

A Writer Identification and Verification System Using HMM Based Recognizers

Andreas Schlapbach¹, Horst Bunke¹

Institut für Informatik und angewandte Mathematik
Universität Bern, Neubrückstrasse 10
CH-3012 Bern, Switzerland
e-mail: {schlpbch, bunke}@iam.unibe.ch

The date of receipt and acceptance will be inserted by the editor

Abstract In this paper, an off-line, text independent system for writer identification and verification of handwritten text lines using Hidden Markov Model (HMM) based recognizers is presented. For each writer, an individual recognizer is built and trained on text lines of that writer. This results in a number of recognizers, each of which is an expert on the handwriting of exactly one writer. In the identification and verification phase, a text line of unknown origin is presented to each of these recognizers and each one returns a transcription that includes the log-likelihood score for the generated output. These scores are sorted and the resulting ranking is used for both identification and verification. Several confidence measures are defined on this ranking. The proposed writer identification and verification system is evaluated using different experimental setups.

Key words writer identification – writer verification – off-line handwriting – HMM based handwriting recognition

1 Introduction

In recent years, significant progress has been made in recognizing a person based on biometric features [14, 13, 15]. Prominent biometric modalities include face, fingerprint, iris, signature, and voice. In this paper we address the problem of personal identification and verification based on a person's handwriting. Writer identification is the task of determining the author of a sample handwriting from a set of writers [22]. Related to this task is writer verification, i.e., the task of determining whether or not a handwritten text has been written by a certain person. If any text may be used to establish the identity of the writer the task is *text independent*. Otherwise, if a writer

In the first place it is not a great deal
 In Gettysburg He prayed that the chap
 Amen. MOST people would probably regard tiredness as a
 War is she necessarily being defeated. She really did feel tired
 One of the greatest steps forward that has been
 As it is, with so much of our life already
 This could hardly happen without the
 In this 200-fathom trench the herring do not touch the bottom.
 Easterly winds, on the other hand,
 Actually the seine net has little or no cover.

Fig. 1 Examples of text lines used for writer identification and verification.

has to write a particular predefined text – such as his or her signature – to identify himself or herself, or to verify his or her identity, the task is *text dependent*. Writer identification and verification can be performed on-line, where temporal and spatial information about the writing is available, or off-line, where only a scanned image of the handwriting is available. The system we propose in this paper performs text independent writer identification and verification using off-line handwritten text lines. Compared to other works which base their approach on either single words or on whole pages of handwritten text (see Section 2 for a discussion of related work), we use text lines as basic input units. Examples of handwritten text lines from our database, produced by different writers, are given in Fig. 1. Possible applications of our system include forensic writer identification [31], the retrieval of handwritten documents from a database [2, 3], or authorship determination of historical manuscripts.

For both isolated word and general text recognition, Hidden Markov Models (HMMs) have become the predominant approach. In fact, HMM based recognizers have a number of advantages over other approaches [9]. First, they are resistant to noise and can cope with shape variations. Secondly, HMM based recognizers are able to implicitly segment a text line into words and characters, a task that is difficult to perform explicitly [32]. Thirdly, there exist standard algorithms for training and testing [23].

This paper is based on the idea of utilizing an existing HMM-based handwriting recognition system, which has been optimized for text recognition, for the purpose of writer identification and verification. For each writer in the considered population, an individual HMM based handwriting recognition system is trained using only data from that writer. Thus for n

different writers we obtain n different HMMs. They all have the same architecture, but their parameters, i.e., transition and output probabilities, are different because they are trained on different data each. Intuitively, each HMM can be understood as an expert specialized in recognizing the handwriting of one particular person. Given an arbitrary line of text as input, each HMM based recognizer outputs a transcription of the input together with a recognition score. These outputs are sorted in decreasing order of the recognition scores, producing a ranking of all systems. Assuming that correctly recognized words have a higher score than incorrectly recognized ones, and assuming furthermore that the recognition rate of a system is higher on input from the writer the system is trained on than on input from other writers, we can utilize the scores produced by the different HMMs for the task of identifying the writer of a text line or verifying whether a text line has actually been written by the person who claims to be the writer.

This paper is structured as follows. In the next section, we present related work. In Section 3 our system for handwritten text line identification and verification is introduced. Section 4 presents experimental results on the identification as well as on the verification task, and Section 5 concludes the paper.

2 Related Work

Surveys covering work in automatic writer identification and signature verification until 1993 are given in [16, 22]. Writer identification can be understood as a classification problem where a word, text fragment, or text is to be assigned to one out of a number of possible writers. Recently, different approaches to writer identification have been proposed. Said et al. [25] treat the writer identification task as a texture analysis problem. They use global statistical features extracted from the entire image of a text using multi-channel Gabor filtering and grey-scale co-occurrence matrix techniques.

Srihari et al. [10, 34] address the problem of writer verification, i.e., the problem of determining whether two documents are written by the same person or not. In order to identify the writer of a given document, they model the problem as a classification problem with two classes, *authorship* and *non-authorship*. Given two handwriting samples, one of known and the other of unknown identity, the distance between two documents is computed. Then the distance value is used to classify the data as positive or negative.

Zois et al. [35] base their approach on single words by morphologically processing horizontal projection profiles. The projections are partitioned into a number of segments from which feature vectors are extracted. A Bayesian classifier and a neural network are then applied to the feature vectors.

In Hertel et al. [12] a system for writer identification is described. The system first segments a given text into individual text lines and then extracts a set of features from each text line. The features are subsequently used in

a k -nearest-neighbor classifier that compares the feature vector extracted from a given input text to a number of prototype vectors coming from writers with known identity.

Bulacu et al. [8] use edge-based directional probability distributions as features for the writer identification task. The authors introduce edge-hinge distribution as a new feature. The key idea behind this feature is to consider two edge fragments in the neighborhood of a pixel and compute the joint probability distribution of the orientations of the two fragments. Additionally, in [29] as a new feature the histogram of connected-component contours (CO^3) for upper-case handwriting is introduced. Combining this feature with the edge-hinge feature achieves better results than each of the features used separately. In [30] this approach is extended to mixed-style handwriting using fragmented connected-component contours (FCO^3).

In a number of papers [4,5,21] graphemes are proposed as features for describing the individual properties of handwriting. Furthermore, it is shown that each handwriting can be characterized by a set of invariant features called the writer's invariants. These invariants are detected using an automatic grapheme clustering procedure. In [6] these graphemes are used to address the writer verification task based on text blocks as well as on handwritten words.

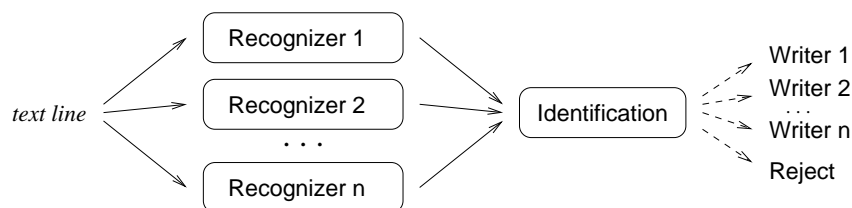
Leedham et al. [17] present a set of eleven features which can be extracted easily and used for the identification and verification of documents containing handwritten digits. These features are represented as vectors, and by using the Hamming distance measure and determining a threshold value for the intra-author variation a high degree of accuracy in authorship detection is achieved.

The idea of using HMM based recognizers for the purpose of writer identification was first proposed in [26] and a preliminary version of the paper appeared in [27] which also addresses the task of writer verification. The current paper has been significantly extended with respect to methodology and experimental evaluation. We also provide an analysis of the relationship between writer identification and text recognition rate.

3 Writer Identification and Verification Using HMM Based Recognizers

In this paper a system for the identification and verification of handwritten text lines using HMM based recognizers is presented. A schematic overview of the system is shown in Fig. 2. For each writer, a text line recognizer as described in Section 3.1 is built and trained with data coming from this writer only. As a result of the training procedure, we get a recognizer for each writer that is an expert on the handwriting style of that particular writer. Presented with a text line of unknown identity each recognizer returns a transcription containing the log-likelihood score of the text line. These scores are sorted and based on the resulting ranking writer identification (see Section 3.2) or writer verification (see Section 3.3) are performed.

Writer Identification



Writer Verification

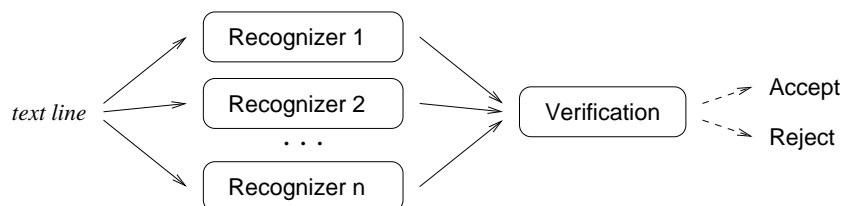


Fig. 2 Schematic overview of the writer identification and verification system.

By the end of the month he still delighted in Naples. He

Text line before normalization.

By the end of the month he still delighted in Naples. He

Text line after normalization.

Fig. 3 A text line before and after normalization.

3.1 HMM Based Recognizer

Our system uses an HMM based recognizer that has been designed and optimized for the task of handwritten text line recognition. Except for some normalization operations (see next two paragraphs), this recognizer is treated as a black box and used without any modification. For the purpose of completeness, we include a brief description of this recognizer here. More details can be found in [18].

In a first preprocessing step, the image is contrast enhanced. Next, the skew of a text line is corrected such that the text line is aligned horizontally. These normalization operations do not remove any writer specific information.

The other normalization operations we apply have been determined empirically. In a previous paper [28], we have studied the effect of normal-

ization operations on the identification rate of our system. The operations considered are slant correction, width normalization and vertical scaling. A detailed description of the normalization operations is given in [18]. On the one hand, there exists a strong correlation between the text recognition and the writer identification rate, and applying normalization operations increases the text recognition rate. On the other hand, normalization also removes writer-specific information from a handwritten text line. Hence there is a trade-off between optimizing the text recognition performance of our system and keeping writer specific features. Our experiments have shown that the highest writer identification rates are obtained if slant correction and width normalization are omitted and only vertical scaling is applied. Consequently, we perform vertical scaling, in the same way as described in [18], but do not apply slant correction and width normalization. In Fig. 3 a text line before and after normalization is shown.

A sliding window of one pixel width moves over the normalized text line from left to right and extracts nine features, three global and six local ones. The three global features are the fraction of black pixels in the window, the center of gravity and the second order moment, while the six local features represent the position and orientation of the upper- and the lower-most pixel, the number of black-to-white transitions in the window, and the fraction of black pixels between the upper- and the lower-most black pixel. Using these nine features, an input text line is converted into a sequence of 9-dimensional feature vectors.

This set of features is not inspired by forensic research but forms an abstract representation of a person's handwriting and has shown good performance in handwritten text recognition [18]. The features extracted from a text line are tested for Roman writing only. Their suitability for non Roman writing styles is an interesting, open question beyond the scope of the current paper.

The characters of the text lines are modeled using HMMs. For each upper and lower case character and for frequent punctuation marks, such as space, colon, semi-colon and full stop, an individual HMM is built. Other, infrequent punctuation marks, such as exclamation or question marks, are mapped to a special garbage model. The collective garbage model is used because even in a large set of text lines there does not exist sufficient data to train such models individually.

The individual HMMs consist of 14 states that are connected in a linear topology. The output probability distribution is modeled by a mixture of Gaussian components. The character models are concatenated to word models, which share the individual character models. This allows us to share training data across words. While our approach is text-independent, we assume that each word in the text is included in the dictionary. A complete text line is modeled by concatenating these word models. We again refer to [18] for further details.

The HMMs are trained by applying the Baum-Welch algorithm [23] using a training strategy proposed in [11]. First, a single Gaussian output

distribution is used for each state. Each model is trained with four iterations. Then in a second step, the number of Gaussian mixture components is increased. This is accomplished by splitting the Gaussian distribution with the highest weight. The mean vectors of the two new Gaussian distributions are the mean of the original Gaussian ± 0.2 times the standard deviation of the original distribution [33]. Then in the third step, we again train each model in four iterations using the new mixture components. Steps two and three are repeated until the desired number of Gaussian mixture components is reached. In the experiments described in this paper, the number of mixture components is increased up to a maximum of five Gaussians.

The system has been implemented using the HTK toolkit [33], originally developed for speech recognition. The toolkit employs the Baum-Welch algorithm for training and the Viterbi algorithm for recognition [23]. The output of the system is a transcription of the considered text line consisting of a sequence of words together with the log-likelihood score of each word. The score of a text line is the sum of the log-likelihood of all words. For both writer identification and verification only the output scores, but not the transcriptions are utilized.

3.2 Writer Identification

For the task of writer identification, we present a text line of an unknown writer to each of the trained recognizers. Each recognizer outputs a transcription of the input text line together with its log-likelihood score. The log-likelihood scores are sorted in descending order. Using a confidence measure [20] defined on these log-likelihood scores enables us to implement a rejection mechanism. If the confidence measure of a text line is above a given threshold, the system returns the identity of the text line with the highest ranked score; otherwise the system rejects the input. Thus, if we have n -writers, the writer identification problem is an n -class classification problem with a reject option.

A very simple way to define a confidence measure, $cm_1(t)$, for a text line t is to use the log-likelihood score l_1 of the first ranked text line:

$$cm_1(t) = l_1 \quad (1)$$

A more elaborate confidence measure is inspired by the cohort score normalization technique used in the field of speaker verification [1, 24]. The log-likelihood scores of the first N ranks are used to calculate the confidence measure $cm_2(t)$ of a text line t as follows:

$$cm_2(t) = \frac{l_1 - l_{avg}}{|t|} \quad (2)$$

where $|t|$ is the length of the text line t in pixels and

$$l_{avg} = \frac{1}{N} \sum_{j=2}^{N+1} l_j \quad (3)$$

In Eq. 3 the sum of log-likelihoods is formed over the log-likelihoods of the competing N -best ranked writers. The index j starts at 2 which requires that $N < n$.

3.3 Writer Verification

For the task of writer verification, the system must decide, based on some criterion, whether a text line with a claimed identity is in fact from this writer or whether it is an impostor attempt. We define the following verification criterion. If the confidence measure of a text line is above a certain threshold, we assume that the text line is in fact from the claimed writer; otherwise the input is classified as not being of the claimed identity. The writer verification problem is thus a two-class classification problem.

A very simple confidence measure is to use the log-likelihood score of the text line of the claimed identity, $l_{claimed_identity}$, and define the confidence measure as:

$$cm_3(t) = l_{claimed_identity} \quad (4)$$

A more refined confidence measure can be defined by using the log-likelihood scores of the first N ranks as follows:

$$cm_4(t) = \frac{l_{claimed_identity} - l_{avg}}{|t|} \quad (5)$$

where l_{avg} is given by

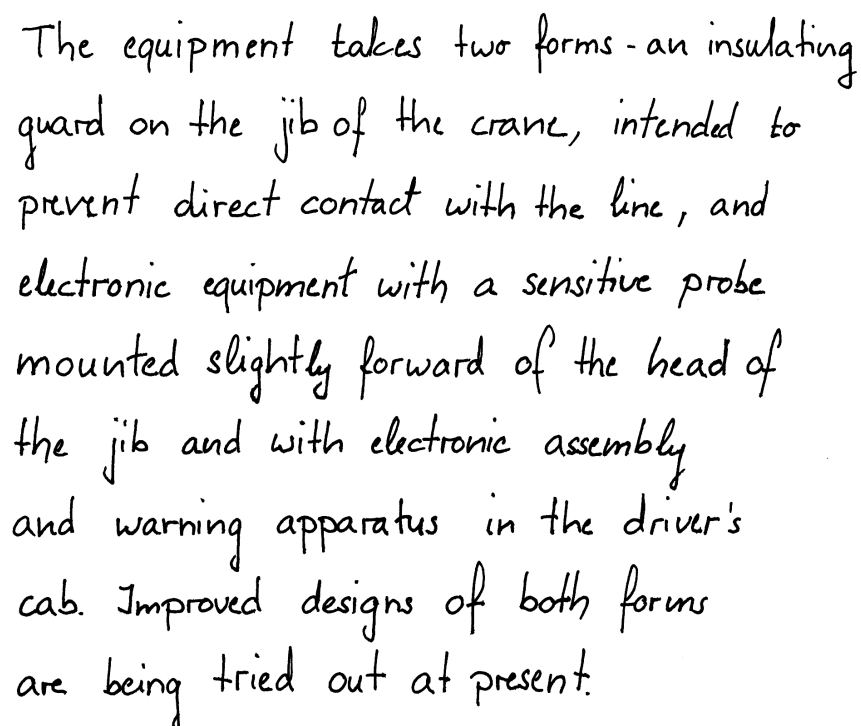
$$l_{avg} = \frac{1}{N} \sum_{\substack{j=1 \\ j \neq r(t)}}^{N+1} l_j \quad (6)$$

The confidence measure in Eq. 5 is calculated from the difference of the log-likelihood score of the claimed identity and l_{avg} , and is normalized by the length of the text line. We calculate l_{avg} based on the N -best ranked competing writers according to Eq. 6, where $r(t)$ is the rank of the claimed identity of text line t . Again, $N < n$ must hold.

4 Experimental Results

4.1 Database and Experimental Setup

A verification system must decide whether a text line with a claimed identity was in fact written by that person or not. In the former case a person is called

A photograph of a page of handwritten text in cursive. The text describes two types of crane equipment: an insulating guard on the jib and electronic equipment with a sensitive probe. The handwriting is clear and legible, with some ink bleed-through from the reverse side of the paper.

The equipment takes two forms - an insulating guard on the jib of the crane, intended to prevent direct contact with the line, and electronic equipment with a sensitive probe mounted slightly forward of the head of the jib and with electronic assembly and warning apparatus in the driver's cab. Improved designs of both forms are being tried out at present.

Fig. 4 Example of a page of handwritten text.

a *client*, in the latter case he or she is called an *impostor* [7]. Consequently, we have to define two sets. First, a client data set needs to be defined that contains text lines of writers that have to be accepted. This set is also used to test the system on the writer identification task. Second, we define an impostor data set that contains text lines of writers that have to be rejected. Impostor attempts can be divided into unskilled forgeries, where the impostor makes no effort to simulate a genuine handwriting, and skilled forgeries, where the impostor tries to imitate the handwriting of a client as closely as possible [22].

Our experiments are based on pages of handwritten text from the IAM database [19]¹. The database currently contains over 1,500 pages of handwritten text written by over 650 different writers. Each page contains between five and eleven text lines. For each writer we use five pages of text. An example of such a page is shown in Fig. 4. To evaluate the performance of our system on the writer identification and verification tasks, the experimental setup consists of text lines from 100 clients, 20 unskilled impostors, and 20 skilled impostors.

¹ The database is publicly available at: www.iam.unibe.ch/~fki/iamDB

The equipment takes two forms - an insulating guard on the jib of the crane, intended to prevent direct contact with the line, and electronic equipment with a sensitive probe mounted slightly forward of the head of the jib and with electronic assembly and warning apparatus in the driver's cab. Improved designs of both forms are being tried out at present.

Fig. 5 Skilled forgery of the handwritten text shown in Fig. 4.

The data set for the identification experiment contains 4,103 text lines from 100 different writers containing 20,391 word instances and 5,578 unique word classes.

We conducted two different writer verification experiments. The first verification experiment consists of data coming from clients and impostors using unskilled forgeries. The unskilled forgeries are obtained from the database by extracting 571 text lines produced by 20 writers. The writers of these text lines are disjoint from the 100 clients and no HMM recognizer exists that is trained on the handwriting of any of these 20 writers. Based on these text lines the impostor data set is constructed by assigning to each of these text lines seven identities of writers known to the system. The client data set is the same as the one used for the identification experiments. The rationale is that the number of text lines to be accepted is approximately the same as the number of text lines that have to be rejected. The impostor data set thus consists of $7 \times 571 = 3,997$ text lines. Overall, the complete data set consists of 8,100 text lines.

The second verification experiment is based on data coming from clients and impostors using skilled forgeries. The following protocol is used to obtain the skilled forgeries. A page of text written by an unknown author

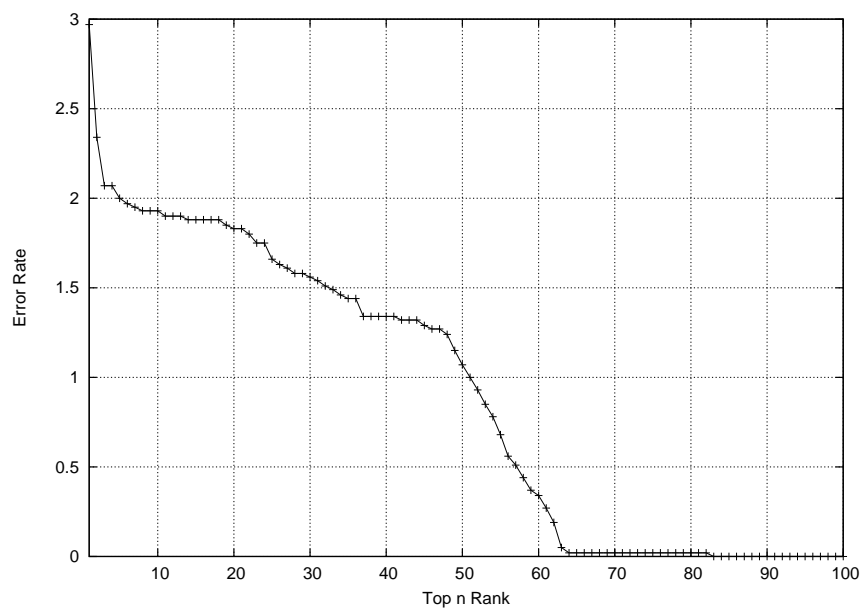


Fig. 6 N -best list for the 100 writers identification experiment.

is presented to a person. The person is then asked to study and train the writing for 15 minutes. When this time period has elapsed, the person is asked to copy the text onto a white sheet of paper where a ruler sheet is put below. An example of a skilled forgery of the text given in Fig. 4 is shown in Fig. 5.

The test set for the second verification experiment is composed of two subsets. The client data set consists of one page of text from twenty different writers. A total of 169 text lines are extracted from these twenty pages. The same twenty pages are then skillfully forged. From these forgeries, another 169 text lines are extracted. Hence, in total 338 text lines are used.

For each writer the data set is split into four disjoint subsets, which enables us to perform full four-fold cross validation experiments. Iteratively, three out of the four sets are used to train the system and the remaining set is used to test its performance. Using cross validation guarantees that the training and test sets are disjoint, that our experiments are text independent, and that the maximum possible amount of test data is used. Of course, the impostor data set is used for testing only.

4.2 Writer Identification Experiments

In this experiment, a correct writer identification rate of 97.03% is achieved. In Fig. 6 the n -best list which measures the identification rate not only based

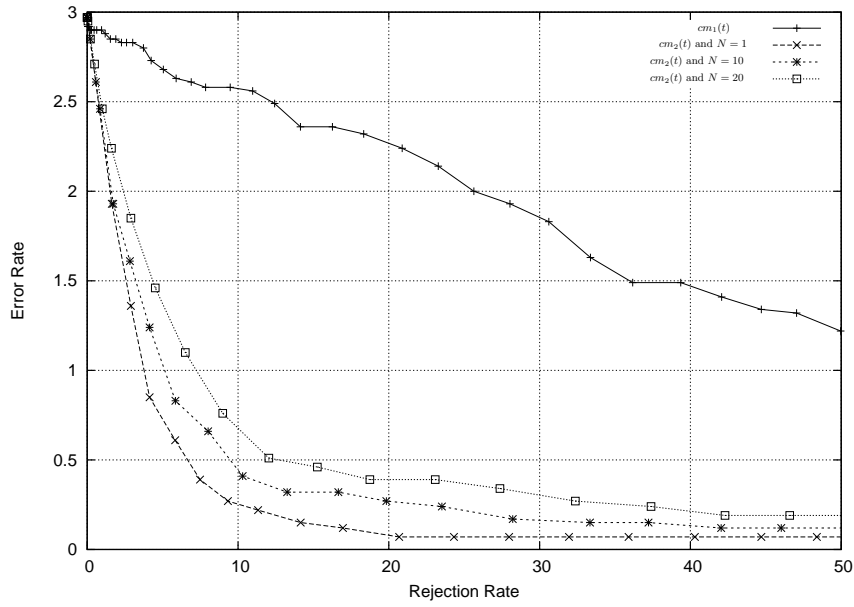


Fig. 7 Error-rejection curves for the 100 writers identification experiment.

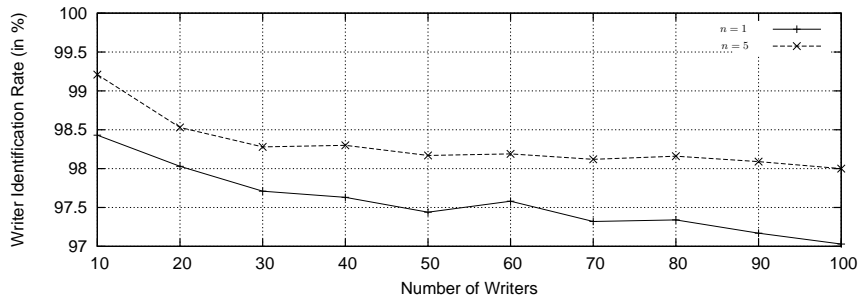


Fig. 8 Identification rate as a function of the number of writers.

on the first rank, but based on the first n ranks is shown. The error rate drops from 2.97% to below 2% if the first six ranks are considered.

We use the confidence measures introduced in Section 3.2 to reject an input in case of uncertainty and calculate the corresponding error-rejection curves. As can be seen in Fig. 7, the $cm_1(t)$ (see Eq. 1) based system performs clearly inferior to the $cm_2(t)$ (see Eq. 2) based system. The best error-rejection curve is achieved using the confidence measure $cm_2(t)$ with Eq. 3 and $N = 1$. By rejecting 4.14% of the text lines with lowest confidence score, the error rate drops below 0.85% and if 21% of the text lines with lowest confidence score are rejected, the error rate drops below 0.1%.

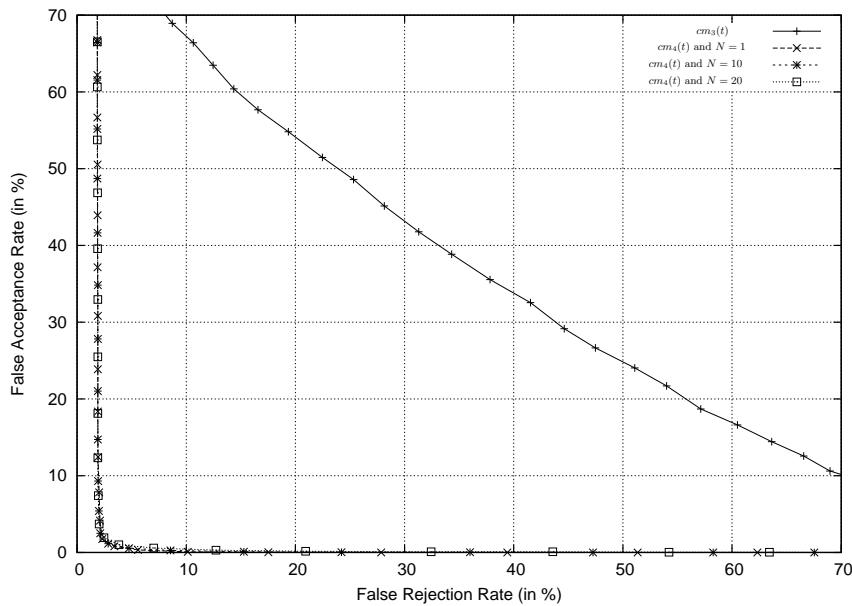


Fig. 9 ROC curves for the verification experiment using unskilled forgeries.

To investigate how our system scales with an increasing number of writers, we perform the following experiment. We start with text lines from ten writers and present them to the ten systems which are trained on these writers. We then increase the number of writers and systems by steps of ten up to hundred and measure the writer identification rate. The relationship between the identification rate and the number of writers is plotted in Fig. 8 for the case where only the best ranked system is considered ($n = 1$), and the case where the top five ranked systems are taken into account ($n = 5$). We observe that our system scales quite favorably with an increasing number of writers. For example, if the number of writers is doubled from 50 to 100 then the writer identification rate decreases by only one percent for $n = 1$ and by 0.2% for $n = 5$.

4.3 Writer Verification Experiments

A verification system can make two types of errors. It can falsely reject a text line coming from a client, or it can falsely accepted a text line coming from an impostor. The results of the verification experiments are presented in the form of Receiver Operator Characteristic (ROC) curves [7] using different confidence measures. To plot the ROC curves the minimal and the maximal confidence measure is calculated and then varied.

In the verification experiment using unskilled forgeries, the ROC curve based on $cm_3(t)$ shows considerably worse performance then the ROC curves

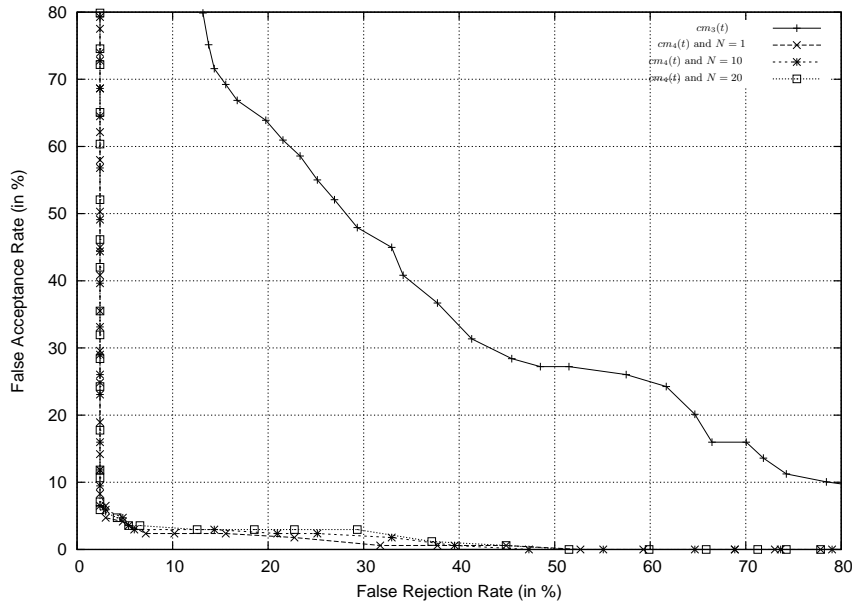


Fig. 10 ROC curves for the verification experiment using skilled forgeries.

based on $cm_4(t)$ (see Fig. 9). The best ROC curve is produced using the confidence measure based on Eqs. 5 and 6. An Equal Error Rate (EER) of about 2.0% is achieved. For $N = 1$, a False Acceptance Rate (FAR) of 0.8% is obtained at a False Rejection Rate (FRR) of 3.4%. Increasing the number of writers N to calculate the confidence measures $cm_4(t)$ does not lead to substantially different ROC curves in this experimental setup.

The ROC curves for the verification experiment using skilled forgeries are given in Fig. 10. Again, the ROC curve based on $cm_4(t)$ produces superior results compared to $cm_3(t)$. The EER is approximately 4.47%. Using $N = 1$ to calculate the confidence measure produces a slightly better ROC curve when compared to using $N = 10$ or $N = 20$.

In Fig. 11 an example of the first type of error is shown. A text line coming from the same writer as the other three text lines is falsely rejected. The second type of error is shown in Fig. 12. A text line coming from an impostor is falsely assumed to come from the same writer who has written the three other text lines. In the second case, it is difficult even for a human observer to discriminate the two handwritings.

4.4 Writer Identification and Text Recognition Rate

Our system uses HMM based handwriting recognizers which are optimized for text recognition and is based on the hypothesis that the word recognition rate is higher when the input comes from the writer whose data are used

He used to come to Pierre's Bar again and again. No he couldn't really remember him, he was one of the crowd. The

Falsely rejected text line:

sun came out; Mart went shopping; I

Fig. 11 Example of false rejection error.

If there was any truth in this, Lord Melbourne's letter of the following day, offering Huskisson the Government of Gibraltar, may have been a

Falsely accepted text line:

Later in the year, the idea of some sort of

Fig. 12 Example of false acceptance error.

	Word Recognition Rate
<i>Same</i>	63.12%
<i>Other</i>	11.38%

Table 1 Word recognition rates for 100 writers identification experiment.

to train the system. We expect a large difference in text recognition performance between the case where the training and input data come from the same writer and the case where they come from different writers. In Table 1 we distinguish between the case where training and input data come from the same writer (row *Same*) and the case where they come from different writers (row *Other*). Obviously, there is a huge difference in word recognition performance between the two cases, which confirms our hypothesis.

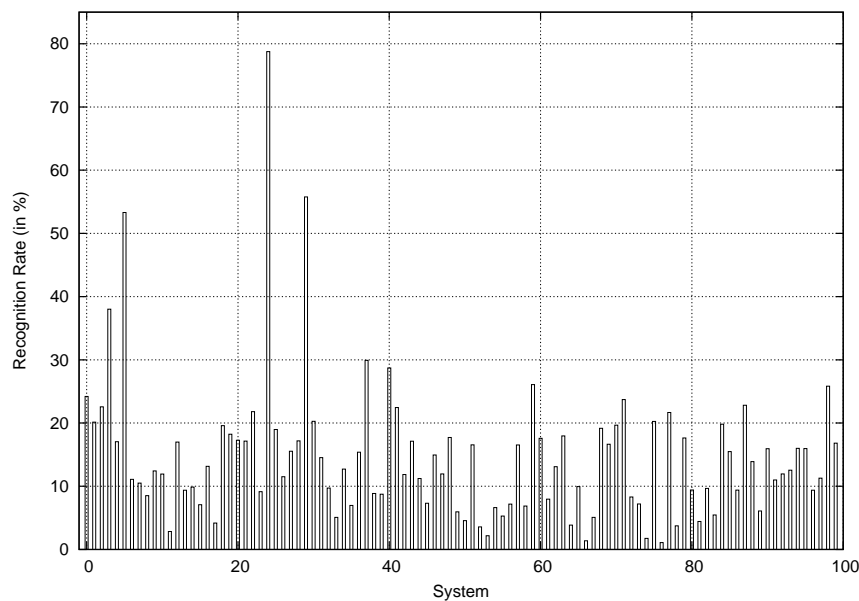


Fig. 13 Average text recognition rate for text lines coming from *Writer 24*.

In Fig. 13, the average text recognition rate per writer is shown on text lines coming from *Writer 24*. The system that is trained with text lines from *Writer 24* achieves an average text recognition rate of 78.76%. In contrast, the text recognition rates of the systems which are trained with text lines from other writers achieve an average text recognition rate of only 14.07% in average. In Fig. 14 training samples of the four systems which achieve the highest text recognition rate in Fig. 13 are shown. Visually inspecting the text lines reveals that all four handwritings are indeed similar, especially those of *Writers 5, 24, and 29*. The letters are standing upright, are of similar width, and the ascender, the middle, and the descender regions are of similar proportion. This confirms the feasibility of the approach proposed in this paper.

5 Conclusion

In this paper, we have presented a system that uses HMM based text line recognizers for the tasks of text independent off-line writer identification and verification. The basic input units presented to the system are handwritten text lines. From each text line, nine features are extracted. Using these features, we train a recognizer for each writer. A text line of unknown identity is presented to each of these recognizers. As output, each recognizer produces a transcription of the input text line with a log-likelihood score.

First ranked system (*Writer 24*)

By the end of the month he still delighted in Naples. He told Conchery that he enjoyed it as much as his health permitted him to enjoy

Second ranked system (*Writer 29*)

When the sailing season was past, he sent Pearl back to England, and returned to Rowe for the winter. In late November, he was 'suffering as usual,' but hoped, he told

Third ranked system (*Writer 5*)

A pleasant life truly!... It so happens that I have an Italian who is perhaps the best Valet de Chambre that ever was.

But he has not one word of English.' While he was

Fourth ranked system (*Writer 3*)

What infatuation! Personally I am indifferent, but I really tremble for my country! I may be mistaken, & tho' I cannot but fear that the exasperation of the People

Fig. 14 Training samples from the four systems which achieve the highest average text recognition rate for *Writer 24* (see Fig. 13).

Based on these scores a ranking in descending order is generated which is used for both identification and verification.

On the writer identification task, we achieve a correct identification rate of 97.03% in a 100 writers experiment using 4,103 lines of text. Experimenting with a set of confidence measures we show that by rejecting 4.14% of

the text lines the error rate drops below 0.85% and a recognition rate of 99.9% is achieved by rejecting 21% of the text lines with the lowest confidence measure. Regarding writer verification, our system performs very well on both tasks of accepting clients and rejecting impostors. An Equal Error Rate of 2.0% is achieved on a total of 8,100 text lines coming from 100 clients and 20 impostors using unskilled forgeries. Using skilled forgeries, the Equal Error Rate raises to 4.47% on a test set that consists of 169 text lines from 20 clients and 169 text lines from 20 impostors.

While we have shown that our system scales from $n = 10$ to $n = 100$ writers, it is an open question how it would scale for a very large number of writers, e.g., $n > 1000$. Performing such an experiment would require a significant extension of our database. One way to reduce training time and the amount of data needed from each writer is to train a general background model and then adapt this model to obtain a specific writer model. These issues are left for future work.

6 Originality and Contribution

Hidden Markov Models (HMMs) have become a standard tool in handwriting recognition. HMM based recognizers have a number of advantages over other approaches. First, they are resistant to noise and can cope with shape variations. Second, they allow to model characters of variable width occurring in the text. Third, HMM based recognizers are able to implicitly segment a text line into words and characters, a task that is difficult to perform explicitly. Last, there exist standard algorithms for training and testing.

The main contribution of this paper is to apply HMMs to the problem of off-line, text independent writer identification and verification. To the best of our knowledge, HMMs have not been applied to these tasks before. Our system is text independent, i.e., any text can be used for writer identification and verification. Using text lines as basic input units, the proposed approach is positioned between the case where only a single word is used and systems that require a whole page of text as input. Single words carry little information and corresponding systems may not scale well with an increasing number of writers. On the other hand, requiring a whole page of text being available may be too restrictive for certain applications. Consequently, the approach proposed in this paper may provide a good trade-off between scalability and performance on the one hand, and user-friendliness and flexibility on the other hand.

Experiments using text lines from 100 writers on the identification and two set of experiments on the verification task show that our approach performs very well. Possible applications of the system are in forensic writer identification, the retrieval of handwritten documents from a database, or authorship determination of historical manuscripts.

Acknowledgements This work is supported by the Swiss National Science Foundation program “Interactive Multimodal Information Management (IM2)” in the Individual Project “Access and Content Protection (ACP)”, as part of NCCR. The authors also thank Dr. Urs-Victor Marti and Dr. Matthias Zimmermann for providing the recognition system and the IAM-Database, and Dr. Simon Günter for valuable comments and advice. Furthermore, we would like to thank the anonymous reviewers for their valuable comments.

References

1. A. M. Ariyaeinia and P. Sivakumaran. Analysis and comparison of score normalization methods for text-dependent speaker verification. In *Proc. 5th European Conf. on Speech Communication and Technology*, pages 1379–1382, 1997.
2. H. S. Baird. Digital libraries and document image analysis. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 2–14, 2003.
3. H. S. Baird and V. Govindaraju, editors. *Proc. First Int. Workshop on Document Image Analysis for Libraries*. IEEE Computer Society, 2004.
4. A. Bensefia, A. Nosary, T. Paquet, and L. Heutte. Writer identification by writer’s invariants. In *Proc. Int. Workshop on Frontiers in Handwriting Recognition*, pages 274–279, 2002.
5. A. Bensefia, T. Paquet, and L. Heutte. Information retrieval based writer identification. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 946–950, 2003.
6. A. Bensefia, T. Paquet, and L. Heutte. Handwriting analysis for writer verification. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 196–201, 2004.
7. F. Bimbot and G. Chollet. Assessment of speaker verification systems. In D. Gibbon, R. Moore, and R. Winski, editors, *Handbook of Standards and Resources for Spoken Language Systems*, pages 408–480. Mouton de Gruyter, 1997.
8. M. Bulacu, L. Schomaker, and L. Vuurpijl. Writer identification using edge-based directional features. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 937–941, 2003.
9. H. Bunke. Recognition of cursive Roman handwriting – past, present and future. *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 448–461, 2003.
10. S.-H. Cha and S. Srihari. Multiple feature integration for writer verification. In *Proc. 7th Int. Workshop on Frontiers in Handwriting Recognition*, pages 333–342, 2000.
11. 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.
12. C. Hertel and H. Bunke. A set of novel features for writer identification. In J. Kittler and M. Nixon, editors, *Audio-and Video-Based Biometric Person Authentication*, pages 679–687, 2003.
13. A. Jain, L. Hong, and S. Pankanti. Biometric identification. *Communications of the ACM*, 43:91–98, 2000.
14. A. K. Jain, R. Bolle, and S. Pankanti, editors. *Biometrics – Personal Identification in Networked Society*. Springer, 2002.

15. J. Kittler and M. S. Nixon, editors. *Audio- and Video-Based Biometric Person Authentication*. Springer, 2003.
16. F. Leclerc and R. Plamondon. Automatic signature verification: The state of the art 1989–1993. In R. Plamondon, editor, *Progress in Automatic Signature Verification*, pages 13–19. World Scientific Publ. Co., 1994.
17. G. Leedham and S. Chachra. Writer identification using innovative binarised features of handwritten numerals. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, pages 413–417, 2003.
18. 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.
19. U.-V. Marti and H. Bunke. The IAM-database: An English sentence database for off-line handwriting recognition. *Int. Journal of Document Analysis and Recognition*, 5:39–46, 2002.
20. S. Marukatat, T. Artières, P. Gallinari, and B. Dorizzi. Rejection measures for handwriting sentence recognition. In *Proc. 8th Int. Conf. on Frontiers in Handwriting Recognition*, pages 25–29, 2002.
21. A. Nosary, L. Heutte, T. Paquet, and Y. Lecourtier. Defining writer’s invariants to adapt the recognition task. In *Proc. 5th Int. Conf. on Document Analysis and Recognition*, pages 765–768, 1999.
22. R. Plamondon and G. Lorette. Automatic signature verification and writer identification – the state of the art. In *Pattern Recognition*, volume 22, pages 107–131, 1989.
23. L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. In *Proc. of the IEEE*, volume 77, pages 257–285, 1989.
24. A. E. Rosenberg, J. Delong, C. H. Huang, and F. K. Soong. The use of cohort normalized scores for speaker verification. In *Proc. Int. Conf. on Spoken Language Processing*, pages 599–602, 1992.
25. H. E. S. Said, T. Tan, and K. Baker. Personal identification based on handwriting. *Pattern Recognition*, 33:149–160, 2000.
26. A. Schlapbach and H. Bunke. Off-line handwriting identification using HMM based recognizers. In *Proc. 17th Int. Conf. on Pattern Recognition*, volume 2, pages 654–658, 2004.
27. A. Schlapbach and H. Bunke. Using HMM based recognizers for writer identification and verification. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 167–172, 2004.
28. A. Schlapbach and H. Bunke. Writer identification using an HMM-based handwriting recognition system: To normalize the input or not? In *Proc. 12th Conf. of the Int. Graphonomics Society*, pages 138–142, 2005.
29. L. Schomaker and M. Bulacu. Automatic writer identification using connected-component contours and edge-based features of uppercase western script. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:787–798, 2004.
30. L. Schomaker, M. Bulacu, and K. Franke. Automatic writer identification using fragmented connected-component contours. In *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, pages 185–190, 2004.
31. S. Srihari and Z. Shi. Forensic handwritten document retrieval system. In *Proc. First Int. Workshop on Document Image Analysis for Libraries*, pages 188–194, 2004.

32. T. Steinherz, E. Rivlin, and N. Intrator. Off-line cursive script word recognition – a survey. *Int. Journal on Document Analysis and Recognition*, 2:90–110, 1999.
33. S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland. *The HTK Book*. 2002.
34. B. Zhang, S. N. Srihari, and S. Lee. Individuality of handwritten characters. In *Proc. 7th Int. Conf. on Document Analysis and Recognition*, volume 7, pages 1086–1090, 2003.
35. E. N. Zois and V. Anastassopoulos. Morphological waveform coding for writer identification. *Pattern Recognition*, 33:385–398, 2000.