

Enhancing Training Data for Handwriting Recognition of Whiteboard Notes with Samples from a Different Database

Marcus Liwicki and Horst Bunke
Department of Computer Science, University of Bern
Neubrückestrasse 10, CH-3012 Bern, Switzerland
{liwicki, bunke}@iam.unibe.ch

Abstract

Recognition of unconstrained handwritten text is still a challenge. In this paper we consider a new problem, which is the recognition of notes written on a whiteboard. Our recognizer is based on Hidden Markov Models (HMMs). As it is difficult to acquire sufficient amounts of training data for the HMMs we propose two strategies for enlarging the training set. Both strategies are based on an existing database of off-line handwritten text, which includes handwriting samples different from whiteboard data. The two proposed strategies are MAP adaptation and merging of training sets. With these methods we can achieve improvements of the word recognition rate of up to 5.7%.

1. Introduction

In this paper we describe continuation of our research on a novel handwriting recognition task, which is the recognition of text written on a whiteboard. Our recognition system for this task has been introduced in [8], where a writer independent handwritten sentence recognizer based on HMMs was presented. The performance of this recognizer was only about 64.27% on the word level. The main reason for the low performance is that the number of writers in the training set is very small. The data set of all available whiteboard recordings (training, test and validation set) consists of only about 6,000 words rendered by a total of 20 writers. We expect to get a better recognition performance if we enlarge this data set. However, it is rather difficult to significantly enlarge the existing database, because the whiteboard is not portable and can be used by only a single writer at a time. For this reason we propose another approach in this paper, where we use data from a large existing database of off-line handwritten sentences [10] to augment the training set.

Although the data output by the sensing device is in the on-line format, we use an HMM-based off-line recog-

nizer in the work described in this paper. Our motivation is twofold. First, on-line data can be easily converted into the off-line format and secondly, we do have an off-line recognizer at our disposal that was developed in the context of previous work [9]. Eventually we plan to combine the existing off-line recognizer with an on-line recognizer to be developed in the near future. From such a combination, an improved recognition performance can be expected [12, 14].

We have investigated two different approaches to combining the training set originally used in [8] with a subset of the IAM-Database [10]. First, an HMM-recognizer trained on the IAM-Database is taken and adapted with the Maximum A Posteriori (MAP) estimation method [4]. Adaptation methods are usually used for writer adaptation in writer dependent systems [13], but have been successfully applied in writer independent tasks as well [1]. MAP estimation has been chosen for adaptation because it produced the best recognition results in [13] on adaptation sets of larger size. In the second approach the HMM-recognizer is trained on the union of the selected IAM-Database subset and the whiteboard data. Thus a significant extension of the training set, when compared to [8], is achieved.

To study the effect of training set expansion in a broader context, we have also applied some optimization strategies [5, 16] to the basic HMM recognizer. The motivation was to find out if enlarging the size of the training set leads to a lower error rate even for highly optimized systems. As will be described in Section 5 in greater detail, for both the unoptimized and the optimized version of the system significant improvements of the recognition rate were achieved by the proposed training set expansion strategies.

The rest of the paper is organized as follows. Section 2 gives an overview of the recognition system for whiteboard notes. Section 3 briefly describes the basic recognition system and some optimization steps. In Section 4 the main ideas for enhancing the training data are presented. Experiments and results are given in Section 5, and finally Section 6 draws some conclusions and gives an outlook for future work.



Figure 1. Illustration of the recording; note the data acquisition device in the left upper corner

2. System overview

The eBeam¹ interface is used for recording the handwriting of a user. It allows us to write on a whiteboard with a normal pen in a special casing, which sends infrared signals to a triangular receiver mounted in one of the corners of the whiteboard. The acquisition interface outputs a sequence of (x,y)-coordinates representing the location of the tip of the pen together with a time stamp for each location. An illustration of the data acquisition process is shown in Fig. 1.

The system described in this paper consists of six main modules (see Fig. 2): the on-line preprocessing, where noise in the raw data is reduced; the transformation, where the on-line data is transformed into off-line format (this allows us to apply the recognizer described in [9] to the considered problem); the off-line preprocessing, where various normalization steps take place; the feature extraction, where the normalized image is transformed into a sequence of feature vectors; the recognition, where the HMM-based classifier generates a list of word sequences, which are arranged in a word lattice; and the post-processing, where a statistical language model is applied to the word lattice to improve the results. A detailed description of the original version of the system is given in [8].

3. The recognizer

The basic recognizer is a Hidden Markov Model (HMM) based cursive handwriting recognizer similar to the one de-

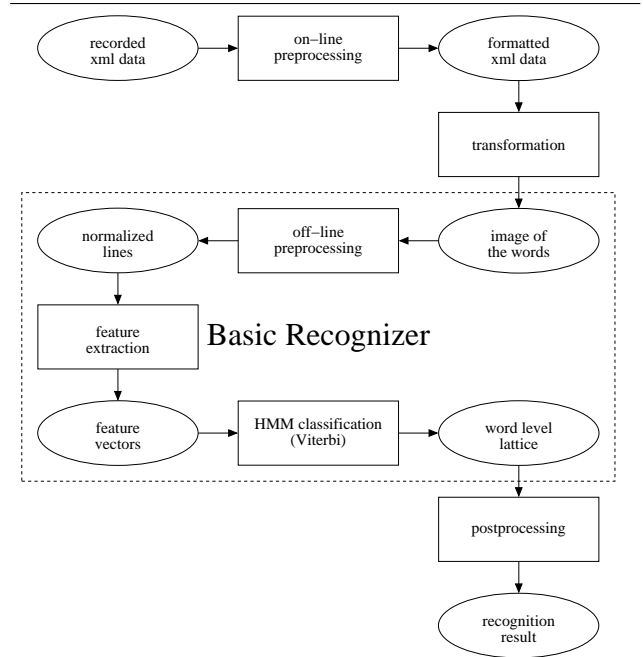


Figure 2. Recognition system overview

scribed in [9]. For the purpose of completeness we include a brief description here. For any further details the reader is referred to [9].

The basic recognizer takes, as an input unit, the image of a complete text line, which is first normalized with respect to skew, slant, writing width and baseline location. Normalization of the baseline location means that the body of the text line (the part which is located between the upper and lower baselines), the ascender part (located above the upper baseline), and the descender part (below the lower baseline) will be vertically scaled to a predefined size each. Writing width normalization is performed by a horizontal scaling operation, and its purpose is to scale the characters so that they have a predefined average width.

To extract the feature vectors from the normalized images, a sliding window approach is used. The width of the window is one pixel and nine geometrical features are computed at each window position. Thus an input text line is converted into a sequence of feature vectors in a 9-dimensional feature space.

An HMM is built for each of the 58 characters in the character set, which includes all small and capital letters and some other special characters, e.g. punctuation marks. In all HMMs the linear topology is used, i.e. there are only two transitions per state, one to itself and one to the next state. In the emitting states, the observation probability distributions are estimated by mixtures of Gaussian components. The character models are concatenated to represent words and sequences of words. For training, the Baum-Welch al-

¹ eBeam System by Luidia, Inc. - www.e-Beam.com

gorithm [2] is applied. In the recognition phase, the Viterbi algorithm [3] is used to find the most probable word sequence. Note that the difficult task of explicitly segmenting a line of text into isolated words is avoided, and the segmentation is obtained as a byproduct of the Viterbi decoding applied in the recognition phase. The output of the recognizer is a sequence of words. In the experiments described in Section 5, the recognition rate will always be measured on the word level.

In [5] it is pointed out that the number of Gaussians and training iterations have an effect on the recognition results of an HMM recognizer. Often the optimal value increases with the amount of training data because more variations are encountered. The system described in this paper has been trained with up to 30 Gaussian components and the classifier that performed best on a validation set has been taken as the final one in each of the experiments described in Section 5.

Another optimization step proposed in [16] is the inclusion of a language model, which corresponds to the post-processing step illustrated in Fig. 2. Since the system described in this paper is performing handwritten text recognition on text lines and not only on single words, it is in fact reasonable to integrate a statistical language model. For further details we refer to [16].

4. Enhancing the training data

Since the off-line images generated from the whiteboard data are similar to the images of the IAM-Database, the IAM-Database can be used to enhance the small dataset of whiteboard recordings. Two strategies for enhancing the dataset are described in this paper. The first strategy is recognizer adaptation. Here we adapt a recognizer that was trained on the IAM-Database to the whiteboard training data. Under the second strategy we use a mixture of data from the IAM-Database and the whiteboard data collection to train the recognizer. We will describe the main ideas of these two strategies in this section.

HMM adaptation is a method to adjust the model parameters θ of a given background model (the HMMs trained on the IAM-Database in our case) to the parameters θ_{ad} of the adaptation set of observations O (the training set of the whiteboard data). The aim is to find the vector θ_{ad} which maximizes the *posterior* distribution $p(\theta_{ad}|O)$:

$$\theta_{ad} = \underset{\theta}{\operatorname{argmax}} (p(\theta|O)) \quad (1)$$

Using Bayes theorem $p(\theta|O)$ can be written as follows:

$$p(\theta|O) = \frac{p(O|\theta)p(\theta)}{p(O)} \quad (2)$$

where $p(O|\theta)$ is the likelihood of the HMM with parameter set θ and $p(\theta)$ is the *prior* distribution of the param-

eters. When $p(\theta) = c$, i.e. when the *prior* distribution does not give any information about how θ is likely to be, Maximum Likelihood Linear Regression (MLLR [7]) can be performed. If the prior distribution is informative, i.e. $p(\theta)$ is not a constant, the adapted parameters can be found by solving the equation

$$\frac{\partial}{\partial \theta} (p(O|\theta)p(\theta)) = 0 \quad (3)$$

This minimizes the Bayes risk over the adaptation set and can be done with Maximum A Posteriori (MAP) estimation, which is also called Bayesian Adaptation. As described in [13], it is feasible to adopt only the Gaussian means μ_{jm} (where m refers to the actual state and j is the index of the considered mixture in state m) of the parameters θ of each HMM. The use of conjugate priors then results in a simple adaptation formula [13]:

$$\hat{\mu}_{jm} = \frac{N_{jm}}{N_{jm} + \tau} \bar{\mu}_{jm} + \frac{\tau}{N_{jm} + \tau} \mu_{jm} \quad (4)$$

where $\hat{\mu}_{jm}$ is the new and $\bar{\mu}_{jm}$ the old mean of the adaptation data, μ_{jm} is the mean of the background model, and N_{jm} is the sum of the probabilities of each observation in the adaptation set being emitted by the corresponding Gaussian. After each iteration the values of $\hat{\mu}_{jm}$ are used in the Gaussians, which leads to new values of $\bar{\mu}_{jm}$ and N_{jm} in Eq. (4). This procedure is repeated until the change in the parameters falls below a predefined threshold. The parameter τ weights the influence of the background model on the adaptation data. Whereas it has been set empirically in [13] it is optimized on a validation set in this paper. The MAP estimation performs better if more adaptation data are available because it adapts each Gaussian separately.

Our second approach to training set enhancement uses data from both the IAM-Database and the whiteboard data collection. The data of the IAM-Database and the training set of the whiteboard data are combined into one large training set. This set is then used for training the HMM recognition system. The validation set for optimizing the parameters is taken only from the whiteboard data. Thus, in contrast to the adaptation, the training of the recognizer is optimized on the whiteboard data. This strategy consumes more time than the adaptation method because the training is performed on a larger dataset.

5. Experiments and results

The overall amount of the recorded whiteboard data is 6,204 words in 1,258 lines from 20 different writers. Each writer wrote approximately the same number of words. The data set was randomly divided into five disjoint sets of approximately equal size (sets s_0, \dots, s_4). On these sets, 5-fold cross validation was performed in the following way

(combinations c_0, \dots, c_4). For $i = 0, \dots, 4$, sets $s_{i\oplus 2}$, $s_{i\oplus 3}$ and $s_{i\oplus 4}$ were taken for adapting (first method) or training (second method) the recognizer, set $s_{i\oplus 1}$ was used as a validation set, i.e. for optimizing the parameters in the optimization steps, and set s_i was used as a test set for measuring the system performance. Each of the sets consists of the data of four writers and no writer appears in more than one set. Consequently, writer-independent recognition experiments were conducted. The word-dictionary includes exactly those 2,337 words that occur in the union of all of the five sets. The language model was generated from the LOB-Corpus [6], which contains 500 printed texts of about 2,000 words each.

The basic recognizer for the whiteboard data has been trained with up to eight Gaussians and four iterations for each Gaussian increment. The average word recognition rate of this recognizer is 59.54% on the five validation sets and 59.59% on the five test sets. Note that this is the baseline experiment, where only whiteboard data is used for training. By integrating a language model as described in Section 3 the recognition rate could be increased to 65.56% on the validation sets and to 64.27% on the test sets.

The IAM-Database includes over 1,500 scanned forms of texts written on normal paper by more than 650 writers. For our experiments a subset containing about 18,000 words in 1,993 text lines produced from 400 different writers was taken. The background model for the adaptation was trained on the data of 320 writers from the IAM-Database. For this recognizer the number of Gaussians were optimized on the validation set consisting of the remaining 80 writers. The performance of this recognizer is 54.94% on the whiteboard data (without using a language model).

As described in Section 4 the recognizer for the IAM-Database was adapted with the training sets from the recorded whiteboard data. To find the best value of parameter τ for the MAP estimation we calculated the performance on the validation set of the whiteboard data for different values of τ . This optimization has been done for each combination of the cross validation separately. The average recognition rate on the validation sets is 65.07% without including a language model. By including a bigram language model, the average performance could be increased to 68.60%. On the test sets it is 64.60% without and 67.48% with inclusion of a language model.

By training on the IAM-Database and the whiteboard data simultaneously as described in Section 4 the performance is even higher. The recognition rate on the validation sets is 65.64% without including a bigram language model. It increases up to 69.37% when the language model is included. On the test sets it is 65.27% and 68.49% respectively.

A summary of all experimental results is provided in Tab. 1 (validation sets) and Tab. 2 (test sets). To improve

combination	$c0$	$c1$	$c2$	$c3$	$c4$	average
validation set	$s1$	$s2$	$s3$	$s4$	$s0$	
basic system	61.2	61.4	66.5	52.2	56.4	59.5
basic system opt.	66.8	66.9	71.2	57.1	65.8	65.6
adapted system	64.1	65.3	68.3	58.6	69.1	65.1
ad. system opt.	67.1	68.4	72.3	62.9	72.4	68.6
mixed training	64.3	68.0	70.7	60.13	65.0	65.6
mixed system opt.	68.1	70.0	73.4	62.6	72.7	69.4

Table 1. Performance of the basic system, the adapted system, and the system with mixed training data without/with language model optimization on the five validation sets.

combination	$c0$	$c1$	$c2$	$c3$	$c4$	average
test set	$s0$	$s1$	$s2$	$s3$	$s4$	
basic system	60.0	62.7	59.8	65.1	50.3	59.6
basic system opt.	66.5	64.1	64.7	70.5	55.6	64.3
adapted system	68.1	63.0	66.1	69.1	57.8	64.6
ad. system opt.	70.2	66.4	68.0	72.1	60.7	67.5
mixed training	66.4	65.4	65.4	70.2	58.3	65.3
mixed system opt.	71.4	67.7	68.9	72.2	61.3	68.5

Table 2. Performance of the basic system, the adapted system, and the system with mixed training data without/with language model optimization on the five test sets.

clarity of presentation, the performance of the optimized systems is printed boldface. A statistically significant improvement over the baseline system has been achieved with both approaches. The best performing system is the HMM recognizer that has been trained on a union of the whiteboard data and a subset of the IAM-Database, including a language model. We note that for both the non-optimized and the optimized system a substantial improvement of the recognition rate is achieved by the proposed methods.

6. Conclusions and future work

In the work described in this paper we enhanced an existing handwriting recognition system for whiteboard notes. Notes written on a whiteboard is a new modality in handwriting recognition research that has received relatively little attention in the past.

The proposed recognition system is based on Hidden Markov Models. We applied two methods for enhancing the training data with data from the IAM-Database. These methods are MAP adaptation and the training on mixed datasets. A statistically significant improvement of the recognition rate over the case where just whiteboard data is used for training could be observed with both strategies. The training on mixed data consumes much more time, i.e. it lasts about ten times longer to optimize the system. But this strategy leads also to a better

performance because all model parameters θ are optimized on the validation set of the whiteboard data. The inclusion of a language model further increases the recognition rate. Note that the enlarging of the training set had a greater effect on the unoptimized system. However, it resulted also in a statistically significant increase on the optimized system.

In future research we plan to enlarge the training set by recording more whiteboard data. So we can compare the two methods proposed in this paper with the brute-force approach of just enlarging the training set. We also plan to develop an on-line recognizer and combine it with the off-line recognition system described in this paper. To further enhance the practical usefulness of the system in real life applications, recognition of mathematical formulas [11] and tables [15] will be considered.

Acknowledgments

This work was supported by the Swiss National Science Foundation program “Interactive Multimodal Information Management (IM2)” in the Individual Project “Scene Analysis”, as part of NCCR.

References

- [1] A. Brakensiek and G. Rigoll. Handwritten address recognition using hidden markov models. In A. Dengel et al., editor, *Reading and Learning*, volume 2956 of *LNCS*, pages 103–122. Springer, 2004.
- [2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of Royal Statistical Society B*, 39(1):1–38, 1977.
- [3] G. D. Forney. The Viterbi algorithm. In *Proc. IEEE*, volume 61, pages 268–278, 1973.
- [4] J.-L. Gauvain and C.-H. Lee. Map estimation of continuous density HMM: Theory and applications. In *Proceedings of DARPA Speech and Natural Language Workshop*, pages 272–277, 1992.
- [5] S. Günter and H. Bunke. HMM-based handwritten word recognition: on the optimization of the number of states, training iterations and Gaussian components. *Pattern Recognition*, 37:2069–2079, 2004.
- [6] S. Johansson. *The tagged LOB Corpus: User’s Manual*. Norwegian Computing Centre for the Humanities, Norway, 1986.
- [7] C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9:171–185, 1995.
- [8] M. Liwicki and H. Bunke. Handwriting recognition of whiteboard notes. In *Proc. 12th Conf. of the International Graphonomics Society*, 2005. Accepted for publication.
- [9] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *IJPRAI*, 15:65 – 90, 2001.
- [10] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *IJDAR*, 5:39 – 46, 2002.
- [11] E. Tapia and R. Rojas. Recognition of on-line handwritten mathematical formulas in the e-chalk system. In *Proc. 7th ICDAR*, pages 980–984, 2003.
- [12] O. Velek, S. Jäger, and M. Nakagawa. Accumulated-recognition-rate normalization for combining multiple on/off-line Japanese character classifiers tested on a large database. In *Proc. 4th Multiple Classifier Systems*, pages 196–205, 2003.
- [13] A. Vinciarelli and S. Bengio. Writer adaptation techniques in HMM based off-line cursive script recognition. *Pattern Recognition Letters*, 23(8):905–916, 2002.
- [14] A. Vinciarelli and M. Perrone. Combining online and off-line handwriting recognition. In *Proc. 7th ICDAR*, pages 844–848, 2003.
- [15] R. Zanibbi, D. Blostein, and J. Cordy. A survey of table recognition: Models, observations, transformations and inferences. *IJDAR*, 7(1):1–16, 2004.
- [16] M. Zimmermann and H. Bunke. Optimizing the integration of a statistical language model in HMM-based offline handwritten text recognition. In *Proc. 17th ICPR*, pages 541 – 544, 2004.