IAM-OnDB - an On-Line English Sentence Database Acquired from Handwritten Text on a Whiteboard

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

Abstract

In this paper we present IAM-OnDB - a new large on-line handwritten sentences database. It is publicly available and consists of text acquired via an electronic interface from a whiteboard. The database contains about 86 K word instances from an 11 K dictionary written by more than 200 writers. We also describe a recognizer for unconstrained English text that was trained and tested using this database. This recognizer is based on Hidden Markov Models (HMMs). In our experiments we show that by using larger training sets we can significantly increase the word recognition rate. This recognizer may serve as a benchmark reference for future research.

1. Introduction

The recognition of unconstrained handwritten text is still a great challenge although research in the area has started more than 30 years ago [1, 17, 22]. Usually the discipline of handwriting recognition is divided into off-line and on-line recognition. In off-line recognition the handwriting of a user is given in terms of a static image, while in the on-line mode it is a time dependent signal that represents the location of the tip of the pen as a user is writing. Traditionally off-line handwriting recognition has applications in postal address reading [19] as well as bank check and forms processing [8]. Recent applications of on-line handwriting recognition include pen computing [4] and tablet pcs [9].

In this paper we consider a new input modality which is text written on a whiteboard. Thanks to inexpensive acquisition devices that became available recently (for more details see Section 3), the automatic transcription of notes written on a whiteboard has gained interest. In the particular application underlying this paper we aim at developing a handwriting recognition system that is to be used in a smart meeting room scenario [24], in our case the smart meeting room developed in the IM2 project [16]. In a smart meeting room we typically find multiple microphones and video cameras that record a meeting. In order to allow for retrieval of the meeting data by means of a browser, semantic information needs to be extracted from the raw sensory data, such as transcription of speech and recognition of persons in video images. Whiteboards are commonly used in meeting rooms. Hence capture and automatic transcription of handwritten notes on a whiteboard are essential tasks in a smart meeting room application.

It is a well-known fact that all handwriting recognizers, such as neural networks, support vector machines, or Hidden Markov Models (HMMs), need to be trained. Common experience is that the larger the training set is the better performs the recognizer. However, the acquisition of large amounts of training data is a time consuming process that has clear limitations. Therefore it is important that existing databases for training and testing are shared in the research community. The UNIPEN database [6] is a large on-line handwriting database. It contains mostly isolated characters, single words, and a few sentences on several topics. Another on-line word database is IRONOFF [21]. It additionally contains the scanned images of the handwritten words. For the task of off-line handwriting recognition there are also databases available, including CEDAR [7], created for postal address recognition, NIST [25], containing image samples of handprinted characters, CENPARMI [11], consisting of handwritten numerals, and the IAM-Database [15], a large collection of unconstrained handwritten sentences.

As automatic reading of whiteboard notes is a relatively new task, no publicly available databases do exist for this modality, to the knowledge of the authors. The purpose of the current paper is twofold. First we describe a large database of handwritten whiteboard data that was recently acquired in our laboratory. This database is publicly available on the World Wide Web[1]. Secondly we describe a first recognizer developed for the task of reading notes on a

[1] http://www.iam.unibe.ch/˜fki/iamondb/

whiteboard. This recognizer may serve as a reference for further research in this field.

The rest of the paper is organized as follows. Section 2 describes the design of IAM-OnDB. In Section 3 the data acquisition process is presented. Section 4 gives an overview of the system for whiteboard note recognition, including some optimization steps. Experiments and results are presented in Section 5, and finally Section 6 draws some conclusions and gives an outlook for future work.

2. The database

The design of the database described in this paper, called IAM-OnDB, is inspired by the IAM-Database presented in [15]. However, while the IAM-Database is an off-line database, the IAM-OnDB consists of on-line data acquired from a whiteboard. All texts included in the IAM-OnDB are taken from the Lancaster-Oslo/Bergen corpus (LOB), which is a large electronic corpus of text [10]. Using the LOB corpus as the underlying source of text makes it possible to automatically generate language models, such as statistical n-grams and stochastic grammars [18]. Consequently linguistic knowledge beyond the lexicon level can be integrated in a recognizer. The LOB corpus contains 500 English texts, each consisting of about 2,000 words. These texts are of quite diverse nature. They are divided into 15 categories ranging from press and popular literature to learned and scientific writing.

To acquire a database of handwritten sentences contained in the corpus we split the texts in the corpus into fragments of about 50 words each. These fragments were copied onto forms on paper and each writer was asked to write down the text of eight forms on the whiteboard. To make sure that many different word samples are obtained from each writer, we have chosen these eight texts from different text categories in the LOB corpus. The resulting database consists of more than 1,700 handwritten forms from 221 writers. It contains 86,272 word instances from a 11,059 word dictionary written down in 13,049 text lines.

In addition to the recorded data and its transcription some informations about the writers, which could be useful for future work, are stored in the IAM-OnDB. These include, for each writer, the native country and language, other mastered languages, age and gender, and the writing style, i. e. right- or left-handed writing style. The writers who contributed to the database were all volunteers. Most of them are students and staff members of the University of Bern. Both genders are about equally represented in the database and about 10% of the writers have left-handed writing.

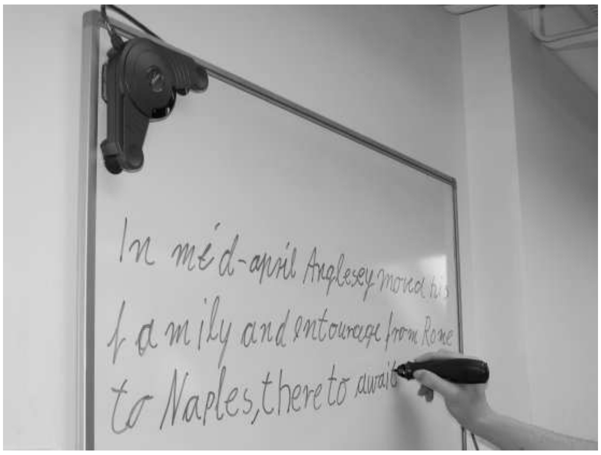
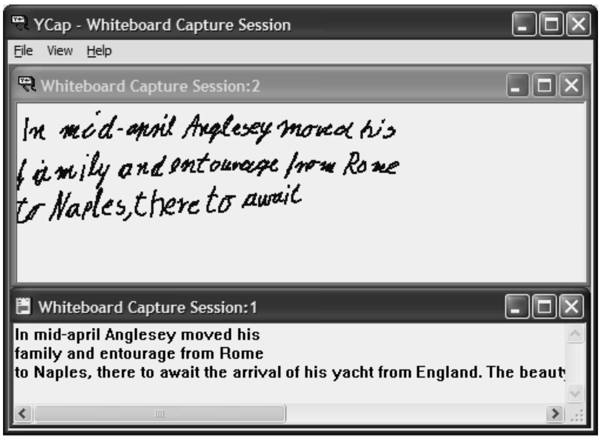


Figure 1. Illustration of the recording; note the data acquisition device in the left upper corner of the whiteboard


YCap - Whiteboard Capture Session
File View Help
Whiteboard Capture Session:2
In mid-april Anglesey moved his
family and entourage from Rome
to Naples, there to await
Whiteboard Capture Session:1

In mid-april Anglesey moved his
family and entourage from Rome
to Naples, there to await the arrival of his yacht from England. The beaut

Figure 2. Interface of the recording software

3. Acquisition

The eBeam[2] interface is used to record 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.

Labeling of the data is a prerequisite for recognition experiments. It is advisable to do as much as possible automatically because labeling is expensive, time consuming and

[2] eBeam System by Luidia, Inc. - www.e-Beam.com

error prone. During the recordings an operator observes the received data with a special recording software written at our laboratory. The software first loads the ASCII transcription of the text to be written. While the writer renders the handwritten text the operator adjusts the line feeds during recording. He or she is also able to make corrections if the handwritten text does not correspond to the printed text, for example, if the writer leaves out some words. Fig. 2 shows a screen shot of the interface. The transcription produced by the operator in the lower window is saved together with the recorded on-line data in one xml-file.

The raw data stored in one xml-file usually includes several consecutive lines of text. For the recognizer and the experiments described in Section 4 and 5, respectively, we need to segment the text into individual lines. The line segmentation process of the on-line data is guided by heuristic rules. If there is a pen-movement to the left and vertically down that is greater than a predefined threshold, a new line is started. This method succeeds on more than 99% of the text forms. There are only two cases where a line is too short and a few cases where the writer moved back and forth across different text lines to render an i-dot. To ensure that the automatic line segmentation has been done correctly the resulting lines are checked by the operator and corrected if necessary. Consecutive lines are highlighted in different colors on the screen so that an error can be easily detected.

4. Recognition system overview

A basic cursive handwriting recognition system has been trained and tested on the database described in the previous sections. A preliminary version of this recognition system was introduced in [13], and its adaptation to a different training modality described in [12]. The recognizer is derived from the Hidden Markov Model (HMM) based system proposed in [14]. Although the handwriting captured in the database described in this paper is in the on-line mode, the recognizer takes off-line handwritten lines of text as its input. Using off-line rather than on-line data has two reasons. First, the existing recognizer [14] has been designed for off-line data, and secondly, it is straightforward to convert on-line data to the off-line modality. Eventually we plan to additionally build an on-line recognizer and combine it with the existing off-line system. From such a combination, enhanced recognition performance can be expected [20, 23].

An overview of our whiteboard data handwriting recognition system is shown in Fig. 3. The system consists of six main modules: 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; 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 an n-best list of word sequences; and the post-processing, where a statistical language model is applied to improve the results generated by the HMM. In the remainder of this section more details of the individual modules will be provided.

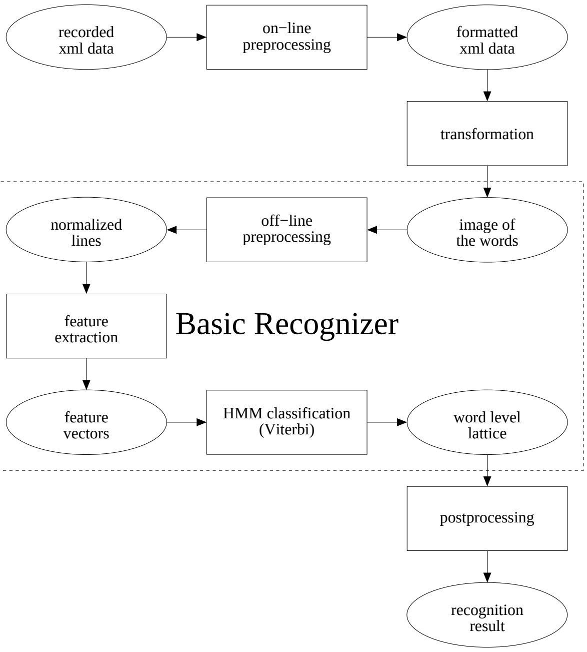


Figure 3. Recognition system overview

The recorded on-line data usually contain noisy points and gaps within strokes. Thus two on-line preprocessing steps are applied to the data, to recover from artifacts of this kind. Let $p_1, ..., p_n$ be the points of a given stroke and q_1 be the first point of the succeeding stroke, if any. To identify noisy points, we check whether the distance between two consecutive points p_i, p_{i+1}, is larger than a fixed threshold. In this case one of the points is deleted. To decide which point has to be deleted, the number of points within a small neighborhood of p_i and p_{i+1} are determined, and the point with a smaller number of neighbors is deleted. To recover from artifacts of the second type, i.e. from gaps within strokes, we check if the distance between the timestamps of p_n and q_1 is under a fixed threshold. If the condition is true the strokes are merged into one stroke.

Since the preprocessed data is still in the on-line format, it has to be transformed into an off-line image, so that it can be used as input for the off-line recognizer. The recognizer was originally designed for the off-line IAM-Database [15] and optimized on gray-scale images scanned with a resolution of 300 dpi. To get good recognition results in the considered application, the produced images should be similar to these off-line images. Consequently the following steps are applied to generate the images. First, all consecutive points within the same stroke are connected. This results in one line segment per stroke. Then the lines are dilated to a width of eight pixels. The center of each line is colored black and the pixels are getting lighter towards the periphery.

The basic recognizer is a Hidden Markov Model (HMM) based cursive handwriting recognizer similar to the one described in [14]. It 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 with 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 algorithm [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 has been 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 36 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 [26] is the inclusion of a language model, which corresponds to the post-processing step illustrated in Fig. 3. 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 [26].

5. Experiments and results

In this section we report on a number of experiments with the database and the recognizer introduced in this paper. These experiments were conducted for the purpose of getting a first impression of how difficult the reading of whiteboard notes is. Intuitively one can expect that the quality of handwriting on a whiteboard is lower than on-line handwriting produced on an electronic writing tablet or off-line handwriting scanned in from paper, for at least two reasons. First, most people are much more used to writing with a normal pen on paper or even with an electronic pen on a writing tablet than to writing on a whiteboard. Secondly, when using a normal pen on paper or an electronic pen on a tablet, the writer's arm usually rests on a table. By contrast, when writing on a whiteboard, one usually stands in front of the whiteboard and the arm does not rest on any surface, which puts much more stress on the writers' hand. Therefore we must expect more noise and distortions in whiteboard handwriting than in normal on-line or off-line handwritten data.

To investigate the effect of a growing amount of training data, we first trained and tested the recognition system on a small data set produced by 20 writers. Next, we used a larger training set produced by 50 writers and tested the recognizer under the same conditions as on the small data set. Finally, we used the full IAM-OnDB for the experiments. For all these experiments the same test set was used always under the same conditions. The language model was generated from the LOB-Corpus, which contains 500 printed texts of about 2,000 words each.

In the first three experiments a dictionary of size 2,337 words was used. It contains exactly those words which occur in the test set. In the experiment with the small training set, 6,204 words in 1,258 lines from 20 different writers were available. This data set was randomly divided into five disjoint sets of approximately equal size (sets $s_0, \ldots, s_4$). On these sets, 5-fold cross validation was performed in the following way (combinations $c_0, \ldots, c_4$). For $i = 0, \ldots, 4$, sets $s_{i \oplus 2}, s_{i \oplus 3}$ and $s_{i \oplus 4}$ were taken for training 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. No writer appeared in more than one set. Consequently, writer-independent recognition experiments were conducted. The average word recognition rate of this recognizer is 59.54% on the validation sets and 59.59% on the test sets. By in-

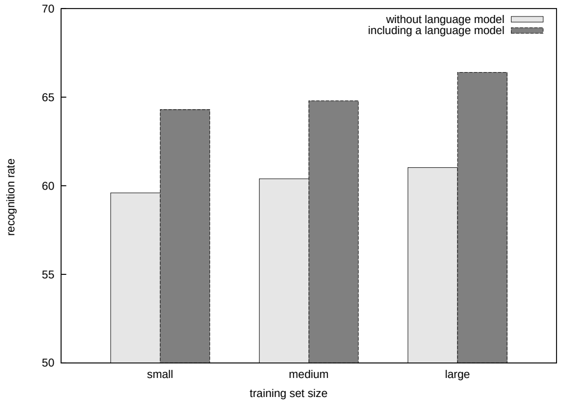


Figure 4. Recognition rate on the test set by using a small dictionary

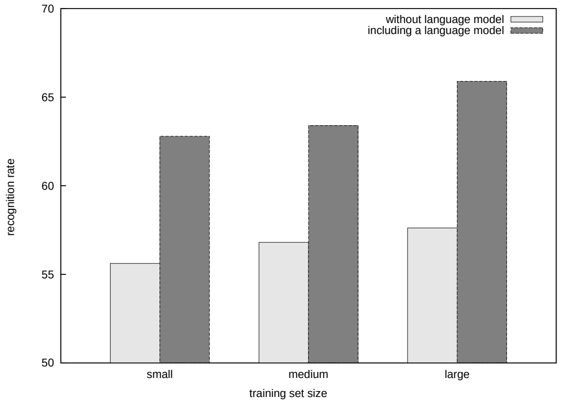


Figure 5. Recognition rate on the same test set by using a large dictionary

tegrating a language model as described in Section 4 the recognition rate could be increased to 65.56% on the validation sets and to 64.27% on the test sets.

For the next experiment we added the texts of 30 more writers to each training set. We validated the optimization parameters on the same validation sets and tested the performance on the same test sets. The average recognition rate on the validation sets is 61.17% without a language model. It increases to 66.22% by including a language model. On the test sets it is 60.39% without and 64.81% with inclusion of a language model.

In the last experiments all data from the 201 writers that do not appear in the test sets was used for training. There the average recognition rate is 61.75% on the validation sets and 61.03% on the test sets. By integrating a language model the performance could be increased to 68.07% on the validation sets and 66.4 on the test sets.

Fig. 4 gives a summary of the experimental results on the test set. The performance could be increased by 2.1% by using the large data sets for training. This increase is statistically significant ($\alpha = 1\%$).

We also tested the trained recognizers on the large 11 K word dictionary that includes all words in the database to study the effect of increasing the dictionary size (see Fig. 5). The average recognition rate of the optimized system which has been trained on the small database is 62.80% on the test sets when the language model is included. The effect of using a larger training database is greater than on the small dictionary. The recognition rate increased by 0.6% to 63.38% for the medium size and statistically significantly ($\alpha = 1\%$) by 3.1% to 65.90% for the large training set. This performance is only 0.5% below the performance on the small dictionary which has only about one fifth of the size.

In Figs. 4 and 5 can be observed that the inclusion of a language model has a larger effect if the word dictionary contains more words. While the performance of the system trained on the large database increases by 5.4% on the 2.3 K dictionary, it increases by 8.3% on the 11 K dictionary. This is because many errors that could have been corrected by using a dictionary are now corrected by using linguistic information.

6. Conclusions and future work

In this paper we have addressed a new task in cursive handwriting recognition, which is the automatic reading of cursive text from a whiteboard. This modality is emerging in new applications, for example, in the context of smart meeting rooms. First a new database of handwritten whiteboard text has been described. To the knowledge of the authors, this is the first public handwritten sentence database which is based on a whiteboard as input modality. It consists of 86,272 word instances over an 11,059 word dictionary written by 221 writers, where each writer wrote approximately the same number of words. It is planned to make this IAM-OnDB a part of the UNIPEN database [6] soon.

Furthermore we have introduced a recognizer for whiteboard handwriting. It is based on HMMs and includes a statistical bigram language model. This recognizer may serve as a benchmark for future research. In a number of experiments it was confirmed that increasing the size of the training set leads in fact to higher recognition rates. On the 11 K word dictionary the recognition rate could be increased by 3.1% to 65.9%. This increase is statistically significant. From this point of view the database described in this paper, which is publicly available, may be useful to the research community for improving the quality of handwriting recognition systems, particularly in the context of handwriting data acquired from a whiteboard.

Acknowledgments

This work was supported by the Swiss National Science Foundation program "Interactive Multimodal Information Management (IM)2" in the Individual Project "Scene Analysis", as part of NCCR. The authors thank all volunteers who took the time for participating in the recordings. We also thank Dr. Darren Moore for providing us a basic driver software for the eBeam system.

References

[1] H. Bunke. Recognition of cursive roman handwriting - past present and future. In *Proc. 7th ICDAR*, volume 1, pages 448–459, 2003.

[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] N. Furukawa, H. Ikeda, Y. Kato, and H. Sako. D-pen: A digital pen system for public and business enterprises. In *Proc. 9th IWFHR*, pages 269–274, 2004.

[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] I. Guyon, L. Schomaker, R. Plamondon, M. Liberman, and S. Janet. Unipen project of on-line data exchange and recognizer benchmarks. In *Proc. 12th ICPR*, pages 29–33, 1994.

[7] J. J. Hull. A database for handwritten text recognition research. *IEEE TPAMI*, 16(5):550–554, 1994.

[8] S. Impedovo, P.Wang, and H.Bunke. *Automatic Bankcheck Processing*. World Scientific, 1997.

[9] N. Iwayama, K. Akiyama, H. Tanaka, H. Tamura, and K. Ishigaki. Handwriting-based learning materials on a tablet pc: A prototype and its practical studies in an elementary school. In *Proc. 9th IWFHR*, pages 533–538, 2004.

[10] S. Johansson. *The tagged LOB Corpus: User's Manual*. Norwegian Computing Centre for the Humanities, Norway, 1986.

[11] S.-W. Lee. Off-line recognition of totally unconstrained handwritten numerals using multilayer cluster neural network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18(6):648–652, 1996.

[12] M. Liwicki and H. Bunke. Enhancing training data for handwriting recognition of whiteboard notes with samples from a different database. Accepted for publication, 2005.

[13] M. Liwicki and H. Bunke. Handwriting recognition of whiteboard notes. In *Proc. 12th Conf. of the International Graphonomics Society*, 2005. Accepted for publication.

[14] 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.

[15] U.-V. Marti and H. Bunke. The IAM-database: an English sentence database for offline handwriting recognition. *IJDAR*, 5:39 – 46, 2002.

[16] D. Moore. The IDIAP smart meeting room. Technical report, IDIAP-Com, 2002.

[17] R. Plamondon and S. N. Srinhari. On-line and off-line handwriting recognition: A comprehensive survey. In *IEEE TPAMI*, volume 22, pages 63–84, 2000.

[18] R. Rosenfeld. Two decades of statistical language modeling: Where do we go from here? In *Proc. IEEE 88 (8)*, 2000.

[19] S. N. Srihari. Handwritten address interpretation: A task of many pattern recognition problems. *IJPRAI*, 14(5):663–674, 2000.

[20] 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.

[21] C. Viard-Gaudin, P. M. Lallican, P. Binter, and S. Knerr. The IRESTE on/off (IRONOFF) dual handwriting database. In *Proc. 5th ICDAR*, pages 455–458, 1999.

[22] A. Vinciarelli. A survey on off-line cursive script recognition. *Pattern Recognition*, 35(7):1433–1446, 2002.

[23] A. Vinciarelli and M. Perrone. Combining online and offline handwriting recognition. In *Proc. 7th ICDAR*, pages 844– 848, 2003.

[24] A. Waibel, T. Schultz, M. Bett, R. Malkin, I. Rogina, R. Stiefelhagen, and J. Yang. Smart: The smart meeting room task at isl. In *Proc. IEEE ICASSP*, volume 4, pages 752–755, 2003.

[25] R. Wilkinson, J. Geist, S. Janet, P. Grother, C. Burges, R. Creecy, B. Hammond, J. Hull, N. Larsen, T. Vogl, and C. Wilson, editors. *1st census optical character recognition systems conf. #NISTIR 4912*, 1992.

[26] 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.