Recognition of Handwritten Historical Documents: HMM-Adaptation vs. Writer Specific Training

Emanuel Indermühle[1] and Marcus Liwicki[1,2] and Horst Bunke[1]

[1]Institute of Computer Science and Applied Mathematics
University of Bern, Neubrückstrasse 10, CH-3012 Bern, Switzerland
{eindermu, bunke}@iam.unibe.ch

[2]German Research Center for AI (DFKI GmbH)
Knowledge Management Department, Kaiserslautern, Germany
marcus.liwicki@dfki.de

Abstract

In this paper we propose a recognition system for handwritten manuscripts by writers of the 20th century. The proposed system first applies some preprocessing steps to remove background noise. Next the pages are segmented into individual text lines. After normalization a hidden Markov model based recognizer, supported by a language model, is applied to each text line. In our experiments we investigate two approaches for training the recognition system. The first approach consists in training the recognizer directly from scratch, while the second adapts it from a recognizer previously trained on a large general off-line handwriting database. The second approach is unconventional in the sense that the language of the texts used for training is different from that used for testing. In our experiments with several training sets of increasing size we found that the overall best strategy is adapting the previously trained recognizer on a writer specific data set of medium size. The final word recognition accuracy obtained with this training strategy is about 80%.

Keywords: handwriting recognition, hidden Markov models, historical documents, writer-specific recognition, maximum a posteriori adaptation

1 Introduction

Historical document analysis is an emerging research topic that has gained increasing attention during the last decade [1]. Problems such as word spotting [8, 17], document layout analysis [3], and handwriting recognition [4, 9] have been investigated by the research community. Especially the latter task, handwriting recognition, is challenging for a number of reasons, including training sets of small size, unusual writing styles, crossed out or overwritten words, and other artifacts.

Previous research on the recognition of handwriting in historical documents has been described in [9], where a hidden Markov model recognizer for holistic handwritten words has been applied to manuscripts of George Washington, and in [4] where HMMs as well as conditional random field models have been used for handwriting recognition on the same manuscript. In [6] the attention has been on speeding up a the recognition task for indexing historical documents, and in [15] it has been focused on character recognition in historical Greek documents.

The system described in this paper is being developed for the recognition of historic manuscripts from Swiss authors in the context of research in literature. One of the main objectives in this research is to investigate the evolution of words in a handwritten manuscript over the whole process of manuscript composition and evolution. For such studies, the transcription as well as the mapping from the text to the transcription is needed. Since machine printed editions of the manuscripts are often available and OCR on machine printed documents has good performance, one can assume that digital ASCII versions of the texts are available or can be made available. For this reason, in order to produce a mapping from the original text to the transcription, an automatic alignment seems to be sufficient, as described in [18], for example. However, as several versions of a manuscript may be involved and some manuscript editing may have been done on the original source, the printed and the handwritten versions are often not identical. Therefore, we aim at developing a recognizer that transforms a handwritten manuscript into the corresponding ASCII transcription based on just an image of the handwriting. Nevertheless, the information provided by the printed edition is highly valuable if used to build the language model.

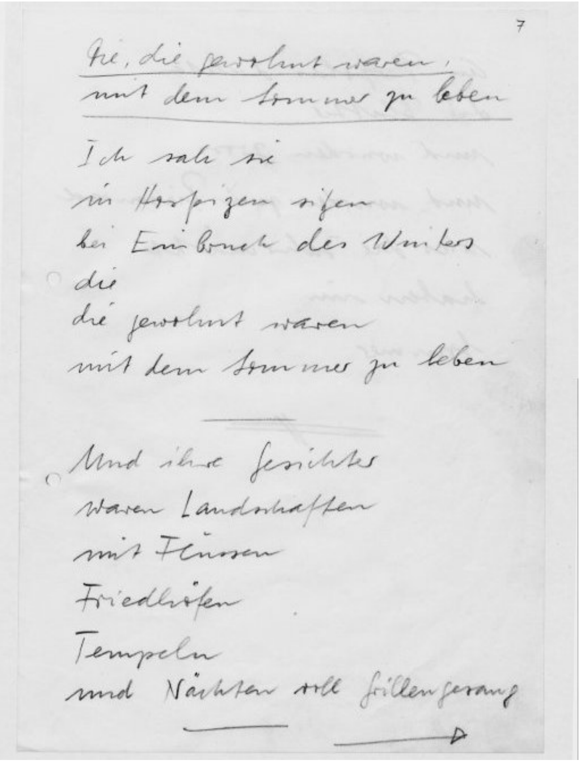
A particular issue studied in the work described in this paper is how to optimally train such a recognizer. From previous studies it is known that writer-dependent systems, i.e. systems that have been trained on just a single writer and are supposed to process only text from the same writer, exhibit superior performance over writer-independent systems, where the population of writers who produced the training set is different from the writers who produced the test set [11]. Clearly, in the scenario considered in this paper, the author of a handwritten text to be transcribed is usually known in advance. Therefore, a writer-dependent approach can be taken. However, the amount of training data available from a single writer is typically limited. By contrast, huge amounts of training data become available if a writer-independent approach is adopted. Yet, such a general system may be not well adapted to the writing style of the particular author under consideration. A third way in between these two possible solutions is an adaptive training procedure [19]. Under such a procedure, a writer-independent system is trained first, using a text collection as large as possible. Then this system is iteratively refined and adapted using additional training data from the specific writer in question. Such an adaptation strategy is investigated in this paper and compared to a writer-dependent approach where a recognizer is trained from scratch.

The rest of this paper is organized as follows. Section 2 gives background information about the underlying data. Next, Section 3 explains the preprocessing, segmentation and feature extraction steps applied before recognition. In Section 4 we introduce the basic recognition system and in Section 5 the adaptation techniques are described. Experiments and results are presented in Section 6. Finally. Section 7 draws some conclusions and provides an outlook to future work.

2 Data of the Swiss Literary Archives

The Swiss Literary Archives[1] in the Swiss National Library maintain a large collection of various works from Swiss authors. In particular, this collection includes a huge number of original handwritten documents. In the context of research in literature, there is an interest in automated tools that make all these materials electronically accessible. Recognition systems that are able to automatically transcribe handwritten manuscripts into ASCII format play an important role in this research.

The handwriting material used in the study described in this paper consists of two volumes of poetry manuscripts from Swiss author Gerhard Meier (* 1917), including 145 pages and 1,640 lines in total. The pages are provided as digital gray scale images with a resolution of 4,000 by 3,000 pixels.



7

Sie, die gewohnt waren,
mit dem Sommer zu leben

Ich sah sie
in Hospizen sitzen
bei Einbruch des Winters
die
die gewohnt waren
mit dem Sommer zu leben

Und ihre Gesichter
waren Landschaften
mit Flüssen
Friedhöfen
Tempeln
und Nächten voll Grillengesang

Figure 1. Page of a manuscript by Gerhard Meier.

Gerhard Meier used to write on individual sheets of paper, which he collected in a folder. He used a pencil and corrected his writing with an eraser until it was ready for printing. Therefore no corrections appear in the writing. He also used rather generous spacing between lines, which is an advantage for the segmentation. However, there are artifacts such as underlining of the title, arrows, lines indicating the end of a page, lines separating two consecutive paragraphs from each other, punching holes, page margins, and some others (see Figure 1).

3 Preprocessing

The proposed recognition system needs individual text lines as input. Since better recognition results are achieved if the text lines are normalized, some preprocessing steps are applied. First the images are binarized. Then the pages are segmented into individual lines. Finally the line images are normalized.

For binarization, we use Otsu's algorithm [16]. In the next step some distortions, mentioned above, are removed. As the main focus of the current paper is on the recognition task, these operations are performed manually. However, as most of these distortions can well be distinguished from the writing, an automated elimination would be possible. Then a recently developed line extraction proce-

[1] Website: `http://www.nb.admin.ch/slb/`

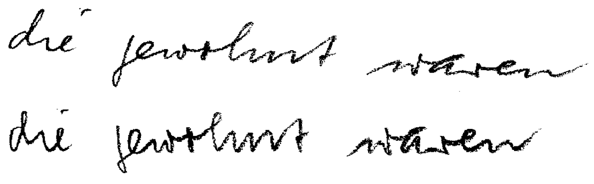


Figure 2. Text line before and after normalization.

dure [12] for online handwriting, which has been modified so as to deal with offline data, is applied on the binarized images.

After line extraction, the normalization steps proposed in [13] are applied. First, the skew of the considered text line is corrected. For this purpose, the lowest black pixel is determined for each image column. Thus the lower contour of the writing is obtained. The skew angle of the line can then be computed by a regression analysis on this set of points. Once the skew angle is determined the line is rotated such that it becomes parallel to the horizontal axis. After deskewing, a slant correction is done. Here we measure the angle between the writing and the vertical direction. For this purpose, the contour of the writing is approximated by small lines. The directions of these lines are accumulated in an angle histogram. The angle corresponding to the maximum value in the histogram gives the slant. After the slant angle has been determined, a shear operation is applied to bring the writing in an upright position. For the vertical positioning of the text line, the lower baseline determined during skew correction serves as a line of reference. Given this line, a scaling procedure is applied. For this procedure, we need to additionally know the upper baseline, which is computed by a horizontal projection of the text line. To the histogram of black pixels resulting from the horizontal projection, an ideal histogram is fitted. From this ideal histogram the position of the upper baseline is obtained. The bounding box of a line of text together with the upper and lower baseline define three disjoint areas (upper, middle, and lower). Each of these areas is scaled in vertical direction to a predefined size. For horizontal scaling the black-white transitions in the considered line of text are counted. This number of transitions can be set in relation to the mean number of transitions in a text line, which is determined over the whole training set. Thus the scaling factor for the horizontal direction is obtained. All preprocessing operations described above, in particular positioning and scaling, are required to make the feature extraction procedure described in the next section properly working. Figure 2 illustrates the normalization on a text line from Figure 1.

4 Recognition System

The recognition system used in this paper is based on hidden Markov models (HMM). It is similar to the one proposed in [13]. For the purpose of completeness a short introduction is given.

The normalized text line images are the input to the recognizer. Prior to recognition, features are extracted. For feature extraction, a sliding window of one pixel width is moved over the image from left to right. At each position of the window, a vector of nine features is extracted. So each text line image is converted into a sequence of 9-dimensional feature vectors. The features extracted are:

- the number of pixels;
- the center of gravity of the pixels;
- the second order moment of the window;
- the location of the upper-most pixel;
- the location of the lower-most pixel;
- the orientation of the upper-most pixel;
- the orientation of the lower-most pixel;
- the number of black-white transitions; and
- the number of black pixels divided by the number of all pixels between the upper- and the lower-most pixel.

The texts to be recognized are based on a set of 78 characters, containing small and capital Latin letters, as well as German and French umlauts and some punctuation marks. For each of these characters, an HMM with a linear topology and 16 states is built. The observation probability distributions are estimated by a mixture of Gaussian components. In other words, continuous HMMs are used. The character models are concatenated to represent words and sequences of words. For training, the Baum-Welch algorithm [2] is applied. In the recognition phase, the Viterbi algorithm [5] is used to find the most probable word sequence. As a consequence, 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. An important parameter in the recognition process is the number of Gaussian components in the observation probability distribution. This parameter is optimized on a validation set.

Since the system proposed in this paper is performing handwriting recognition on text lines and not on single words, it is reasonable to integrate a statistical language model. In [20] it was shown that by means of such an integration the performance of the recognizer can be significantly improved. Two parameters are needed for the language model integration, viz., the Grammar Scale Factor (GSF) and the Word Insertion Penalty (WIP). The first parameter weights the influence of the language model against the HMM, while the latter parameter can prevent the system from over- and undersegmentation. We also optimize those parameters on a validation set.

As for our particular data set an electronic transcription exists, we calculate the language model therefrom. Clearly, in general such a transcription does not exist. However, in the presented case, this procedure seems adequate as the Swiss Literary Archives are in possession of a transcription of many handwritten manuscripts. Otherwise, the language model has to be generated from a general text corpus.

5 Adaptation

An HMM recognizer, as described in Section 4, was trained using exclusively data from author Gerhard Meier. However, since there is only a limited amount of annotated data available, we also trained an HMM-recognizer on the IAM-database [14] and then adapted it to the manuscript data.

The IAM-Database consists of 13,353 lines and 115,320 words from 657 writers. All texts are sentences from the LOB Corpus [7]. Since this corpus is in English it is not possible to train the HMMs of all characters of the manuscript (mainly German characters), such as ä,ö,ü,Ä,Ö,Ü,ï, and ë. Furthermore, although they may occur in English text, the symbols (,),/,- and the numbers from 0 to 9 are missing in the transcription of the IAM-database as well [14]. To solve this issue we took the missing HMMs from the HMM-recognizer trained with the manuscript data.

HMM adaptation [19] 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 manuscript data). The aim is to find the vector θ_{ad} which maximizes the *posterior* distribution $p(\theta_{ad}|O)$:

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

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

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

where $p(O|\theta)$ is the likelihood of the HMM with parameter set θ and $p(\theta)$ is the *prior* distribution of the parameters. 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 [10]) 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

$$\frac{\partial}{\partial\theta}\left(p(O|\theta)p(\theta)\right) = 0 \tag{3}$$

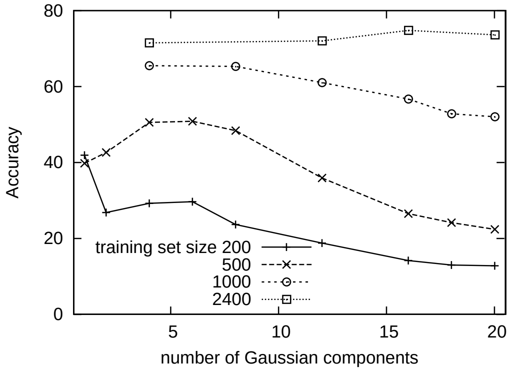


Figure 3. The number of Gaussian components used to train recognizers from scratch has an important influence on the accuracy. The accuracy values are calculated on the validation set.

This minimizes the Bayes risk over the adaptation set and can be done with Maximum A Posterior (MAP) estimation, which is also called Bayesian Adaptation. As described in [19], 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 [19]:

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

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 in Eq. (4) weights the influence of the background model on the adaptation data. If the parameter τ is set to 0 the new means $\hat{\mu}_{jm}$ become equal to the means $\bar{\mu}_{jm}$ of the adaptation data, ignoring the means μ_{jm} of the background model. That is, only the manuscript data have an influence on the adapted Gaussians. Otherwise, if τ is very large only the means of the background model are considered and no adaptation takes place. Whereas parameter τ has been set empirically in [19] it is optimized on a validation set in this paper.

6 Experiments and Results

For the experiments described in this section we used the manuscripts by Gerhard Meier mentioned in Section 2. They contain 4,890 words in 1,640 lines. The data were randomly divided into a training, a validation, and a test set. While the size of the training set varies from 200 to

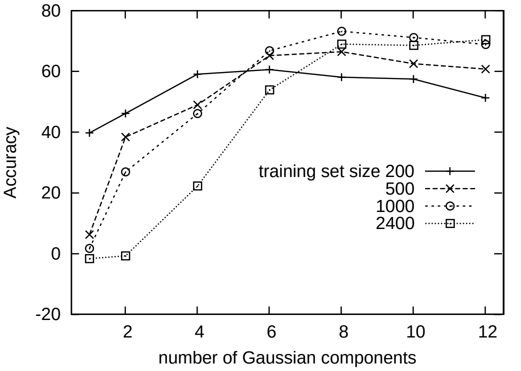


Figure 4. A lower number of Gaussians used to train the HMMs on the IAM-database leads to better results on smaller training sets. However, for large training sets, more Gaussians are needed. The accuracy values are calculated on the validation set.

500, 1,000 and 2,400 word instances, the validation and the test sets are fixed. The validation set contains 20% and the test set 30% of the data, and all the sets are mutually disjoint. The word dictionary includes those words that occur in the union of all lines.

In the first set of experiments (later referred to as *trained from scratch*), the HMM-recognizer is trained on the four different writer specific training sets, varying the number of Gaussian components from 1 to 20. Furthermore, the parameter GSF has been varied from 0 to 70 and the parameter WIP from -50 to 50.

In the second set of experiments (referred to as *adaptation*), the HMM-recognizer is trained on a subset of the IAM-database, varying the number of Gaussian components from 1 to 12. This subset contains 6,161 lines and 53,841 words from 283 writers. As mentioned before there are HMMs missing in the IAM-database, these models have been taken from the recognizer trained from scratch with the best performance on the validation set. The adaptation is made on the same training set on which the recognizer trained from scratch is trained. The parameter τ is optimized on the validation set to find the best level of adaptation.

For all the experiments, the task was to transcribe the text lines in the test set, given the words in the dictionary. As basic performance measure the *word accuracy* was used, which is defined as:

$$100 * \left(1 - \frac{\textit{insertions} + \textit{substitutions} + \textit{deletions}}{\textit{total length of test set transcriptions}}\right) \quad (5)$$

where *insertions*, *substitutions*, and *deletions* denotes the number of insertions, substitutions, and deletions on the word level, respectively, needed to make the recognition result identical to the ground truth.

It is well known that the number of Gaussian components has an important influence on the accuracy of a recognizer. We compare this influence in conjunction with varying the size of the training set. For the results of the recognizer trained from scratch see Figure 3. In Figure 4 the results of the adaptation are presented. Note that the x-axis has a different scale in Figures 3 and 4. In both figures, one can see that on training sets of smaller size (used for adaptation in Figure 4), the experiments with less Gaussian components lead to better results. This can be explained by the fact that a higher number of Gaussians results in an overfitting on smaller training sets.

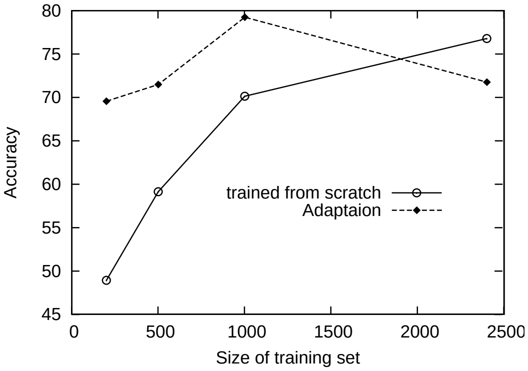


Figure 5. Performance of the recognizer trained from scratch vs. the adaptation-based recognizer on the test set with training sets of increasing size.

The results on the test set are compared in Figure 5. As can be seen, adaptation raises the accuracy on smaller sizes of the training set. The final performance of 79.24% is a quite promising result.

7 Conclusion and Future Work

The recognition of handwritten text in historical manuscripts has gained increasing attention in recent years. In this paper we have described a prototypical system that addresses this problem. Our particular focus of attention is on the automatic reading of handwritten manuscripts by Swiss authors of the 20th century.

It is a well known fact that the more training data is available for a recognizer the higher is its expected recognition performance. Furthermore, writer-specific recognizers which have been trained on the handwriting of one particular person and have to recognize only this person's handwriting, exhibit a higher performance than writer-independent systems which have been trained on data from multiple writers and are applied to recognize text from writers not represented in the training set. The task considered in this paper is amenable to using a writer-specific approach as the identity of the author whose handwriting is to be transcribed is always known. However, the amount of training data is severely limited if only data from a single writer can be used.

In this paper we have investigate an adaptation based approach, where a general recognizer is trained first, using training data from a large database including handwriting samples of many different writers. This writer-independent recognizer is then adapted with training data from the specific writer whose manuscripts are to be transcribed. The amount of training data used for the adaptation is varied. The resulting system is compared to a system that is trained from scratch using only training data from the specific writer. Our results on an independent test set indicate that the performance of the second system increases monotonically with a growing amount of training data, which perfectly fits our expectation. The first system, however, reaches its peak performance if a writer-specific adaptation set of medium size is used. The maximum performance of the first system is superior to that of the second system at a statistical significance level of 99%, using a standard Z-test. A recognition accuracy of almost 80%, as reported in Section 6, seems very promising for future practical applications of the system.

The IAM-database, used for training the writer-independent system, has several differences from the manuscripts. Beside the use of other writing tools, it is important to note that the database contains text written in English, while the manuscripts to be recognized are written in German. This raises the problem of missing characters models of the recognizer. However, this problem can be solved by taking the missing models from the writer-dependent system.

Our results lead to the conclusion that the best strategy to build a recognizer for the handwriting of a particular author consists in taking a general writer-independent system, which has been constructed before, using general training data from various writers. Then only a relatively small amount of writer-specific training data (in our case 1,000 words) are needed in order to obtain a recognizer that performs better than both the writer-independent system and a recognizer trained with only writer specific data.

From the practical point of view, labeling a set of training data consisting of approximately 1,000 words seems not too expensive. Hence, in order to transcribe larger portions of an archive with material from multiple writers, it appears feasible to configure a recognizer for each individual writer by means of the adaptation procedure described in this paper. In future work we plan to verify the findings reported in this paper on data from other writers. Also the extension of this study from the case of recognition to automatic alignment is a topic worth to be investigated.

Acknowledgments

We thank Dr. Corinna Jäger-Trees, curator of the Gerhard Meier fonds in the Swiss Literary Archives, and her colleagues for their kind support.

References

[1] A. Antonacopoulos and A. C. Downton. Special issue on the Analysis of Historical Documents. *Int. Journal Document Analysis Recognition*, 9(2):75–77, 2007.

[2] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.

[3] M. Feldbach and K. D. Tonnies. Line detection and segmentation in historical church registers. In *Proc. 6th Int. Conference on Document Analysis and Recognition*, pages 743–747, 2001.

[4] S. Feng, R. Manmatha, and A. McCallum. Exploring the use of conditional random field models and HMMs for historical handwritten document recognition. In *Proc. 2nd Int. Conference on Document Image Analysis for Libraries*, pages 30–37, 2006.

[5] G. D. Forney. The Viterbi algorithm. *Proc. IEEE*, 61(3):268–278, 1973.

[6] V. Govindaraju and H. Xue. Fast handwriting recognition for indexing historical documents. In *Proc. 1st Int. Workshop on Document Image Analysis for Libraries*, pages 314–320, 2004.

[7] S. Johansson, E. Atwell, R. Garside, and G. Leech. *The Tagged LOB Corpus, User's Manual*. Norwegian Computing Center for the Humanities, Bergen, Norway, 1986.

[8] E. M. Kornfield, R. Manmatha, and J. Allan. Text alignment with handwritten documents. In *Proc. 1st Int. Workshop on Document Image Analysis for Libraries*, pages 195–205, 2004.

[9] V. Lavrenko, T. Rath, and R. Manmatha. Holistic word recognition for handwritten historical documents. In *Proc. Int. Workshop on Document Image Analysis for Libraries*, pages 278–287, 2004.

[10] C. Leggeter and P. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models. *Computer Speech and Language*, 9:171–185, 1995.

[11] M. Liwicki and H. Bunke. Writer-dependent handwriting recognition of whiteboard notes. 2008. Submitted.

[12] M. Liwicki, E. Indermühle, and H. Bunke. Online handwritten text line detection using dynamic programming. In *Proc. 9th Int. Conference on Document Analysis and Recognition*, volume 1, pages 447–451, 2007.

[13] U.-V. Marti and H. Bunke. Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system. *Int. Journal of Pattern Recognition and Artificial Intelligence*, 15:65–90, 2001.

[14] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *Int. Journal on Document Analysis and Recognition*, 5:39–46, 2002.

[15] K. Ntzios, B. Gatos, I. Pratikakis, T. Konidaris, and S. J. Perantonis. An old greek handwritten OCR system based on an efficient segmentation-free approach. *Int. Journal Document Analysis Recognition*, 9(2):179–192, 2007.

[16] N. Otsu. A threshold selection method from gray-scale histogram. *IEEE Trans. Systems, Man, and Cybernetics*, 8:62–66, 1978.

[17] T. M. Rath and R. Manmatha. Word spotting for historical documents. *Int. Journal Document Analysis Recognition*, 9(2):139–152, 2007.

[18] C. I. Tomai, B. Zhang, and V. Govindaraju. Transcript mapping for historic handwritten document images. In *Proc. 8th Int. Workshop on Frontiers in Handwriting Recognition*, pages 413–418, Washington, DC, USA, 2002.

[19] A. Vinciarelli and S. Bengio. Writer adaptation techniques in hmm based off-line cursive script recognition. *Pattern Recognition Letters*, 23(8):905–916, 2002.

[20] M. Zimmermann and H. Bunke. N-gram language models for offline handwritten text recognition. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 203–208, 2004.