

Figure 5: Recognition results with respect to the dictionary size if the $N = 1 \dots 10$ best words are counted as correct.

incorrect output of the recognizer shows that we can expect further improvements of the word recognition rate by using language models for the recognition of sentences.

5 Conclusions

In this paper we have presented the **NPen⁺⁺** system a connectionist recognizer for writer independent on-line cursive handwriting recognition. This system combines a robust input representation, which preserves the dynamic writing information, with a neural network integrating recognition and segmentation in a single architecture. This architecture has been shown to be suitable for handling temporal sequences as well as for handling different kinds of input. The system was evaluated on different dictionary sizes, showing recognition rates from 98.0% for a 1,000 word dictionary to 82.9% for the 100,000 word dictionary. The largest dictionary used in the evaluation was the German dictionary for a pattern recognition time of 1.5 seconds. These results are especially good since they were achieved with a simple architecture. Other systems (e.g. [4]). Our system has proved to be robust on a large dictionary. Though the system works on different dictionaries, the recognition rate and much more depend on the length of the input.

full
y-

3.3 Training algorithm

During training the goal is to determine a set of parameters θ that will maximize the posterior probability $p(w|\mathbf{x}_0^T, \theta)$ for all training input sequences. But in order to make that maximization computationally feasible even for a large dictionary system we had to simplify that maximum posteriori approach to maximum likelihood training procedure that maximizes $p(w|\theta)$ for all words instead.

First step of our maximum likelihood training is to adapt the recognizer using a subset of approximately 500 words of the training set that were chosen from the database: characters with the character boundaries to adapt a mixture of word layer correctly. After trained on this data, the recognizer is used on a set of unlabeled training data. The training set is processed by the recognizer and the target word unit sequence is determined automatically.

Then, in the second step, the recognizer is trained on both data sets to improve the recognizer.

Its

erent writer inde-
g from 1,000
in the dic-
er case let-
y 5,700
y 80

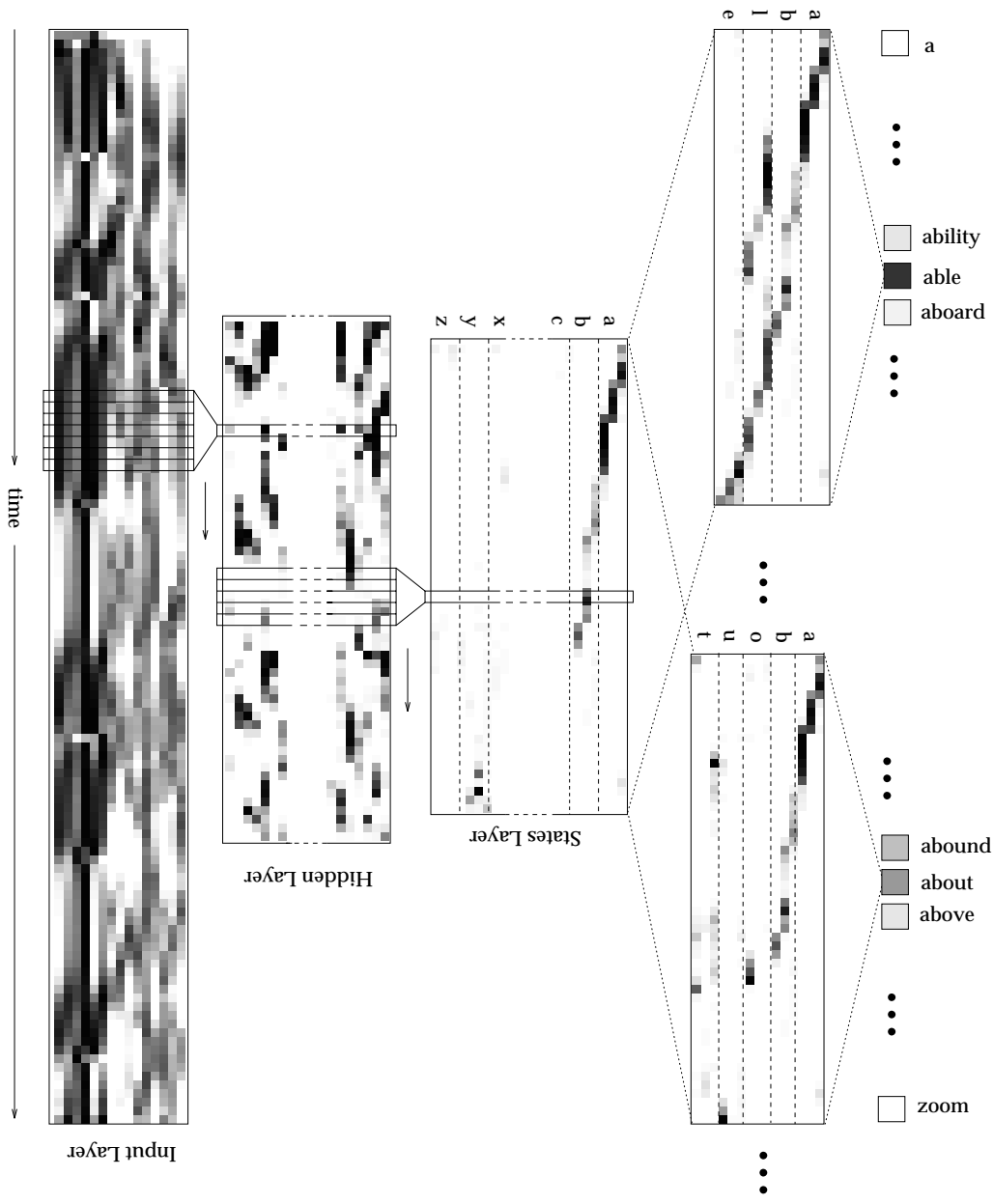


Figure 3: The Multi-State TDNN architecture, consisting of a 3-layer TDNN to estimate the a posteriori probabilities of the character states combined with word units, whose scores are derived from the word models by a posteriori approximation of the likelihoods.

windows in each layer. In the current implementation, the likelihoods of the feature vector system a TDNN with 15 input states are given by the word model ψ , i. e. $\log \mathcal{P}(x_0^T | \psi)$ is an layer, and 78 state output probabilities are derived.

time delays in the input layer and 5 time den layer.
 normalized output of the states 1 given by the word model ψ , i. e. $\log \mathcal{P}(q_t | q_{t-1}, \psi)$
 estimate of the probabilities of the states 1 given by the word model ψ , i. e. $\log \mathcal{P}(x_0^T | \psi)$
 input window $x_{t-d}^{t+d} = x_{t-d} \dots x_{t+d}$
 . e. $q_0^T = \max_{t=1}^T \log \frac{\mathcal{P}(q_t | x_{t-d}^{t+d})}{\mathcal{P}(q_t)} + \log \mathcal{P}(q_t | q_{t-1}, \psi)$.

$\approx \sum_k \frac{\exp(\eta_k(t))}{\exp(\eta_k(t))}$ Here, the maximums over all possible sequences of states $q_0^T = q_1 \dots q_T$ given a word model, $\mathcal{P}(q_t | x_{t-d}^{t+d})$ the weighted states outputs of the states layer as defined in (1) used on these states is the prior probability of observing a state defined to be $\mathcal{P}(q_t | q_{t-1}, \psi)$ on the training data.

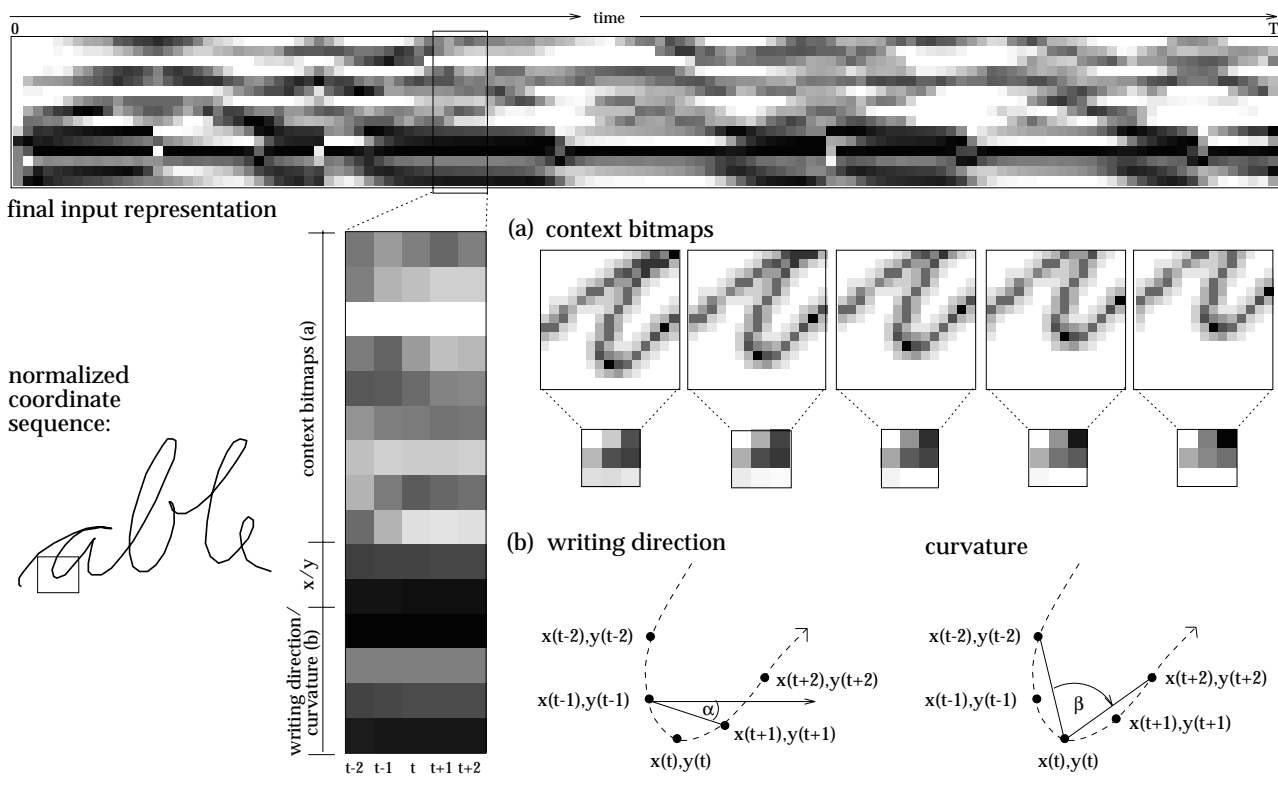


Figure 2: Feature extraction for the normalized word “able”. The final input representation is calculated by calculating a 15-dimensional feature vector for each data point, which includes information about the curvature of the stroke.

recognition task
p

on the LCD tablet or digitizer [10]. The system is designed to make heavy use of this temporal information.

NPen⁺⁺ (Figure 1) combines a neural network recognizer, which was originally proposed for continuous speech recognition tasks [7, 8], with learning techniques, which are used to improve the performance of the recognizer.

NPen⁺⁺: A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System

Stefan Manke, Michael Finke, and Alex Waibel

University of Karlsruhe

Computer Science Department

D-76128 Karlsruhe, Germany

Carnegie Mellon University

School

Abstract

In this paper we describe the NPen⁺⁺ system for writer independent on-line handwriting recognition. This recognizer needs no training for a particular writer and can recognize any common writing style (cursive, hand-printed, or a mixture of both). The neural network architecture, which was originally proposed for continuous speech recognition tasks, and the preprocessing techniques of NPen⁺⁺ are designed to make heavy use of the dynamic writing information, i.e. the temporal sequence of data points recorded on a LCD tablet or digitizer. We present results on the writer independent recognition of isolated words. Tested on different dictionary sizes from 100 to 100,000 words, recognition rates range from 82.9% for the 1,000 word dictionary to 98.9% for the 100,000 word dictionary. No language models are used.

1 Introduction

The success