Lexicon-Free Handwritten Word Spotting Using Character HMMs

Andreas Fischer*, Andreas Keller, Volkmar Frinken, Horst Bunke

University of Bern, Institute of Computer Science and Applied Mathematics, Neubrückstrasse 10, 3012 Bern, Switzerland

Abstract

For retrieving keywords from scanned handwritten documents, we present a word spotting system that is based on character Hidden Markov Models. In an efficient lexicon-free approach, arbitrary keywords can be spotted without pre-segmenting text lines into words. For a multi-writer scenario on the IAM off-line database as well as for two single writer scenarios on historical data sets, it is shown that the proposed learning-based system outperforms a standard template matching method.

Keywords: Handwriting recognition, Keyword spotting, Hidden Markov models

1. Introduction

Handwriting recognition of scanned or photographed text images is still a widely unsolved problem in pattern recognition, although it has been an active area of research for several decades [1]. The automatic recognition of handwritten text images is an *offline* task that is considered to be harder than *online* recognition, where temporal information can be exploited [2]. For large vocabularies and different writing styles in general [3], and for degraded historical manuscripts in particular [4], the accuracy of an automatic transcription is far from being perfect. Under these conditions, word spotting has been proposed instead of a complete transcription for the restricted task of retrieving keywords from document images [5].

Handwritten word spotting is of great interest in different application areas. For modern handwriting, an important application is given by automatic mail sorting. Nowadays, large companies still receive a high volume of handwritten correspondence. One might be interested, for example, to give more priority to mails containing the word "urgent" [6]. For historical documents, a key application is given by integrating handwritten documents in digital libraries [7]. In the context of cultural heritage preservation, many libraries all around the world have digitized their most valuable old handwritings, ranging from religious Old Greek manuscripts to handwritings from Modern Ages, e.g., George Washington's papers at the Library of Congress. Word spotting can be used for indexing the vast amount of available document images, in order to make them amenable to searching and browsing [8].

Two different approaches to handwritten word spotting can be distinguished in the literature. On the one hand, *template-based* methods [9, 10, 11, 12, 13, 14, 15, 16, 17, 18] match a query word image with labeled keyword template images. An advantage of this approach is that template images are rather easy to obtain, even if the underlying language and its alphabet are unknown. However, such systems are limited by the fact that for each possible keyword that is to be spotted, at least one template image is needed and unknown out-of-vocabulary words can not be spotted at all. Also, they typically have a low generalization capability to unknown writing styles.

On the other hand, *learning-based* methods [6, 19, 20, 21, 22, 23, 24, 25] employ statistical learning methods to train a keyword model that is then used to score query images. A very general approach was recently introduced in [6], where the learning-based approach is applied at word level based on Hidden Markov Models (HMMs). The trained word models are expected to show a better generalization capability than template images. However, the word models still need a considerable amount of templates for training and the system is not able to spot out-of-vocabulary keywords.

When the learning-based approach is applied at character level, a word spotting system obtains, in principle, the capability to spot arbitrary keywords by concatenating the character models appropriately. Although this approach is well-known for the recognition of speech [26] and poorly printed documents [27], only few reports can be found in the handwriting recognition literature. While an earlier approach presented in [21] was based on a small set of manually extracted letter-templates, there is a recent tendency towards fully-fledged learning-based systems at character level [24, 25].

In the present paper, we propose a learning-based word spotting system for unconstrained handwritten text that is based on character HMMs. When compared with previous word spotting systems, the proposed approach has several advantages. First, the system is segmentation-free at the training stage as well as at the recognition stage, i.e., it is not dependent on a segmentation of text lines into words that can be prone to errors. An advantage over learning-based systems at word level is given by the fact that only a small number of character classes needs to be trained for which a rather large number of training samples is available in a given handwritten text. Furthermore, the proposed system is able to spot arbitrary keywords that are not

*Corresponding author. Fax: +41 31 631 32 62
Email address: afischer@iam.unibe.ch (Andreas Fischer)

required to be known at the training stage. When compared with transcription-based systems, the proposed word spotting approach has the advantage that no lexicon is used. For a large vocabulary task, such a lexicon imposes a high computational complexity.

A known disadvantage of the proposed system is, however, that it requires a set of transcribed text line images for training that may be costly to obtain, in particular for historical documents. If neither the language nor the alphabet of a historical document are known, template-based image matching might be the only option available.

In an experimental evaluation, we have tested the proposed system on three data sets that represent different word spotting conditions. First, the system was tested on the IAM off-line database that contains modern English texts written by several hundred writers. The main difficulty with this database is to spot keywords in unknown handwriting styles, taking into account a large amount of training data from other writers. Secondly, tests were conducted on the George Washington database that includes a small collection of handwritten historical letters sharing a common writing style. Here, the main difficulty is to cope with the small amount of training data available. Finally, the system was tested on the Parzival database that contains a larger collection of medieval manuscripts with a common writing style. Taking into account a rather large amount of training data, the best word spotting conditions are met for this database.

The proposed system is compared with a well-established template matching method based on Dynamic Time Warping (DTW) [15]. It is demonstrated that the proposed learning-based approach outperforms the reference system not only for the multi-writer case, as expected, but also for both single-writer scenarios.

The remainder of the paper is organized as follows. In Section 2, the proposed word spotting system is presented in detail. Section 3 introduces the reference system that is used for experimental evaluation. Next, Section 4 describes the experimental setup. In particular, the data sets are discussed in Sections 4.1–4.3. Results and discussion are given in Section 5 and finally, conclusions are drawn in Section 6.

1.1. Related Work

Keyword spotting has been applied to speech [28, 26] and poorly printed documents [27, 29] before. For handwritten text, it was proposed a few years later in [9]. In a template-based approach, the Scott and Longuet-Higgins distance (SLH) [30] was used in [9] to compare keyword template images and unknown word images. In the following, other features based on global image characteristics have been proposed, e.g., gradient, structural and convexity features (GSC) [10], and features based on moments of binary images [11].

A different template-based approach is given by using local features in conjunction with elastic matching. Notable work in this domain includes the corner features proposed in [12], sliding window features in combination with dynamic time warping (DTW) [13], and gradient angle features in combination with a cohesive elastic distance [14]. DTW, in particular, is well-established in the field and has been used in combination with different features in recent work, e.g., based on word profiles [15], closed contours [16], and local gradients [17, 18]. In [18], an extension of DTW is presented that allows to match keyword templates with complete text lines rather than segmented word images.

A very general learning-based approach at word level was presented in [6]. Based on local gradient features, posterior probabilities of keyword HMMs are used for keyword spotting in conjunction with universal vocabularies for score normalization. A similar approach was presented in [19] for non-symmetric half plane HMMs (NSHP-HMMs). Instead of using posterior probabilities, a scoring method based on Fisher kernels has been proposed in [20].

For spotting arbitrary keywords, the learning-based approach has been investigated at character level as well. Character template images were used in [21, 22] to train generalized Hidden Markov Models (gHMMs) for keyword spotting. While promising results were reported for historical Latin manuscripts [21] and Arabic scripts [22], an automatic acquisition of such character templates from handwritten text images is difficult in general. The same limitation is faced in [31], where segmented character images and Gabor features were used in a template-based method for keyword spotting.

In an earlier work [23], character HMMs were used to spot street names for the constrained task of address reading. In a segmentation-free approach, character models are trained on complete text line images and are then connected to model keywords as well as general non-keyword text. Recently, this line of research was followed in [24] for unconstrained handwritten word spotting. Another system that is based on learning at character level was presented in [25] based on bidirectional long short-term memory neural networks (BLSTM-NNs).

In contrast to [23, 24], the method proposed in this paper is not dependent on a lexicon that imposes a high computational complexity for large vocabulary tasks. Instead of a comparison with other lexicon words, a log-odds scoring with respect to a general filler model is employed as a confidence measure for keyword spotting. This technique is well-known from other application domains of HMMs, e.g., speech recognition [26] and bioinformatics [32], and has been used similarly in [6].

1.2. Contribution

In this work, a novel lexicon-free keyword spotting system using trained character HMMs is presented for handwritten word spotting. On several data sets, its superior performance over a standard template matching approach is demonstrated.

This article is an extended version of a conference paper [33] that has introduced a preliminary version of the system. In general, a more thorough description and evaluation of the system is provided in this work. In particular, in Section 2.4, a theoretical justification for the proposed text line scoring for keyword spotting is given. The computational complexity of this scoring mechanism is analyzed in Section 2.5. Furthermore, the experimental evaluation is extended by including an additional data set, i.e., the Parzival database described in Section 4.3.

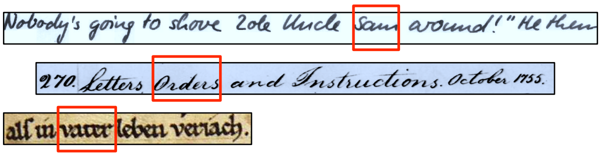

Nobody's going to shove 2ole Uncle Sam around!" He then
270. Letters Orders and Instructions. October 1755.
alſ in vatter leben veriach.

Figure 1: Word Spotting

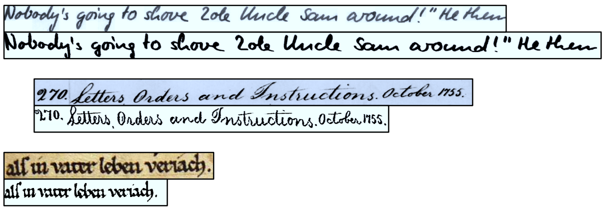

Nobody's going to shove 2ole Uncle Sam around!" He then
Nobody's going to shove 2ole Uncle Sam around!" He then
270. Letters Orders and Instructions. October 1755.
270. Letters Orders and Instructions. October 1755.
alſ in vatter leben veriach.
alſ in vatter leben veriach.

Figure 2: Image Preprocessing

Finally, because it was often argued against the use of character-based systems for word spotting in the handwriting recognition literature, a brief discussion is provided in Section 5.3 that results in the conclusion that most of the arguments against such systems do not hold true for the proposed approach.

2. Word Spotting System

Word spotting refers to the task of retrieving keywords from document images. In this paper, we consider handwritten documents written in natural language, such as letters, memorandums, and historical manuscripts. Without transcribing the handwriting, the proposed system allows a user to search for arbitrary keywords in the document.

The input of the word spotting system is given by an arbitrary keyword string and a text line image. A score is then assigned to the text line image that represents the likelihood of the text line to contain the keyword. If this likelihood is greater than a certain threshold, the image is returned as a positive match alongside with the position of the keyword. Three examples from different data sets are given in Figure 1 for spotting the keywords “Sam”, “Orders”, and “vater”, respectively.

The proposed word spotting system includes several processing steps. At the preprocessing stage, the input text line images are normalized in order to cope with different writing styles. Afterwards, a sequence of local feature vectors is extracted using a sliding window. The statistical learning model employed is Hidden Markov Models (HMMs). In the training phase, character HMMs are trained for each character of the alphabet based on transcribed text line images. At the recognition stage, the trained character HMMs are connected to a keyword text line model in order to calculate the likelihood score of the input text line. This likelihood score is finally normalized with respect to a general filler text line model and the length of the keyword feature vector sequence before it is compared to a threshold.

In the following, the word spotting system is described in detail. In Section 2.1, image normalization is addressed, Section 2.2 covers feature extraction, Section 2.3 discusses training and recognition with HMMs, and Section 2.4 presents the HMM-based text line score for word spotting. The computational complexity of this text line scoring is discussed in Section 2.5.

2.1. Image Preprocessing

In the first preprocessing step, the handwritten document images are binarized and individual text lines are extracted. The binarization and text line extraction methods depend on the type and quality of the documents. In general, special acquisition forms simplify both tasks for modern documents, while for real-world historical documents, more sophisticated methods are needed in order to cope with noisy background, touching text lines, and paper or parchment degradation artifacts [4].

In this paper, we focus on keyword spotting and do not take text line segmentation errors into account. Consequently, we work with perfectly segmented text line images. For binarization, a simple global threshold is used. For the historical George Washington and Parzival data sets (see Sections 4.2 and 4.3), a local edge enhancment with a Difference of Gaussian filter is applied first in order to cope with the noisy background [34]. Note that no segmentation of the text lines into words is needed for the proposed word spotting system.

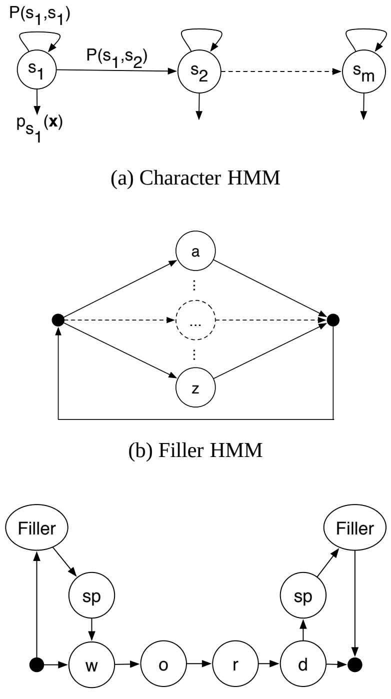
In the second step, the binary text line images are normalized in order to cope with different writing styles. As proposed in [35], the skew, i.e., the inclination of the text line, is removed first by rotation. Then, the slant, i.e., the inclination of the letters, is removed using a shear transformation. Next, a vertical scaling procedure is applied to normalize the height with respect to the lower and upper baseline. Finally, horizontal scaling normalizes the width of the text line with respect to the estimated number of letters.

For more details on text line normalization, we refer to [35]. In Figure 2, exemplary image preprocessing results are shown for all three data sets considered in this paper.

2.2. Feature Extraction

For representing binary, normalized text line images with statistical feature vectors, we employ a set of nine local features that were used in previous work on handwriting recognition [35]. In the field of word spotting, a subset of these features has been used in [13] and, more recently, the complete set was used among others in [6]. Although the features can be outperformed in terms of accuracy by more complex features, e.g., gradient features [6] and graph similarity features [36], they are still appealing for handwriting recognition, since they can achieve high performance at very low computational costs and don’t need any parameters to tune.

Using a sliding window technique, text line images are represented by a sequence of N local feature vectors $\mathbf{x}_1, \ldots, \mathbf{x}_N$ with $x_i \in \mathbb{R}^n$. This sequence is extracted by a sliding window of one



(a) Character HMM

(b) Filler HMM

(c) Keyword HMM

Figure 3: Hidden Markov Models

pixel width moving from left to right over the image. At each of the N positions of the sliding window, $n = 9$ geometrical features are extracted. 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 upper and lower contour, the number of black-white transitions, and the fraction of black pixels between the contours. For a more detailed description of the features, we refer to [35].

2.3. Hidden Markov Models

The proposed word spotting system is based on Hidden Markov Models (HMMs) as the underlying statistical learning model. As a matter of fact, HMMs have been widely used in the field of handwriting recognition [37]. A key property of HMMs for modeling handwriting is that they are able to cope with the problem that characters are connected in a cursively handwritten text and can not be segmented reliably before recognition. This "chicken-and-egg" problem is also known as Sayre's Paradox [38]. Using variable-length character models, HMMs optimize both character segmentation and character recognition at the same time.

The structure of the character HMMs used is shown in Figure 3a. Each character model has a certain number m of hidden states $s_1, \dots, s_m$ arranged in a linear topology. 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 Gaussian Mixture Model (GMM). 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, taking into account variable character lengths.

Using continuous HMMs, the underlying GMM of the output probability distributions $p_{s_j}(\mathbf{x})$ is given by

$$p_{s_j}(\mathbf{x}) = \sum_{k=1}^{G} w_{jk} \mathcal{N}(\mathbf{x} \mid \mu_{jk}, \Sigma_{jk})$$

where G is the number of Gaussians and $\mathcal{N}(\mathbf{x} \mid \mu_{jk}, \Sigma_{jk})$ is a normal distribution with mean μ_{jk} and covariance matrix Σ_{jk}. The weights w_{jk} are positive and sum up to one. In order to reduce the number of model parameters, we assume zero covariance, taking only the mean and variance into account.

The character models are trained using labeled text line images. First, a text line model is created as a sequence of character models according to the transcription. Then, the probability of this text line model to emit the observed feature vector sequence $\mathbf{x}_1, \dots, \mathbf{x}_N$ is 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 [39].

Using trained character models, different text line models can be created. In Figure 3b, a general *filler text line model* is shown that represents an arbitrary sequence of characters. For a given text line model, the likelihood of the observed feature vector sequence $\mathbf{x}_1, \dots, \mathbf{x}_N$ is calculated, using the Viterbi algorithm [39]. The output is the most likely character sequence together with the start and end position of the characters.

For our HMM implementation, we have used the popular HTK toolkit[1]. Important HMM parameters that are optimized using an independent validation set are the number of states m of the character models and the number of Gaussian mixtures G of the output probability distributions.

2.4. Text Line Scoring

A decision rule for word spotting is usually based on the score $s(X, W)$ that is assigned to the text image X for the keyword W. If the score is better than a certain threshold, the image is returned as a positive match. Examples of such scores include the minimum distance $s(X, W) = \min\{d(X, T_j)\}$ of the word image X to a number of keyword template images $T_1, \dots, T_l$ used by template-based methods and the posterior probability $s(X, W) = p(W|X)$ based on trained keyword models used in [6]. In contrast to word image scores, the proposed word spotting system relies on a score for complete text line images based on trained character models.

In [33], we have presented a text line score based on the likelihood ratio between a keyword text line model and a filler text line model. As an extension of the work presented in [33], a theoretical justification of this score is given below[2].

The proposed score $s(X, W)$ of the text line image X for the keyword W is based on the posterior probability $p(W|X_{a,b})$

[1] `http://htk.eng.cam.ac.uk/`

[2] Note that a slightly different notation is used, i.e., we use a log-likelihood difference instead of a likelihood ratio.

where a and b are the most likely start and end position of the keyword and $X_{a,b}$ is the corresponding part of the text line image. Applying Bayes' rule to logarithmic values, we obtain

$$\log p(W|X_{a,b}) = \log p(X_{a,b}|W) + \log p(W) - \log p(X_{a,b})$$

The prior $p(W)$ can be integrated into a keyword specific threshold that is optimized at training stage. For arbitrary keywords that are not known at the training stage, we assume equal priors. In both cases, we only take the terms

$$\log p(X_{a,b}|W) - \log p(X_{a,b})$$

into account for calculating the text line score. This log-likelihood difference is obtained by two text line models.

The first model used for the term $\log p(X_{a,b}|W)$ is the *keyword text line model* K shown in Figure 3c for the keyword "word". It is constrained to contain the exact sequence of characters of the keyword at the beginning, in the middle, or at the end of the text line, separated by the space character "sp". The rest of the text line is an arbitrary sequence of characters that is modeled with the *filler text line model* F shown in Figure 3b. From Viterbi recognition with keyword model K we obtain the most likely start position a and end position b of the keyword, alongside with the log-likelihood $\log p(X_{a,b}|W) = \log p(X_{a,b}|K)$.

The second model used for the term $\log p(X_{a,b})$ is the unconstrained filler model F. The obtained log-likelihood $\log p(X_{a,b}) = \log p(X_{a,b}|F)$ indicates the general conformance of the text image to the trained character models. A known writing style that was used for training has a higher likelihood than an unknown writing style. By subtracting the term $\log p(X_{a,b})$ from the keyword log-likelihood $\log p(X_{a,b}|W)$ the score is normalized with respect to the writing style and allows a better generalization to unseen test images. Also known as log-odds scoring [32], this technique has been used similarly in [6] with a single GMM filler.

For the final text line score we can ignore the start and end position of the keyword, because the log-likelihood difference between the keyword model K and the filler model F is zero outside the keyword position, not taking into account the small deviations at the keyword borders. That is, the log-likelihood difference $\log p(X_{a,b}|W) - \log p(X_{a,b})$ is given by the log-likelihood difference

$$\log p(X|K) - \log p(X|F)$$

of the text line models K and F over the complete text line image X. In a final step, the log-likelihood score is normalized with respect to the keyword length $L_K = b - a$ and compared to a threshold T for word spotting.

$$s(X, W) = \frac{\log p(X|K) - \log p(X|F)}{L_K} \geq T$$

Note that the maximum text line score $s(X, W)$ is zero. This maximum score is achieved if the exact keyword character sequence is, in fact, the most likely character sequence in the filler model. The optimal value of T can be determined in the training phase with respect to the user needs as a trade-off between system precision and recall. Precision and recall are discussed in Section 5 and are calculated for all possible thresholds for system evaluation.

In case of multiple occurrences of a keyword within a text line, the most likely position is taken into account for scoring the text line image and can be returned if needed. In order to return all keyword positions, a modification of the keyword model given in Figure 3c or an iterative application of the model would be needed. For the data sets considered in this paper (see Section 4), multiple keyword occurrences are very rare, i.e., less than one percent of the text lines are affected.

2.5. Computational Complexity and Discussion

In the proposed lexicon-free approach to word spotting, the computational time needed to score a text line image is given by $O(n^2L)$ for Viterbi recognition [39] with the keyword and filler text line models where n indicates the number of characters in the alphabet and L the length of the text line.

This is a great advantage in terms of computational speed when compared to transcription-based approaches using a lexicon of words. Here, the time complexity would be $O(N^2L)$ for a lexicon of size N. For transcription-based systems that deal with texts written in natural language, typical values for N are several ten thousands, whereas the number n of characters in an alphabet is usually below one hundred.

A drawback of the lexicon-free approach, however, is that no language model can be applied at word level. Instead, a simple language model is used at character level for the filler model given by equal probabilities. However, as pointed out in Section 2.4, the word prior $\log p(W)$ can be integrated using keyword specific local thresholds that are discussed in Section 4.4.

3. Reference System

In this paper, we compare the proposed word spotting system with a widely used reference system that is based on Dynamic Time Warping (DTW). DTW-based keyword spotting was proposed in [28] for speech recognition and is also well-established in the field of handwritten word spotting [15, 16, 17, 18].

Since DTW can be applied to the same feature vector sequence that is used for HMM-based spotting (see Sections 2.1 and 2.2), a fair comparison between classical template matching and the proposed learning-based approach is possible when using this reference system. While the proposed system uses the feature sequence for training character models, DTW directly matches the feature sequence of keyword template images and unknown text images. If the resulting distance score is lower than a certain threshold, a positive match is returned.

In order to apply DTW at text line level, a segmentation of the text line into words is needed. Especially for historical documents, an explicit segmentation is a challenging problem [40]. The impact of errors in word image extraction can be reduced by adding an uncertainty margin around candidate words [41]. Another solution is to use continuous dynamic programming to perform a subsequence matching within a complete text line as recently proposed in [18]. In this paper, we consider a perfect, manually corrected word segmentation that is readily available for the data sets used for experimental evaluation. Hence, the results reported for DTW in Section 5 can be seen as an upper bound of the DTW performance with respect to word segmentation.

3.1. DTW Distance

For calculating the DTW distance between two feature vector sequences $\mathbf{x}_1, \ldots, \mathbf{x}_N$ and $\mathbf{y}_1, \ldots, \mathbf{y}_M$, an optimal alignment of the two sequences is determined that takes into account different sequence lengths. By this process, the feature vectors are aligned along a common, warped time axis. The cost of an alignment is given by the sum of distances $d(\mathbf{x}, \mathbf{y})$ of each aligned vector pair. The distance measure $d(\mathbf{x}, \mathbf{y})$ employed is the squared Euclidean distance

$$d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{9} (\hat{x}_i - \hat{y}_i)^2$$

based on normalized features $\hat{x}_i$. For normalization, a linear scaling is applied such that $\hat{x}_i \in [0, 1]$. The DTW distance $DTW(X, Y)$ of the word images X and Y is then given by the minimum alignment cost that is found by means of dynamic programming [15]. For speeding up the distance calculation, a Sakoe-Chiba band [42] is used. The resulting distance is finally normalized with respect to the length of the optimal alignment, also called warping path. For more details on the DTW distance algorithm, we refer to [15].

3.2. Text Line Scoring

For word spotting, the score $s(X, W)$ of the text line image X for the keyword W is based on the DWT distance between segmented word images. Given the images $X_1, \ldots, X_k$ of all words contained in the text line and the available keyword template images $T_1, \ldots, T_l$, the text line score is given by the minimum DTW distance

$$s(X, W) = \min\{D(X_i, T_j)\} \leq T$$

for $1 \leq i \leq k$ and $1 \leq j \leq l$. That is, a text line is returned as a positive match, if the distance of one of its word images to any keyword template image is smaller than a certain threshold T.

Note that unlike the proposed system, the DTW-based reference system requires at least one template image for spotting a keyword. For a large amount of template images, the computation time poses another problem. While all training data is incorporated in the character HMMs in the proposed system and can be used for a single match against a given feature vector sequence, DTW requires matching the sequence with all available keyword templates.

4. Experimental Evaluation

For evaluating the proposed word spotting system, three data sets were used: the IAM off-line database[3] [43], the George Washington database[4] [44], and the Parzival database [45]. On each data set, the proposed HMM-based word spotting system of Section 2 is compared to the DTW-based reference system of Section 3, based on different evaluation measures.

Database	Train	Valid	Test	Char.	Keywords
IAM	6161	920	929	81	882
GW	328	164	164	83	105
PAR	2236	911	1328	96	1217

Table 1: Number of text lines used for training, validation, and testing as well as number of characters and keywords.

Database	Char.	Keywords
IAM	4054.9	7.4
GW	220.0	2.7
PAR	797.2	7.6

Table 2: Average number of available training samples. The character samples occur in unsegmented text lines.

The data sets are introduced in Sections 4.1– 4.3, the word spotting task and the evaluation measures are presented in Section 4.4, and the system setup is discussed in Section 4.5.

4.1. IAM off-line database (IAM)

The IAM data set consists of 1,539 pages of handwritten modern English text from the Lancaster–Oslo/Bergen corpus (LOB) [46], written by 657 writers. An exemplary document image is shown in Figure 4a and the most important statistics of the data set are listed in Tables 1 and 2.

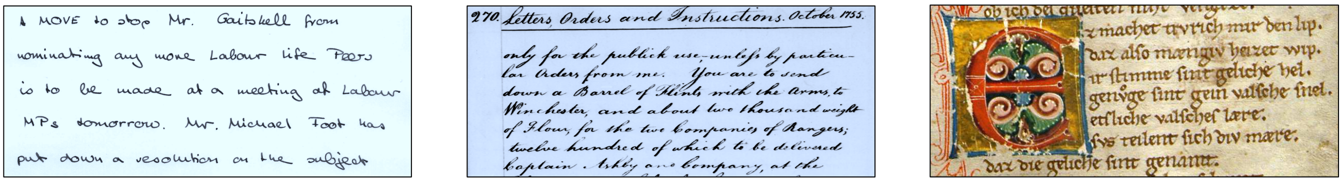
An important characteristic of the data set is that each of the three subsets for training, validation, and testing, respectively, contains text lines from a different set of writers. Hence, the main challenge for word spotting using this data set is to retrieve keywords in writing styles that are unknown at training stage. In order to achieve this challenging goal, a large amount of training data are used. For each of the characters appearing in at least one keyword, 4054.9 training samples are available on average in the training text line images, while only 7.4 keyword templates are present on average. This clearly motivates the use of trained character models.

4.2. George Washington database (GW)

The GW data set includes 20 pages of letters written by George Washington and his associates in the year 1755. An example of the relatively clean document images is shown in Figure 4b and statistics appear in Tables 1 and 2. Because of the small size of the data set, we have performed a four-fold cross validation for experimental evaluation. Therefore, Tables 1 and 2 include rounded mean values over all cross validation sets.

[3] http://www.iam.unibe.ch/fki/databases/iam-handwriting-database

[4] George Washington Papers at the Library of Congress, 1741–1799: Series 2, Letterbook 1, pages 270–279 & 300–309, http://memory.loc.gov/ammem/gwhtml/gwseries2.html



(a) IAM database

(b) George Washington database

(c) Parzival database

Figure 4: Data Sets

The same set of pages is considered as in [44], but we do not use the automatically segmented and extracted word images. In this paper, we focus on keyword spotting and do not take segmentation errors into account. Therefore, we semi-automatically created a ground truth for text line and word segmentation following the procedure described in [34].

The writing style is very similar for all 20 pages. Hence, the main challenge for word spotting using this data set is to deal with a small amount of training data.

4.3. *Parzival database (PAR)*

The PAR data set presented in [45] contains 45 pages of a medieval manuscript originating in the 13th century. The manuscript contains the epic poem *Parzival*, one of the most important epic works in the European Middle Ages, and is written with ink on parchment in the Middle High German language. An example document image is shown in Figure 4c and data set statistics are given in Tables 1 and 2.

Although several writers have contributed to the manuscript, the writing styles found on the 45 pages of the data set are very similar. When compared with the GW data set, a much larger amount of training data is available. Hence, the best conditions for word spotting are met here among the three data sets.

4.4. *Task and Performance Evaluation*

For experimental evaluation, a set of keywords is spotted on the test set of the IAM, GW, and PAR database, respectively. The test set of a database contains a number of text line images perfectly extracted from whole manuscript pages. For the DTW-based reference system, perfectly segmented word images are provided additionally. For each keyword, the word spotting systems are required to return all text line images that contain the keyword. The decision whether to return a text line image for a given keyword or not is based on the score discussed in Section 2.4 for the proposed system and in Section 3.2 for the reference system.

For measuring the performance, we consider two possible scenarios. In the first scenario, a *local threshold* is used for each keyword separately. Given a keyword, the number of *true positives (TP)*, *false positives (FP)*, and *false negatives (FN)* are evaluated for all possible local thresholds. From these values, the *recall* $R = \frac{TP}{TP+FN}$ and *precision* $P = \frac{TP}{TP+FP}$ of the word spotting system are obtained and are presented in a recall-precision curve by averaging over all keyword queries. Using the popular `trec_eval` [5] software, the mean average precision (MAP), given by the area under curve, as well as the R-precision (RP), given by the point where recall and precision are equal, are considered to assess the system performance.

In the second scenario, a *global threshold* is used that is independent of the keyword. For a single query and all possible global thresholds, the recall-precision curve as well as the MAP and RP are calculated in the same way as for local thresholds. Note that using a global threshold for all keywords is sound because the text line scores of the proposed system and the reference system are normalized with respect to the keyword length.

In a real-world situation, the proposed word spotting system can be used in both scenarios. For a vocabulary of common keywords, local thresholds can be optimized at training stage. For arbitrary out-of-vocabulary keywords, a global threshold has to be applied.

The set of keywords used for evaluation is derived as follows. First we consider, for the IAM database, all 3,421 non stop words [6] among the 4,000 most frequent words. For the GW database, all words of the cross validation training set are used, and for the PAR database, all 3,220 words from the training set are used. In the second step, we constrain the keyword list to those keywords that appear at least once in the training set and at least once in the test set, resulting in the number of keywords given in Table 1. For a fair and reliable system evaluation, the first constraint takes into account that at least one template image is needed for the DTW-based reference system and the second constraint accounts for the fact that the recall is not well-defined for a keyword that is not present in the test set.

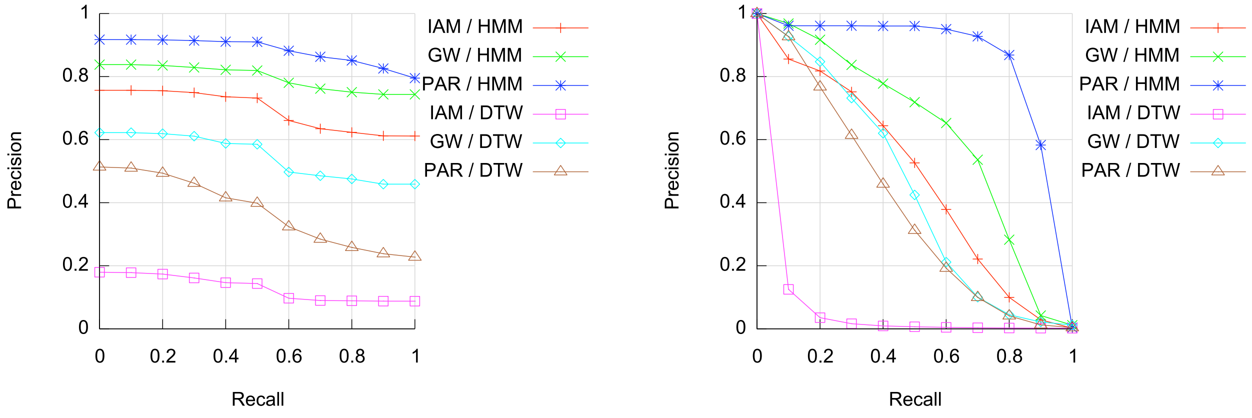
Note that in [33] no constraints were used on the set of keywords. Thus, the results obtained are not directly comparable. Using an unconstrained set of keywords results in a more difficult keyword spotting task that may be biased towards the proposed system. In the present paper, we have ruled out this possible bias.

4.5. *System Setup*

For the proposed HMM-based system, the training and validation set of each database is used for statistical learning and parameter optimization, respectively. For the number of states

[5] http://trec.nist.gov/trec_eval/

[6] We used the stop word list from the SMART project [47], which can be found at http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop



(a) Local Thresholds

(b) Global Threshold

Figure 5: Word Spotting Performance

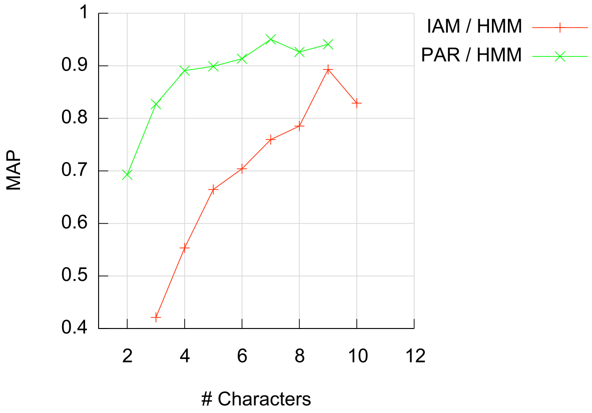


Figure 6: Keyword Size Evaluation

System	L-MAP	L-RP	G-MAP	G-RP
IAM / HMM	68.92	61.83	47.75	51.20
IAM / DTW	12.65	9.50	4.07	10.66
GW / HMM	79.28	72.73	62.08	63.78
GW / DTW	54.08	45.70	43.95	47.54
PAR / HMM	88.15	84.34	85.53	82.67
PAR / DTW	36.85	31.08	39.22	42.30

Table 3: Recall-precision evaluation. For each system the mean average precision (MAP) and the R-precision (RP) are indicated for the local threshold scenario (L) and the global threshold scenario (G).

m of the character HMMs, we have adopted an optimal number from previous work on handwriting recognition [35, 45].

The number of Gaussian mixtures *G* was optimized with respect to a small keyword spotting task on the validation set. Hereby, the 10 most frequent keywords in the validation set were spotted and the performance was compared based on the MAP using a global threshold. The number of states of the separating space character “sp” had a large influence on the word spotting result. Therefore, it was optimized separately on the validation set with respect to the same word spotting task.

For the DTW-based reference system, no training is required. Instead, the keyword template images from the training set are used directly for image matching.

5. Results and Discussion

For the three data sets IAM, GW, and PAR, the recall-precision curves corresponding to keyword specific local thresholds are given in Figure 5a for the proposed HMM-based system as well as for the DTW-based reference system. The corresponding recall-precision curves for the global threshold scenario are shown in Figure 5b. Note that for the GW data set, the recall-precision curves are given by the mean over the four cross validation runs. The MAP and RP are listed in Table 3.

5.1. Local Threshold

In the local threshold scenario, the proposed system clearly outperforms the reference system with respect to both, the MAP and the RP given in Table 3 on all three data sets. The improvements are statistically significant over all keyword queries (t-test, $\alpha = 0.05$).

For the multi-writer scenario of the IAM data set, the proposed system has the lowest performance, but also the largest improvement over the reference system. The template-based reference approach clearly fails for the large number of different writing styles, while the proposed learning-based approach shows a good generalization behavior.

For the single writer condition of the GW data set, the smallest difference between the systems can be observed. Here, the small amount of training data available imposes a problem for the proposed learning-based approach. Still, a better performance is achieved than by the reference system.

Finally, for the single writer setting of the PAR data set, the best performance is achieved with the proposed system when compared with the two other data sets. The relatively large amount of training data allows keyword retrieval with a high precision. Interestingly, the DTW-based reference system achieves worse results on the PAR data set than on the GW data set despite the fact that a large number of keyword templates are available.

An evaluation of the MAP depending on different keyword sizes is presented in Figure 6 for the proposed system and the two larger data sets, i.e., IAM and PAR. Keywords are grouped together that have the same number of characters. For each set that contains at least ten keywords, the MAP is indicated. The performance is remarkably improved for large keywords that can be retrieved quite reliably even in the multi-writer scenario of the IAM data set. The reason is twofold. First, unreadable parts within longer words can be compensated for large keyword sizes in the HMM-score. Secondly, a problem for small words is given by the fact that they are prone to be spotted within larger words. This problem could be circumvented by working on word images rather than text line images. However, the system would then rely on an additional preprocessing step and would be prone to word segmentation errors.

5.2. Global Threshold

For comparison with the reference system, the same observations hold true for the global threshold scenario as for the local thresholds. When using global thresholds, a lower overall performance in terms of MAP and RP is achieved with the proposed system. Following the reasoning of Section 2.4, this can be explained by the fact that the word prior $\log p(W)$ is no longer considered for scoring. Nevertheless, remarkable results are achieved in this out-of-vocabulary scenario.

In general, higher precision values are achieved for low recall values in the global threshold scenario at the expense of lower precision for high recall. This can be explained by the fact that the best scores among all keywords contribute to the high precision for low recall values. Interestingly, for the DTW reference system, this leads to an improvement of the RP for all data sets. Also, the MAP is improved for the PAR data set.

5.3. Comparison with Transcription-Based Approaches

In the literature, it was often argued against word spotting systems that rely on trained character models based on the argument that they share the same limitations as transcription-based handwriting systems [6]. The three major arguments are, first, that they have a slow computational speed, secondly, that they need a large amount of training data, and thirdly, that they can not be easily adapted to new languages and alphabets.

Regarding the first argument, it is shown in Section 2.5 that the proposed system does not suffer from the high computational requirements imposed by a lexicon. Instead, character models are used to efficiently score text line images.

The second argument is proven wrong in the experimental scenario considered for the GW data set where it is shown that even in case of a small amount of available training data, the proposed system is able to outperform the template-based DTW reference system.

The third argument holds true to some degree and shows a limitation of the proposed approach. If the language under consideration and, in particular, its alphabet are unknown, no character HMMs can be trained. Instead, the only option available might be template matching.

6. Conclusions

A novel lexicon-free keyword spotting system using trained character HMMs was presented for handwritten word spotting. With the proposed method, arbitrary keywords can be spotted and text line images are not required to be segmented into words. The text line score is calculated efficiently without suffering from high computational costs imposed by a lexicon.

In an experimental evaluation, the proposed system was compared with a standard DTW-based template matching technique for three different word spotting conditions. A multi-writer scenario with many training samples is considered on the modern IAM database, a single writer scenario with only few training samples on the historical George Washington database, and a single writer scenario with many training samples on the historical Parzival database. For each database, the difficult task of spotting between 105 up to 1,217 keywords was investigated.

On all data sets, the proposed method clearly outperformed the reference system. The largest difference was observed for the multi-writer task on the IAM database. Here, the template-based reference system clearly failed, while the learning-based system showed good generalization behavior. Even in the absence of a large training corpus for the George Washington database, the template-based approach was outperformed.

A limitation of possible application areas of the proposed method, however, is given by the fact that the language under consideration and its alphabet are required to be known. In order to train character HMMs, a set of transcribed word or text line images is needed, which can be costly to obtain, especially for historical documents.

In order to improve the proposed system, future research will include the integration of more language knowledge at word level in the absence of a lexicon, the investigation of score normalization methods that are not based on filler models, and the improvement of the space model that separates keywords from the rest of the text. Finally, an extension to the search for regular expressions would be interesting.

The high performance and efficiency of the proposed method is promising for industrial applications, such as automatic mail sorting or digital libraries, in particular if the keywords under consideration are large enough to achieve high precision values. Scientifically, a rewarding line of research would be to use the high keyword spotting performance in the context of interactive systems and semi-supervised learning.

Acknowledgments

This work has been supported by the Swiss National Science Foundation (Project CRSI22-125220).

References

[1] A. Vinciarelli, A Survey on Off-Line Cursive Word Recognition, Pattern Recognition 35 (7) (2002) 1433–1446.

[2] R. Plamondon, S. Srihari, Online and Off-Line Handwriting Recognition: A Comprehensive Survey, IEEE Trans. PAMI 22 (1) (2000) 63–84.

[3] H. Bunke, T. Varga, Off-Line Roman Cursive Handwriting Recognition, in: B. Chaudhuri (Ed.), Digital Document Processing: Major Directions and Recent Advances, vol. 20, Springer, 165–173, 2007.

[4] A. Antonacopoulos, A. Downton, Special Issue on the Analysis of Historical Documents, Int. Journal on Document Analysis and Recognition 9 (2) (2007) 75–77.

[5] R. Manmatha, W. B. Croft, Word Spotting: Indexing Handwritten Archives, in: M. T. Maybury (Ed.), Intelligent Multimedia Information Retrieval, MIT Press, 43–64, 1997.

[6] J. Rodriguez, F. Perronnin, Handwritten Word-Spotting Using Hidden Markov Models and Universal Vocabularies, Pattern Recognition 42 (9) (2009) 2106–2116.

[7] G. Nagy, D. Lopresti, Interactive Document Processing and Digital Libraries, in: Proc. 2nd Int. Workshop on Document Image Analysis for Libraries, 2–11, 2006.

[8] T. M. Rath, R. Manmatha, V. Lavrenko, A Search Engine for Historical Manuscript Images, in: Proc. 27th Int. Conf. on Research and Development in Information Retrieval, 369–376, 2004.

[9] R. Manmatha, C. Han, E. Riseman, Word Spotting: A New Approach to Indexing Handwriting, in: Proc. Int. Conf. on Computer Vision and Pattern Recognition, 631—637, 1996.

[10] B. Zhang, S. N. Srihari, C. Huang, Word Image Retrieval Using Binary Features, in: Proc. Document Recognition and Retrieval XI, vol. 5296, 45–53, 2004.

[11] A. Bhardwaj, D. Jose, V. Govindaraju, Script Independent Word Spotting in Multilingual Documents, in: Proc. 2nd Int. Workshop on Cross Lingual Information Access, 48–54, 2008.

[12] J. L. Rothfeder, S. Feng, T. M. Rath, Using Corner Feature Correspondences to Rank Word Images by Similarity, in: Proc. Workshop on Document Image Analysis and Retrieval, 30–35, 2003.

[13] T. M. Rath, R. Manmatha, Word Image Matching Using Dynamic Time Warping, in: Proc. Int. Conf. on Computer Vision and Pattern Recognition, 521–527, 2003.

[14] Y. Leydier, A. Ouji, F. LeBourgeois, H. Emptoz, Towards an Omnilingual Word Retrieval System for Ancient Manuscripts, Pattern Recognition 42 (9) (2009) 2089–2105.

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

[16] T. Adamek, N. E. Connor, A. F. Smeaton, Word Matching Using Single Closed Contours for Indexing Historical Documents, Int. Journal on Document Analysis and Recognition 9 (2) (2007) 153–165.

[17] J. Rodriguez, F. Perronnin, Local Gradient Histogram Features for Word Spotting in Unconstrained Handwritten Documents, in: Proc. 1st Int. Conf. on Frontiers in Handwriting Recognition, 7–12, 2008.

[18] K. Terasawa, Y. Tanaka, Slit Style HOG Features for Document Image Word Spotting, in: Proc. 10th Int. Conf. on Document Analysis and Recognition, vol. 1, 116–120, 2009.

[19] C. Choisy, Dynamic Handwritten Keyword Spotting Based on the NSHP-HMM, in: Proc. 9th Int. Conf. on Document Analysis and Recognition, 242–246, 2007.

[20] F. Perronnin, J. Rodriguez-Serrano, Fisher Kernels for Handwritten Word-spotting, in: Proc. 10th Int. Conf. on Document Analysis and Recognition, vol. 1, 106–110, 2009.

[21] J. Edwards, Y. W. Teh, D. Forsyth, R. Bock, M. Maire, G. Vesom, Making Latin Manuscripts Searchable Using gHMM's, in: Advances in Neural Information Processing Systems, 385–392, 2004.

[22] J. Chan, C. Ziftci, D. Forsyth, Searching Off-line Arabic Documents, in: Proc. Int. Conf. on Computer Vision and Pattern Recognition, 1455–1462, 2006.

[23] A. El Yacoubi, M. Gilloux, J.-M. Bertille, A Statistical Approach for Phrase Location and Recognition within a Text Line: An Application to Street Name Recognition, IEEE Trans. PAMI 24 (2) (2002) 172–188.

[24] S. Thomas, C. C., L. Heutte, T. Paquet, An Information Extraction Model for Unconstrained Handwritten Documents, in: Proc. 20th Int. Conf. on Pattern Recognition, 3412–3415, 2010.

[25] V. Frinken, A. Fischer, H. Bunke, A Novel Word Spotting Algorithm Using Bidirectional Long Short-Term Memory Neural Networks, in: Proc. 4th Int. Workshop on Artificial Neural Networks in Pattern Recognition, vol. 5998 of *LNCS*, Springer, 185–196, 2010.

[26] R. C. Rose, D. B. Paul, A Hidden Markov Model Based Keyword Recognition System, in: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 1, 129–132, 1990.

[27] F. R. Chen, L. D. Wilcox, D. S. Bloomberg, Word Spotting in Scanned Images Using Hidden Markov Models, in: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, vol. 5, 1–4, 1993.

[28] A. E. R. Cory S. Myers, Lawrence R. Rabiner, An Investigation of the Use of Dynamic Time Warping for Word Spotting and Connected Speech Recognition, in: Proc. Int. Conf. on Acoustics, Speech, and Signal Processing, 173–177, 1980.

[29] S.-S. Kuo, O. E. Agazzi, Keyword Spotting in Poorly Printed Documents Using Pseudo 2-D Hidden Markov Models, IEEE Trans. PAMI 16 (8) (1994) 842–848.

[30] G. Scott, H. Longuet-Higgins, An Algorithm for Associating the Features of Two Patterns, in: Proc. Royal Society London, vol. B244, 21–26, 1991.

[31] H. Cao, V. Govindaraju, Template-free Word Spotting in Low-Quality Manuscripts, in: Proc. 6th Int. Conf. on Advances in Pattern Recognition, 135–139, 2007.

[32] C. Barrett, R. Hughey, K. Karplus, Scoring Hidden Markov Models, CABIOS 13 (1997) 191–199.

[33] A. Fischer, A. Keller, V. Frinken, H. Bunke, HMM-Based Word Spotting in Handwritten Documents Using Subword Models, in: Proc. 20th Int. Conf. on Pattern Recognition, 3416–3419, 2010.

[34] A. Fischer, E. Indermühle, H. Bunke, G. Viehhauser, M. Stolz, Ground Truth Creation for Handwriting Recognition in Historical Documents, in: Proc. 9th Int. Workshop on Document Analysis Systems, 3–10, 2010.

[35] U.-V. Marti, 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 (2001) 65–90.

[36] A. Fischer, K. Riesen, H. Bunke, Graph Similarity Features for HMM-Based Handwriting Recognition in Historical Documents, in: Proc. 12th Int. Conf. on Frontiers in Handwriting Recognition, 253–258, 2010.

[37] T. Ploetz, G. A. Fink, Markov Models for Offline Handwriting Recognition: A Survey, Int. Journal on Document Analysis and Recognition 12 (4) (2009) 269–298.

[38] K. M. Sayre, Machine Recognition of Handwritten Words: A Project Report, Pattern Recognition 5 (3) (1973) 213–228.

[39] L. Rabiner, A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proceedings of the IEEE 77 (2) (1989) 257–285.

[40] R. Manmatha, J. L. Rothfeder, A Scale Space Approach for Automatically Segmenting Words from Historical Handwritten Documents, IEEE Trans. PAMI 27 (8) (2005) 1212–1225.

[41] A. Kolcz, J. Alspector, M. Augusteijn, R. Carlson, G. Viorel Popescu, A Line-Oriented Approach to Word Spotting in Handwritten Documents, Pattern Analysis and Applications 3 (2) (2000) 153–168.

[42] H. Sakoe, S. Chiba, Dynamic Programming Algorithm Optimization for Spoken Word Recognition, Trans. on Acoustics, Speech, & Signal Processing 26 (1978) 43–49.

[43] U.-V. Marti, H. Bunke, The IAM-Database: An English Sentence Database for Off-line Handwriting Recognition, Int. Journal on Document Analysis and Recognition 5 (2002) 39–46.

[44] V. Lavrenko, T. M. Rath, R. Manmatha, Holistic Word Recognition for Handwritten Historical Documents, in: Proc. Int. Workshop on Document Image Analysis for Libraries, 278–287, 2004.

[45] A. Fischer, M. Wüthrich, M. Liwicki, V. Frinken, H. Bunke, G. Viehhauser, M. Stolz, Automatic Transcription of Handwritten Medieval Documents, in: Proc. 15th Int. Conf. on Virtual Systems and Multimedia, 137–142, 2009.

[46] S. Johansson, G. Leech, H. Goodluck, Manual of Information to Accompany the Lancaster-Oslo/Bergen Corpus of British English, for Use with Digital Computers, Department of English, University of Oslo, 1978.

[47] G. Salton, The SMART Retrieval System—Experiments in Automatic Document Processing, Prentice-Hall, Inc., 1971.