. While the phoneme boundaries within a word are freely aligned by the DTW during "word level training", the word boundaries are fixed and might be error prone or suboptimal. By extending the alignment one phoneme to the left (last phoneme of previous word) and the right (first phoneme of next word), the word boundaries can be optimally adjusted in the same way as the phoneme boundaries within a word. Figure 2(a) shows an example in which the word to recognize is surrounded by a silence and a **B**, thus the left and right context (for all words to be recognized) are the phonemes **sil** and **b**, respectively. The gray shaded area indicates the extension necessary to the DTW alignment. It is shown how a new boundary for the beginning of the word **A** is found.



(a) alignment across word boundaries



(b) duration dependent word penalties

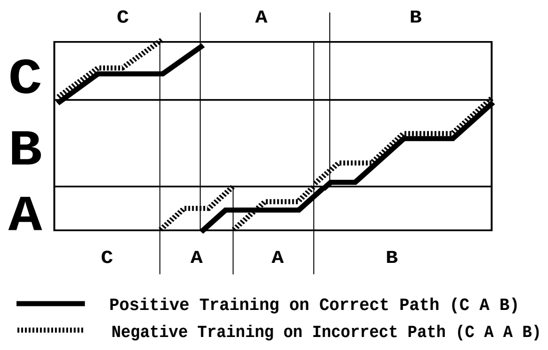

C
A
B
C
B
A
C
A
A
B
Positive Training on Correct Path (C A B)
Negative Training on Incorrect Path (C A A B)
(c) training on the sentence level

Figure 2: Various techniques to improve sentence level recognition performance.

Word Duration Dependent Penalizing of Insertion and Deletion Errors. In continuous recognition mode, instead of looking at word units the well-known "One Stage DTW" algorithm [11] is used to find an optimal path through an unspecified sequence of words. The short and easily confused English letters cause many word insertion and deletion errors, such as "**T E**" vs. "**T**" or "**O**" vs. "**O O**" , therefore proper duration modeling is essential. As suggested in [6], minimum phoneme duration can be enforced by "state duplication". In addition, we are modeling a duration (d) and word (w)

dependent penalty $Pen_w(d) = log(k + prob_w(d))$, where the pdf $prob_w(d)$ is approximated from the training data and k is a small constant to avoid zero probabilities. $Pen_w(d)$ is added to the accumulated score AS of the search path, $AS = AS + \lambda_w * Pen_w(d)$, whenever a word boundary is crossed, as indicated in figure 2(b). The ratio λ_w, which determines the degree of influence of the duration penalty, is another important degree of freedom. There is no straightforward mathematically exact way to compute the effect of a change of the "weight" λ_w to the insertion and deletion rate. Our approach is a (pseudo) gradient descent, which changes λ_w proportional to $E(w) = (\#ins_w - \#del_w)/\#w$, i.e. we are trying to maximize the relative balance of insertion and deletion errors.

Error Backpropagation at the Sentence Level. Usually the MS-TDNN is trained to classify excerpted words, but evaluated on continuously spoken sentences. We propose a simple but effective method to extend training on the sentence level. Figure 2(c) shows the alignment path of the sentence "**C A B**" , in which a typical error, the insertion of an **A**, occurred. In a forced alignment mode (i.e. the correct sequence of words is enforced), positive training is applied along the correct path, while the units along the incorrect path receive negative training. Note that the effect of positive and negative training is neutralized where the paths are identical, only differing parts receive non-zero error backpropagation.

3. INTERNAL SPEAKER MODELS

The idea of **"Internal Speaker Models" (ISMs)** is to have submodules in a network, each of which is specialized on one particular speaker, or a group of speakers, found by clustering, or simply male/female speakers. The number of specialized parameters can vary from simple speaker-specific bias connections to speaker-specific subnets, and finally entire speaker-specific TDNNs. When an unknown speaker is presented, somehow one or a (normalized) mixture of appropriate submodule(s) has to be selected. This is done by "ISM selection units" (ISM-SUs, one for each ISM), which influence the contribution of each ISM network . We explored two different mechanisms (figure 3): **(a)** An additional "speaker identification net" is trained to control the internal speaker models, i. e. the activations of the ISM-SUs are computed by this net each time before an utterance is recognized, and **(b)** a "tuning-in" process, in which a small set of speech samples from an unknown speaker is used to adapt the selection of the speaker-specific parameters for this speaker. "Tuning-in" is relatively straightforward for labeled samples, the "mixture parameters", i. e. the activations of the ISM-SUs, are found via gradient descent (error is backpropagated into the ISM-SUs, all other weights are frozen), where the objective function is to maximize the performance on the adaptation data. The so found

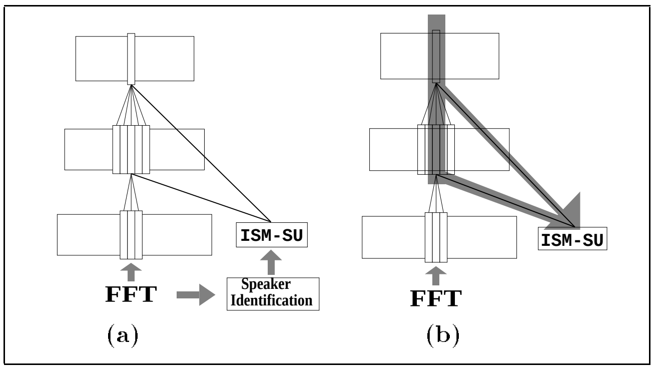


Figure 3: Two methods to adjust the internal speaker models, shown for the BIASED architecture: "Speaker Identification" (a), and "Tuning in" (b).

mixture of ISMs is then used for recognition on the entire rest of the test data. "Tuning-in" can also be applied in an unsupervised fashion [1], in which case "phantom targets" (derived from the actual net output) are used. These experiments and their results are described in more details in [8].

4. EXPERIMENTAL RESULTS

4.1. ENGLISH LETTERS

Our MS-TDNN achieved excellent performance on both speaker dependent and independent tasks. For **speaker dependent** testing, we used the CMU "Alph-Data", with 1000 sentences (i. e. continuously spelled strings of letters in our context) from each of 3 male and 3 female speakers. **Speaker-independent** performance was measured on the DARPA Resource Management Spell-mode data, consisting of a total of 1680 spelled words from 120 speakers. Table 1 indicates the usage of training and test sets.

Speaker Dependent (CMU Alph Data) 600/3000 train, 400/2000 test sentences/words		
speaker	SPHINX[5]	our MS-TDNN
mjmt	96.0	**98.5**
mdbs	83.9	**91.1**
maem	–	**94.6**
fcaw	–	**98.8**
flgt	–	**86.9**
fee	–	**91.0**

Speaker Independent (Res. Man. Spell-mode) 109/11000 train, 11/900 test speaker/words			
SPHINX[9]		our MS-TDNN	
	+ Senone		gender specific
88.7	90.4	**90.8**	**92.0**

Table 1: Word accuracy (in % on the test set) on speaker dependent/independent connected letter tasks.

4.2. GERMAN LETTERS

We are in the process of creating a large data base of German spelled letters. At this time, more than 40.000 letters from 55 speakers (table 2) were collected and labeled. Volunteers are asked to spell a set of 50 to 150 sentences in a natural manner, without artificial pauses between letters. Each individual spells a different set, consisting of three categories: proper names, drawn randomly from a large list of 100.000 names, some random city names and some pseudo-random letter sequences. The latter subset is designed to increase the percentage of the less frequent words, such as **Q** or **X**, to make sure there is a reasonable amount of training data for all letters. For example, after adding the pseudo random sequences, the ratio of **Q** to **E** increases from 3 : 1000 to 75 : 1000 .

To obtain word and phoneme boundaries, the data were first labeled by running the JANUS LVQ recognizer [13] in a forced alignment mode. After a initial training with these labels, the MS-TDNN was used to relabel the data. This procedure almost halved the error rate on successive training runs.

	male		female		all	
	Spr	letter	Spr	letter	Spr	letter
train	33	24752	10	10071	**43**	**34823**
test	9	6341	3	1865	**12**	**8206**
all	42	31093	13	11936	55	43029

Table 2: The German Spell Data Base.

The Vocabulary. In addition to "Silence" and the 26 letters of the English alphabet, the German alphabet provides the 3 modified vowels ("Umlaute") **Ä**, **Ö**, and **Ü**, as well as **ß**. Unfortunately, there are several possible ways to spell **ß**, besides the official version "eszett", the pronunciations "scharf-S", "scharfes-S", or in some dialects even "Dreierles-S" are also used. Since contained by many proper names, the hyphen (**-**) was also included into the vocabulary, with three different possible pronunciations ("Strich", "Bindestrich", and "Gedankenstrich"). With the above additions, the German spelling task sums up to a vocabulary size of 37 words.

Interestingly, the German letters have much less problems with insertion and deletion errors than the English letters. When the English letters are tested on sentence level, the initial combined insertion and deletion error rate is greater 10%, and only after learning word entrance penalties and sentence level training (as described in section 2), it is reduced to roughly 4%. German letters have an initial combined insertion and deletion rate of only ca. 3%, probably due to the more "staccato-like German uttering", and to a more implicit modeling of glottal stops. Table 3 summarizes first results on the German letter data base. Further improvements are expected with more training speakers and by applying the speaker modeling techniques described in section 3.

Speaker-Independent German Spell Task 43/34823 train, 12/8206 test speaker/words				
Net Size	excerpted words		continuous	
(hidden layer)	train	test	train	test
25 units	97.6	92.5	94.3	88.1
50 units	98.4	93.8	96.4	**90.3**

Table 3: Word accuracy in % on German Spelling.

Acknowledgements

The authors gratefully acknowledge support by the NSF, ARPA, and Siemens AG. Creating the large German data base would have been impossible without the tireless work of Thomas Schaaf, Gudrun Socher, Henrike Triess, Monika Woszczyna and many others.

References

[1] J.S. Bridle and S. J. Cox. RecNorm: Simultaneous Normalisation and Classification applied to Speech Recognition. In *Adv. in Neural Network Information Processing Systems 4* , Morgan Kaufmann, 1992.

[2] M. Fanty and R. Cole. Spoken letter recognition. In *Adv. in Neural Network Information Processing Systems 3* , Morgan Kaufmann, 1991.

[3] J. Hampshire and A. Waibel. The Meta-Pi Network: Connectionist Rapid Adaptation for High-Performance Multi-Speaker Phoneme Recognition. In *Proc. Intern. Conf. on Acoustics, Speech and Signal Processing,* IEEE, 1990

[4] J. Hampshire and A. Waibel. A Novel Objective Function for Improved Phoneme Recogn. Using Time Delay Neural Networks. *IEEE Trans. on Neural Networks*, June 1990.

[5] P. Haffner, M. Franzini, and A. Waibel. Integrating Time Alignment and Neural Networks for High Performance Continuous Speech Recognition. In *Proc. Intern. Conf. on Acoustics, Speech and Signal Processing,* IEEE, 1991

[6] P. Haffner and A. Waibel. Multi-State Time Delay Neural Networks for Continuous Speech Recogn. In *Adv. in Neural Network Inf. Proc. Syst. 4*, M. Kaufmann, 1992.

[7] H. Hild and A. Waibel. Connected Letter Recogn. with a Multi-State Time Delay Neural Network. In *Adv. in Neural Network Inf. Proc. Syst. 5.* M. Kaufmann, 1993.

[8] H. Hild and A. Waibel. Multi-Speaker/Speaker-Independent Architectures for the Multi-State Time Delay Neural Network. In *Proc. Intern. Conference on Acoustics, Speech, and Signal Processing.* IEEE, 1993.

[9] M.Y. Hwang and X. Huang. Subphonetic Modeling with Markov States - Senone. In *Proc. Intern. Conference on Acoustics, Speech and Signal Processing,* IEEE, 1992

[10] R.D.T. Jansen, M. Fanty, and R. A. Cole. Speaker-Independent Phonetic Classification in Continuous Engl. Letters. In *Proc. IJCNN 90, Wash. DC.*, July 1990.

[11] H. Ney. The Use of a One-Stage Dynamic Progr. Algorithm for Connected Word Recognition. In *IEEE Trans. on Acoustics, Speech, and Signal Proc.*, April 1984.

[12] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. Lang. Phoneme Recognition Using Time-Delay Neural Networks. *IEEE, Transactions on Acoustics, Speech and Signal Processing,* March 1989.

[13] M. Woszczyna,A. Waibel et al. Recent Advances in JANUS: A Speech Transl. System. *EUROSPEECH '93*