

Automatic Transcription of Handwritten Medieval Documents

Andreas Fischer*, Markus Wüthrich*, Marcus Liwicki†, Volkmar Frinken*, Horst Bunke*
Gabriel Viehhauser‡, Michael Stolz‡

*Institute of Computer Science and Applied Mathematics, University of Bern, Neubrückestrasse 10, CH-3012 Bern
Email: {afischer, mwuethri, frinken, bunke}@iam.unibe.ch

†German Research Center for Artificial Intelligence (DFKI), Trippstadter Strasse 122, D-67663 Kaiserslautern
Email: marcus.liwicki@dfki.de

‡Institut für Germanistik, University of Bern, Länggassstrasse 49, CH-3012 Bern
Email: {viehhauser, michael.stolz}@germ.unibe.ch

Abstract—The automatic transcription of historical documents is vital for the creation of digital libraries. In order to make images of valuable old documents amenable to browsing, a transcription of high accuracy is needed. In this paper, two state-of-the-art recognizers originally developed for modern scripts are applied to medieval documents. The first is based on Hidden Markov Models and the second uses a Neural Network with a bidirectional Long Short-Term Memory. On a dataset of word images extracted from a medieval manuscript of the 13th century, written in Middle High German by several writers, it is demonstrated that a word accuracy of 93.32% is achievable. This is far above the word accuracy of 77.12% achieved with the same recognizers for unconstrained modern scripts written in English. These results encourage the development of real world systems for automatic transcription of historical documents with a view to image and text browsing in digital libraries.

I. INTRODUCTION

Handwriting recognition of scanned or photographed text images is still a widely unsolved problem in computer science and an active area of research. In particular, the interest in the recognition of handwritten text in historical documents has grown strongly in recent years [1]. In order to preserve valuable old documents, a huge number of original scripts have been digitized. Examples include writings of famous presidents, e.g., George Washington's papers at the Library of Congress¹, or scientists, e.g., Sir Isaac Newton's writings at the University of Cambridge Library². As a consequence of the creation of these large collections, there is an increasing demand from researchers in history, philology and related fields as well as from laypersons to access the document images in digital libraries [2]. Therefore, a content based indexing is required to search and browse the images. Automatic recognition and transcription of the handwriting are needed to create the corresponding indices.

The automatic reading of historical handwritten documents is an *off-line* task, where only the images of the documents are available. This task is considered to be harder than *on-line* recognition, where also temporal information can be

exploited [3]. For an up to date survey of approaches to off-line handwriting recognition, see [4]. Commercial systems with a high recognition accuracy are available only for restricted domains with modern scripts, e.g., for postal address [5] and bank check reading [6]. In case of unconstrained handwriting recognition, the recognition accuracy is far from being perfect and only few systems exist that can cope with the high variety of writing styles and the extensive vocabulary occurring in real world documents. A widely known type of recognizer suited for the task of handwriting recognition is Hidden Markov Models (HMM), which has been used, e.g., in [7] for modern scripts. Recently, an alternative to HMMs has been introduced in [8]. This novel type of recognizer is based on Neural Networks (NN) with a bidirectional Long Short-Term Memory (BLSTM). Although it has been used for on-line handwriting recognition originally, it is also applicable to off-line recognition.

For the analysis and recognition of historical documents, only little work has been reported in the literature so far. In [9], a survey of text line extraction methods is presented. This important image preprocessing step is complicated by adverse artifacts caused by the ravages of time, resulting in damaged paper or parchment and ink bleed-through. In [10], the issue of text line segmentation into single words has been discussed. Avoiding a complete transcription, word spotting has been proposed in [11] in order to efficiently match keywords against the document images. Also, computer aided manual transcription has been attempted, e.g., in [12]. In [13], documents from a single writer, i.e., letters from George Washington, are transcribed with an HMM-based recognizer and so-called alphabet-soups. A very general approach is presented in [14], where an HMM-based recognizer similar to the system in [7] is applied to medieval documents. Taking into account the small amount of manuscript data, the important issue of language modeling is addressed.

Currently, the key question for many researchers, archivists, and developers of digital libraries is: How accurate is an automatic transcription of historical documents? While the question is difficult to answer in general, we intend to clarify

¹<http://lcweb2.loc.gov/ammem/gwhtml/gwhome.html>

²<http://www.lib.cam.ac.uk/MSS/Newton.html>

the matter in this paper for medieval documents written with ink on parchment. On the one hand, handwriting recognition is difficult in this case, because the parchment may be damaged, the ink may have faded over time, and ink bleed-through may occur. Furthermore, old languages are understood only by experts in linguistics. Thus, manual transcriptions necessary to train a recognition system are hard to obtain and language models needed for whole sentence recognition are hard to build. On the other hand, the writing style is often more regular than in modern handwriting and single word images may be easier to extract from the document images. Therefore, a high recognition accuracy seems possible for single word recognition in well preserved manuscripts, provided that enough manual transcriptions are available to train a recognition system.

In the present paper, we apply two recognizers originally developed for modern scripts to medieval documents. The first recognizer is derived from the HMM-based system presented in [7] and the second recognizer is a modified version of the NN-based system proposed in [8]. Both systems can be considered as state-of-the-art recognizers for modern scripts. The underlying dataset consists of word images taken from the Parzival database presented in [14] that contains medieval documents originating in the 13th century written in Middle High German by multiple authors. The results of Medieval handwritten text recognition are compared with the accuracy achieved for unconstrained modern scripts, i.e., isolated words taken from the IAM database presented in [15]. For both datasets, a perfectly correct segmentation of the document images into single words is assumed. Solely relying on a dictionary and not using any additional language information (such as n-gram language models), it turns out that a word accuracy of 93.32% can be reached for the medieval documents, which is far above the achieved word recognition accuracy of 77.12% for unconstrained modern scripts.

The remainder of this paper is organized as follows. Section II describes the Parzival database and the preprocessing of the document images necessary for recognition, Section III introduces the HMM-based and NN-based recognizers, Section IV discusses the experimental results, and Section V draws some conclusions.

II. PARZIVAL DATABASE

In this paper, we use the Parzival database presented in [14] for our experimental evaluation. This database contains digital images of medieval manuscripts originating in the 13th century. Arranged in 16 books, the epic poem *Parzival* by Wolfram von Eschenbach was written down in Middle High German with ink on parchment. There exist multiple manuscripts of the poem that differ in writing style and dialect of the language. The manuscript used for experimental evaluation is *St. Gall, collegiate library, cod. 857* that is written by multiple authors. Figure 1 shows an example page with two columns of pairwise rhyming lines.

Transcriptions are available for 45 out of 318 folios (sheets). Altogether, 4,478 lines of text were transcribed by the Ger-

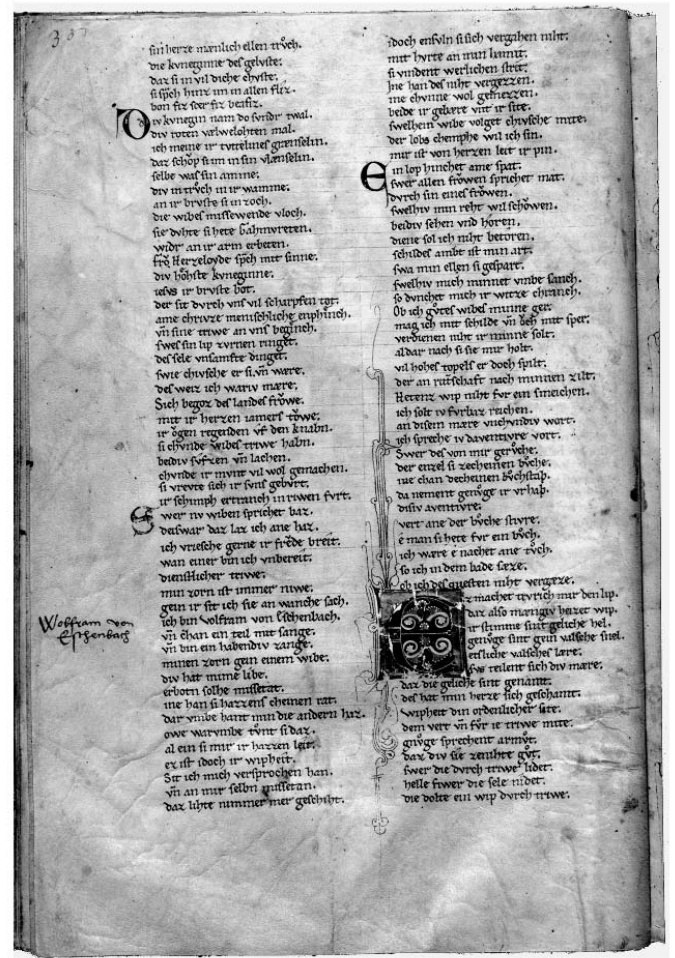


Fig. 1: Example page from the Parzival database. St. Gall, collegiate library, cod. 857, page 36.

man Language Institute of the University of Bern using the *TUSTEP*³ tool for managing transcriptions for Latin and non-Latin manuscripts. Note that this transcription is the basis for the recognition systems presented in Section III.

A. Image Preprocessing

In order to automatically transcribe medieval documents, we pursue the approach of single word recognition. This requires several preprocessing steps to isolate the handwritten words from the document images. Since the focus of this paper is on the recognition system, some of these preprocessing steps are performed in an interactive manner. In the following, these steps are discussed in greater detail.

In the first preprocessing step, decorated initial letters are manually removed from the document image. Three examples of such initial letters can be seen in Figure 1 at the left side of the columns. In the lower half of the right column, another example of a richly ornamented initial letter is found. Furthermore, annotations like the handwritten note on the left

³<http://www.zdv.uni-tuebingen.de/tustep/>

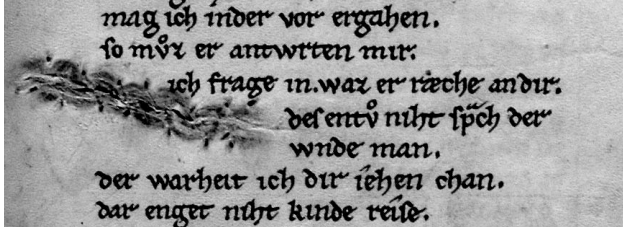


Fig. 2: Example of a seam resulting from damaged parchment being stitched together. St. Gall, collegiate library, cod. 857, page 145.

margin of the page, which were added later to the original manuscript, are removed as well as artifacts caused by the age of the parchment including holes and seams. Seams occur because parchment was expensive in medieval times and tears were stitched together. An example can be seen in Figure 2. The number of decorated initial letters, annotations, holes, and seams is rather small and their removal simplifies the subsequent preprocessing steps. The affected letters are not taken into account in the recognition experiment.

Next, the two columns of a page are separated and the colored background is removed with a Difference of Gaussian (DoG) edge detection method [16]. A binarization operation is applied to the resulting grayscale image to eliminate the remaining background noise. From the single columns, text lines are extracted with a line segmentation method based on dynamic programming [17]. The highly accurate text line segmentation is manually corrected if necessary. As proposed in [7], the resulting text lines are normalized in order to cope with different writing styles. First the skew, i.e., the inclination of the text line, is corrected. Then vertical scaling is applied with respect to the upper and lower baseline. Next, a horizontal scaling operation is performed using the mean distance of black-white transitions in a text line.

In a final step, single words are extracted from the normalized text lines using the procedure proposed in [18]. Under this procedure, an HMM-based recognizer is applied in forced alignment mode to separate the individual words based on the correct transcriptions. Again, this highly accurate word segmentation is manually corrected whenever an error occurs. Of course, since a fully automatic system does not have access to the correct transcription, other methods would be required, e.g., the word segmentation approach presented in [10]. Figure 3 shows examples of word images used in the recognition experiments after document image preprocessing.

III. RECOGNITION SYSTEMS

For the recognition of cursive handwritten text, the intrinsic difficulty arises from the fact that the letters in a word are connected and can not be segmented reliably before recognition. On the other hand, the recognition of the letters requires a segmentation. This “chicken-and-egg” problem is also known

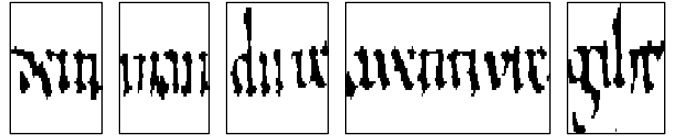


Fig. 3: Example word images from the Parzival database after preprocessing: “dem man dirre aventivre giht”.

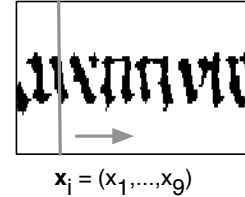


Fig. 4: Sliding window feature extraction.

as Sayre’s Paradox [4]. In this paper, two segmentation-free state-of-the-art recognition systems are used that are able to recognize handwritten words without the need of segmenting them into individual letters prior to recognition.

In the following, Section III-A describes the extraction of features from individual words, Section III-B discusses the HMM-based recognizer, and Section III-C presents the NN-based recognizer.

A. Feature Extraction

The patterns of interest for cursive handwriting recognition are the single letters of a word. In the present paper, we apply a sliding window approach [7] to the word images in order to obtain a representation of these single letters suitable for algorithmic processing. Here, unlike standard pattern representation, a letter is not represented by a single, n -dimensional, real-valued feature vector $\mathbf{x} \in \mathbb{R}^n$, because the corresponding letter image can not be extracted unambiguously from the word image before recognition. Instead, a sequence of N feature vectors $\mathbf{x}_1, \dots, \mathbf{x}_N$ with $\mathbf{x}_i \in \mathbb{R}^n$ is extracted from the word images using a sliding window moving from the left to the right over the word image. Based on this representation of the handwriting, the single letters are then recognized as contiguous subsequences of this feature vector sequence.

We use a set of nine geometrical features for medieval handwriting recognition that was proposed in [7] for modern scripts. A sliding window with width of one pixel and height of the image moves from left to right over the word and at each step, a nine-dimensional feature vector $\mathbf{x}_i \in \mathbb{R}^9$ is calculated from the corresponding pixels as illustrated in Figure 4. Three global features capture the fraction of black pixels, the center of gravity, and the second order moment. The remaining six local features consist of the position of the upper and lower contour, the gradient of the contours, the number of black-white transitions, and the fraction of black pixels between the contours.

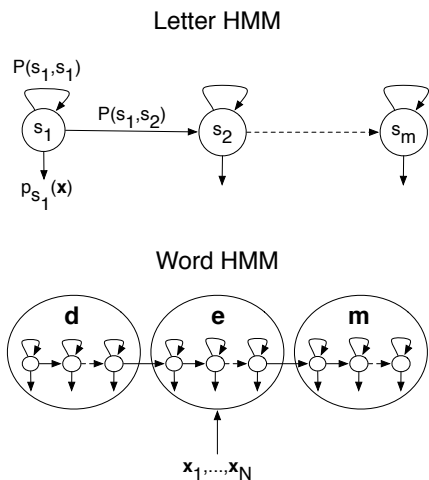


Fig. 5: Hidden Markov models.

Note that the normalization of the word images in terms of horizontal and vertical scaling mentioned in Section II-A is necessary to obtain normalized feature vectors for different writing styles. The correction of the skew, i.e., the inclination of the base line, is necessary to ensure a proper horizontal movement of the sliding window.

B. HMM Recognizer

The Hidden Markov Model (HMM) recognizer used in this paper is similar to the one presented in [7]. The recognizer is based on letter models with a certain number m of hidden states s_1, \dots, s_m arranged in a linear topology that can then be connected to meaningful words. An illustration of a single letter HMM is given in Figure 5 (top). The states s_j with $1 \leq j \leq m$ emit observable feature vectors $\mathbf{x} \in \mathbb{R}^n$ with output probability distributions $p_{s_j}(\mathbf{x})$ given by a mixture of Gaussians. Starting from the first state s_1 , the model either rests in a state or changes to the next state with transition probabilities $P(s_j, s_j)$ and $P(s_j, s_{j+1})$, respectively.

During training of the recognizer, word HMMs are built by concatenating single letter HMMs as illustrated in Figure 5 (bottom) for the word “dem”. The probability of a word HMM to emit the observed feature vector sequence $\mathbf{x}_1, \dots, \mathbf{x}_N$ is then maximized by iteratively adapting the initial output probability distributions $p_{s_j}(\mathbf{x})$ and the transition probabilities $P(s_j, s_j)$ and $P(s_j, s_{j+1})$ with the Baum-Welch algorithm [19]. The trained letter HMMs can be used afterwards to recognize arbitrary new words.

For isolated word recognition, each possible word is modeled by an HMM built from the trained letter HMMs, and the most probable word is chosen with the Viterbi algorithm [19] for an unknown input word image. In this paper, we follow a closed vocabulary assumption by taking only words into account that exist in the dataset and treat them with equal probability, i.e., we do not use a language model. Language models would treat words differently based on their frequency

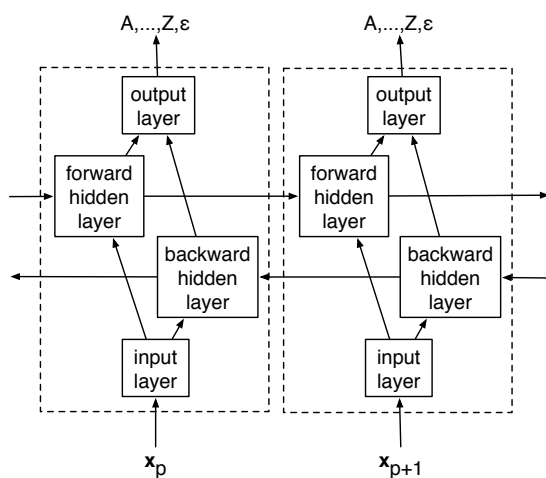


Fig. 6: Neural Network architecture.

of occurrence. For Medieval languages, no large corpora exist and language modeling becomes an issue [14].

Important parameters of the HMM recognizer are the number of states for each letter and the number of Gaussian mixtures of the output probability distributions. They can be optimized with respect to the recognition accuracy of independent validation samples as discussed in Section IV.

C. NN Recognizer

The second recognizer used in this paper is a recently developed recurrent Neural Network (NN), termed bidirectional long short-term memory (BLSTM) Neural Network [8]. Each hidden layer of the NN is made up of so called Long Short-Term Memory blocks instead of simple nodes. These memory blocks are designed specifically to address the vanishing gradient problem that describes the exponential increase or decay of values as they cycle through recurrent network layers. This is done by nodes that control the information flow in and out of each memory block. For details about BLSTM networks, we refer to [8].

An illustration of the network architecture is given in Figure 6. The input layer contains one node for each one of the nine geometrical features and is connected with two distinct recurrent hidden layers. Both hidden layers are in turn connected to a single output layer. The network is bidirectional, i.e., the feature vector sequence is fed into the network in both the forward and the backward mode. At each position p of the input sequence $\mathbf{x}_1, \dots, \mathbf{x}_N$, the output layer sums up the values coming from the forward hidden layer that has processed the feature vectors \mathbf{x}_1 to \mathbf{x}_p and the backward hidden layer that has processed feature vectors \mathbf{x}_N down to \mathbf{x}_p . The output layer contains one node for each possible letter in the sequence plus a special ϵ node to indicate “no letter”.

At each position, the output activations of the nodes are normalized so that they sum up to 1 and are treated as probabilities that the node’s corresponding letter actually occurs at this position. The Connectionist Temporal Classification Token

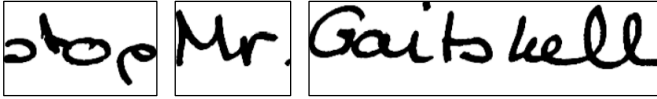


Fig. 7: Example word images from the IAM database after preprocessing: “stop Mr. Gaitskell”.

Passing algorithm [20] is then used to compute the probability of a given word. The algorithm proceeds by constructing a path through the output sequence, such that the nodes along this path yield the word’s letters. Dynamic programming is used to find the path with the highest probability.

IV. EXPERIMENTAL RESULTS

We have tested automatic recognition of medieval documents for isolated words from the Parzival database described in Section II and compared the results with the word accuracy achieved for modern scripts taken from the widely used IAM database [15]. Two state-of-the-art recognition systems for handwriting recognition are employed, an HMM-based system discussed in Section III-B and an NN-based system introduced in Section III-C.

In the following, a description of the IAM database and the characteristics of the datasets used are given in Section IV-A, the experimental setup is outlined in Section IV-B, and the results are presented in Section IV-C.

A. Datasets

Besides the Parzival database, the publicly available IAM database presented in [15] is considered for comparison. It contains modern English handwritings from several hundred writers. The word images are normalized in the same manner as the Parzival word images (see Section II-A) except for an additional slant correction. The slant (inclination) of the individual letters is normalized by means of a shearing operation such that letters are in an upright position, in order to cope with the high variability observed for unconstrained modern handwriting. Figure 7 shows three example word images from the IAM database after preprocessing.

In order to keep the computational effort within reasonable bounds, a subset of the Parzival word database is considered for the experimental evaluation, i.e., each second word. Similarly, a subset of about the same size is chosen from the IAM word database. Table I summarizes the main statistics of the datasets.

B. Experimental Setup

The recognition experiment is conducted for both datasets in the same manner. First, the set of all words is divided into a distinct training, validation, and test set. Half of the words, i.e., each second word, is used for training and a quarter of the words for validation and testing, respectively.

For the HMM-based recognizer, the optimal number of states for each letter is adopted from previous work [14]. In [14], the number of states was optimized with respect to

Dataset	Word Instances	Word Classes	Letters
Parzival	11,743	3,177	87
IAM	12,265	2,460	66

TABLE I: Statistics of the datasets.

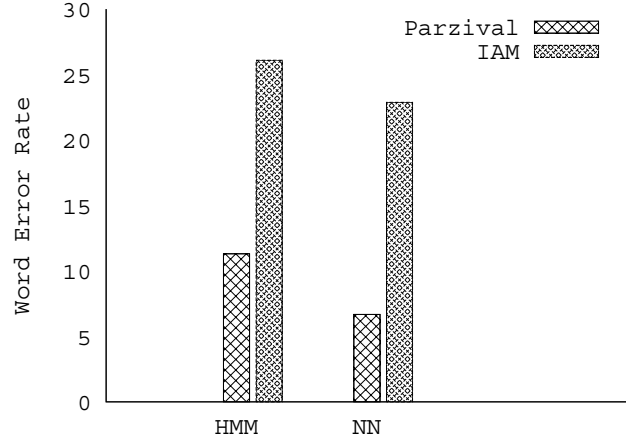


Fig. 8: Experimental results in terms of word error rate.

the mean width of the letters as proposed in [21]. The number of Gaussian mixtures for the HMM output probabilities is optimized with respect to the word accuracy on the validation set over a range from 1 to 30. The number of Gaussians is increased incrementally and before each increase, 4 training iterations are performed. The HTK toolkit⁴ for the implementation of the Baum-Welch and Viterbi algorithm is applied.

For the NN-based recognizer, we use the network architecture described in Section III-C that has proven successful in previous work [8] for modern scripts. 50 nodes are used for the forward and the backward hidden layers, each.

C. Results

The results of the experimental evaluation are shown in Figure 8 in terms of word error rate (i.e. the percentage of the words from the test set that were not recognized correctly).

With the HMM-based recognizer, a word accuracy of 88.69% is achieved for the Parzival dataset. This is far above the achieved word accuracy of 73.91% for the IAM dataset. The word error rate of 11.31% for the medieval documents is less than half of the error rate of 26.09% for modern scripts. This improvement is statistically significant (z-test, $\alpha = 0.01$). The optimal number of Gaussian mixtures found with respect to the validation set accuracy was 7 for the Parzival dataset and 16 for the IAM dataset.

With the NN-based recognizer, a word accuracy of 93.32% is achieved for the Parzival dataset, which is statistically significantly better than the word accuracy of 77.12% achieved for the IAM dataset. Here, the word error rate of 6.68% for

⁴<http://htk.eng.cam.ac.uk/>

the medieval documents is less than a third of the word error rate of 22.88% for modern scripts.

For both datasets, the NN recognizer outperforms the HMM recognizer with statistical significance (t-test, $\alpha = 0.01$).

V. CONCLUSIONS

Two state-of-the art recognizers originally developed for modern scripts, i.e., an HMM-based and an NN-based recognizer, were applied to isolated handwritten words from the Parzival database that contains medieval documents originating in the 13th century. Both word recognizers are based on nine geometric features and a Middle High German vocabulary, without taking any language models into account. The results were compared with the word accuracy achieved for unconstrained modern scripts from the IAM database.

The best recognition rate of 93.32% for the medieval Parzival dataset was achieved with the NN-based recognizer. With the HMM-based recognizer, 88.69% of the words were correctly recognized. Both results are statistically significantly higher than the word accuracies of 77.12% and 73.91% for the modern IAM dataset achieved with the same NN-based and HMM-based recognizer, respectively. The NN-based recognizer with a bidirectional Long Short-Term Memory architecture outperformed the HMM-based recognizer with statistical significance for both datasets.

The results indicate that the handwriting recognition of medieval documents can be considered easier than the task of unconstrained handwriting recognition of modern scripts (if the medieval documents are similar to the Parzival manuscripts). This is due to the fact that the medieval handwriting style often is much more regular when compared to modern handwriting. Based on a word error rate of less than 7%, one expects that systems can be developed to automatically provide transcriptions of medieval documents with an acceptable quality for keyword search in digital libraries.

Note that the high word accuracy assumes a perfectly correct segmentation of the document image into single words. Hence, fully automatic systems would require highly accurate image preprocessing procedures to obtain comparable results. Alternatively, computer aided systems could be developed that are based on manual or interactive segmentation. While the manual transcription of medieval documents requires linguistic expertise, a manual segmentation of the document image into single words does not require special skills. Therefore, such computer aided systems could significantly reduce the effort of manual transcription.

Future work for further improving the automatic transcription of historical documents includes a combination of the HMM-based and NN-based recognizers in order to create a multiple classifier system [22].

ACKNOWLEDGMENTS

This work has been supported by the Swiss National Science Foundation (Project CRSI22_125220/1) and by the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2).

Furthermore, we thank Alex Graves of the TU Munich for providing us with an implementation of the neural networks.

REFERENCES

- [1] A. Antonacopoulos and A. Downton, "Special issue on the analysis of historical documents," *Int. Journal on Document Analysis and Recognition*, vol. 9, no. 2-4, pp. 75-77, 2007.
- [2] *Second International Workshop on Document Image Analysis for Libraries (DIAL 2006)*. IEEE, 2006.
- [3] R. Plamondon and S. Srihari, "Online and off-line handwriting recognition: A comprehensive survey," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 63-84, 2000.
- [4] H. Bunke and T. Varga, "Off-line Roman cursive handwriting recognition," in *Digital Document Processing: Major Directions and Recent Advances*, ser. Advances in Pattern Recognition, B. Chaudhuri, Ed. Springer, 2007, vol. 20, pp. 165-173.
- [5] S. Srihari, Y. Shin, and V. Ramanaprasad, "A system to read names and addresses on tax forms," in *Proc. IEEE*, vol. 84, no. 7, 1996, pp. 1038-1049.
- [6] N. Gorski, V. Anisimov, E. Augustin, O. Baret, and S. Maximor, "Industrial bank check processing: the A2iA check reader," *Int. Journal on Document Analysis and Recognition*, vol. 3, pp. 196-206, 2001.
- [7] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system," *Journal of Pattern Recognition and Art. Intelligence* 15, vol. 15, pp. 65-90, 2001.
- [8] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for improved unconstrained handwriting recognition," *IEEE Trans. PAMI*, in print.
- [9] L. Likforman-Sulem, A. Zahour, and B. Taconet, "Text line segmentation of historical documents: a survey," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 9, no. 2-4, pp. 123-138, 2007.
- [10] R. Manmatha and J. L. Rothfeder, "A scale space approach for automatically segmenting words from historical handwritten documents," *IEEE Trans. PAMI*, vol. 27, no. 8, pp. 1212-1225, 2005.
- [11] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 9, no. 2-4, pp. 139-152, 2007.
- [12] F. L. Bourgeois and H. Emptoz, "Deborah: Digital access to books of the renaissance," *International Journal on Document Analysis and Recognition (IJ DAR)*, vol. 9, no. 2-4, pp. 193-221, 2007.
- [13] S. Feng, N. Howe, and R. Manmatha, "A hidden markov model for alphabet-soup word recognition," in *Proc. IEEE Int. Conf. on Frontiers in Handwriting Recognition (ICFHR 2008)*, 2008.
- [14] M. Wüthrich, M. Liwicki, A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, and M. Stolz, "Language model integration for the recognition of handwritten medieval documents," in *Accepted for publication in Proc. IEEE Int. Conf. on Document Analysis and Recognition*, 2009.
- [15] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for off-line handwriting recognition," *Int. Journal on Document Analysis and Recognition*, vol. 5, pp. 39-46, 2002.
- [16] D. Marr and E. Hildreth, "Theory of edge detection," *Proceedings of the Royal Society of London. Series B, Biological Sciences*, vol. 207, no. 1167, pp. 187-217, 1980.
- [17] M. Liwicki, E. Indermühle, and H. Bunke, "On-line handwritten text line detection using dynamic programming," in *Proc. 9th Int. Conf. on Document Analysis and Recognition*, 2007, pp. 447-451.
- [18] M. Zimmermann and H. Bunke, "Automatic segmentation of the IAM off-line database for handwritten English text," in *Proc. 16th Int. Conf. on Pattern Recognition*, vol. 4, 2002, pp. 35-39.
- [19] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257-285, Feb. 1989.
- [20] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequential Data with Recurrent Neural Networks," in *23rd Int. Conf. on Machine Learning*, 2006, pp. 369-376.
- [21] S. Günter and H. Bunke, "Optimizing the number of states, training iterations, Gaussians in an HMM-based handwritten word recognizer," in *7th Int. Conf. on Document Analysis and Recognition*, vol. 1, 2003, pp. 472-476.
- [22] M. Haindl, J. Kittler, and F. Roli, Eds., *7th Int. Workshop on Multiple Classifier Systems (MCS 2007)*, ser. LNCS 4472. Springer, 2007.