Abstract—Handwritten word spotting aims at making documento images amenable to browsing and searching by keyword retrieval. In this paper, we present a word spotting system based on Hidden Markov Models (HMM) that uses trained subword models to spot keywords. With the proposed method, arbitrary keywords can be spotted that do not need to be present in the training set. Also, no text line segmentation is required. On the modern IAM off-line database and the historical George Washington database we show that the proposed system outperforms a standard template matching approach based on dynamic time warping (DTW).

Keywords—Handwriting recognition; Hidden Markov models;

I. INTRODUCTION

Offline recognition of unconstrained cursively handwritten text is still a widely unsolved problem and an active area of research. For large vocabularies and different writing styles in general [1], and for degraded historical manuscripts in particular [2], the accuracy of an automatic transcription is far from being perfect. Under these conditions, word spotting has been proposed instead of a complete recognition for the restricted task of retrieving keywords from document images. In this way, the documents under consideration become amenable to searching and browsing [3].

Two different approaches for handwritten word spotting can be distinguished. On the one hand, template-based methods match a query image with labeled template images using, e.g., holistic gradient-based binary features (GSC) [4] or local information by means of dynamic time warping (DTW) [5]. Typically, no training is needed for template-based methods and word template images are rather easy to obtain.

Learning-based methods, on the other hand, train keyword models for scoring query images. In [6], a very general approach is introduced for handwritten word spotting, based on Hidden Markov Models (HMM). It is proposed to train one HMM for each keyword separately. Obviously, this method is limited in three aspects. Firstly, the keywords are required to appear in the training set. Secondly, for each keyword a considerable amount of data is required for training the word HMM. And thirdly, the method is dependent on line segmentation.

When training subword models, e.g., letters rather than words, only a small number of models need to be trained. They can be used to spot arbitrary keywords that are not required to be present in the training set. In [7], [8], generalized HMM (gHMM) letter models are trained based on a small set of manually extracted letter template images. This approach is particularly appropriate when no aligned transcription is available for training. However, only a small variety of letter shapes is considered and letter template images are rather difficult to obtain.

In this paper, we present a learning-based word spotting system that uses HMM subword models. This approach is inspired by the successful application of letter HMMs for handwriting recognition [9]. The proposed method is based on an aligned transcription for training, is able to spot arbitrary keywords, and is segmentation-free, i.e., text lines are not required to be segmented into words. On the publicly available modern IAM off-line database (IAMDB) and the historical George Washington database (GWDB) it is demonstrated that the proposed method outperforms a standard template-based approach using DTW.

In the remainder of this paper, HMM-based word spotting is discussed in detail in Section II, the DTW reference system is introduced in Section III, Section IV presents the experimental results, and Section V draws some conclusions.

II. HMM-BASED WORD SPOTTING

The aim of word spotting is to retrieve keywords from document images. In this paper, we focus on handwritten documents, such as letters or historical manuscripts, and consider complete text line images for keyword matching, i.e., no segmentation of text lines into words is required. Based on Hidden Markov Models (HMM), the proposed word spotting system uses a sequence of letter models to represent text line images and returns a score for each text
line that is based on the likelihood ratio between a keyword model and a filler model representing general non-keyword text. If this score is greater than a certain threshold, the text line is returned as a positive match.

In the following the proposed system is discussed in detail. In Section II-A, image normalization and feature extraction are addressed, Section II-B discusses HMM training and decoding, and Section II-C presents text line scoring.

A. Preprocessing

For word spotting, binary text line images are considered that are normalized prior to feature extraction in order to cope with different writing styles. As proposed in [9], 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 and finally, horizontal scaling normalizes the width of the text line with respect to the estimated number of letters.

For algorithmic processing, a normalized text line image is represented by a sequence of $N$ feature vectors $x_1, \ldots, x_N$ with $x_i \in \mathbb{R}^n$. This sequence is extracted by a sliding window of one 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. These features were originally presented in [9]. 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 more details on image normalization and feature extraction, we refer to [9].

B. Hidden Markov Models

The proposed word spotting system is based on Hidden Markov Models (HMM) of individual letters. Each letter model has a certain number $m$ of hidden states $s_1, \ldots, s_m$ arranged in a linear topology as shown in Figure 1a. The states $s_j$ with $1 \leq j \leq m$ emit observable feature vectors $x \in \mathbb{R}^m$ with output probability distributions $p_{s_j}(x)$ 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.

The letter models are trained using labeled text line images. First, a text line model is created as a sequence of letter models according to the transcription. Then, the probability of this text line model to emit the observed feature vector sequence $x_1, \ldots, x_N$ is maximized by iteratively adapting the initial output probability distributions $p_{s_j}(x)$ and the transition probabilities $P(s_j, s_j)$ and $P(s_j, s_{j+1})$ with the Baum-Welch algorithm [10].

Using trained letter models, different text line models can be created. In Figure 1b, an unconstrained filler model of a text line is shown that represents an arbitrary sequence of letters. For a given text line model, the likelihood of the observed feature vector sequence $x_1, \ldots, x_N$ is calculated, using the Viterbi algorithm [10], and output as a result the most likely letter sequence together with the start and end position of the letters.

C. Text Line Scoring

The score for word spotting is based on the likelihood ratio between two text line models. The first model is the keyword model shown in Figure 1c for the keyword “word” and the second model is the filler model shown in Figure 1b. The keyword model is constrained to contain the exact keyword letter sequence at the beginning, in the middle, or at the end of the text line, respectively, separated with the space character “sp”. A common approach to scoring query images for keyword spotting is given by the likelihood ratio $R = \frac{L_K}{L_F}$ between the likelihood $L_K$ of the keyword model and the likelihood $L_F$ of the filler model. $R$ is bounded by $0 \leq R \leq 1$ in our case, because the same letter models
are used for both the keyword model and the filler model. $R = 1$ if the keyword model letter sequence is the most likely letter sequence.

The final text line score is obtained by normalizing the likelihood ratio with the length $l_K = i_e - i_s$ of the keyword, given by the difference of its end position $i_e$ and start position $i_s$. Independently of the keyword, a positive match is returned if this normalized score is greater than a given threshold $T$.

$$\frac{R}{l_K} > T$$

### III. Reference System

The performance of the proposed word spotting system is compared with a standard word image matching approach described in [5] that is based on Dynamic Time Warping (DTW). For the feature vector sequence representations $x$ and $y$ of two text images, DTW finds an alignment with minimum cost between $x$ and $y$ by means of dynamic programming. Here, each feature vector of the first sequence is mapped to one or several feature vectors of the second sequence. Each mapping produces a cost with respect to the distance of the mapped feature vectors. In our implementation, we use the feature vector representation described in Section II-A and employ the squared Euclidean distance $d(x, y) = \sum_{i=1}^{n} (x_i - y_i)^2$ as the alignment cost.

The word image score is then given by the minimum alignment cost of all keyword images present in the training set, normalized with the length of the alignment path. The text line score is finally given by the minimum word image score calculated over all words of the text line.

Note that unlike the proposed system, the reference system requires the text lines to be segmented into words and the keyword to be present in the training set. Disregarding possible segmentation errors, we used perfectly segmented words for the evaluation of the reference system. If a keyword is not present in the training set, a negative match is returned.

### IV. Experimental Evaluation

We used two different data sets for testing. The IAM off-line database (IAMDB) [11] consists of 1,539 pages of handwritten modern English text, written by 657 writers. From this database, 6,161 writer independent text lines are used for training and 929 for testing. Among the 4,000 most frequent words, 3,421 non stop words [12] are used as keywords.

The George Washington database (GWDB) [13] consists of 20 pages of historical letters, written by George Washington and his associates. Due to the small size of this database (it includes only 675 text lines altogether), a four-fold cross validation is performed and the results are averaged. All 1,067 non stop words are used as keywords.

In Figure 2a and 2b, sample text line images are shown for the IAMDB and the GWDB, respectively. The binary images result from preprocessing as discussed in Section II-A.

#### A. Experimental Setup

While for DTW no training is required, the HMMs are trained on the IAMDB with an optimal number of states and Gaussian mixtures adopted from previous work [9]. For the GWDB, the number of states and Gaussian mixtures was optimized with respect to text line recognition on the validation set based on a closed vocabulary without using a language model.

For evaluating the performance of the word spotting systems, all observed text line scores are used as possible thresholds. If the text line score is better than a given
threshold (greater for HMM and lower for DTW), the text line is returned as a positive match. From the number of true positives (TP), false positives (FP), and false negatives (FN), the \( \frac{TP}{TP+FP} \) and the recall \( \frac{TP}{TP+FN} \) are calculated for each threshold. The proposed HMM-based system and the DTW-based reference system are then compared based on their mean precision.

B. Results

In Figure 3, the precision-recall curves are plotted for both data sets and in Table I, the corresponding mean precision is listed. Note that in case of the GWDB, the average precision from the four-fold cross validation is reported. The proposed word spotting system clearly outperforms the reference system on both datasets. While the precision of the DTW system quickly drops as the recall value increases, the precision of the HMM system remains considerably higher which leads to a higher mean precision. When comparing the two systems with each other, one should furthermore keep in mind that the DTW system is based on a (perfect) manual segmentation of a text line into individual words, while the HMM system operates completely automatically.

V. Conclusions

In this paper, we presented a learning-based approach for handwritten word spotting based on HMM subword models. With this approach, arbitrary keywords can be spotted and text line images are not required to be segmented into words. On the modern IAMDB as well as on the historical GWDB, the proposed system was compared with a standard template-based reference system using DTW.

Two difficult tasks were investigated for evaluating the proposed learning-based word spotting system. The first task was to spot 3,421 keywords in the unconstrained modern IAMDB database using a large amount of training data. The second task was to spot 1,067 keywords in the small historical GWDB, making use of only a small number of training data. For both tasks it was demonstrated that the proposed learning-based system clearly outperformed the template-based reference system. Note, however, that for training the HMMs a transcription is needed that is aligned with the text lines.

Future research includes the optimization of the score normalization, i.e., the filler model, the investigation of language models, and a broader comparison with other word spotting systems.

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).

REFERENCES


