Improving HMM-Based Keyword Spotting with Character Language Models

Andreas Fischer
Concordia University
CENPARMI
H3G 1M8 Montreal, Canada
an_fisch@encs.concordia.ca

Volkmar Frinken
Univ. Autònoma de Barcelona
Computer Vision Center
08193 Bellaterra, Spain
vfrinken@cvc.uab.cat

Horst Bunke
University of Bern
IAM
3012 Bern, Switzerland
bunke@iam.unibe.ch

Ching Y. Suen
Concordia University
CENPARMI
H3G 1M8 Montreal, Canada
suen@encs.concordia.ca

***Abstract*—Facing high error rates and slow recognition speed for full text transcription of unconstrained handwriting images, keyword spotting is a promising alternative to locate specific search terms within scanned document images. We have previously proposed a learning-based method for keyword spotting using character hidden Markov models that showed a high performance when compared with traditional template image matching. In the lexicon-free approach pursued, only the text appearance was taken into account for recognition. In this paper, we integrate character n-gram language models into the spotting system in order to provide an additional language context. On the modern IAM database as well as the historical George Washington database, we demonstrate that character language models significantly improve the spotting performance.**

***Keywords*—handwriting recognition; keyword spotting; hidden Markov models; language models**

I. Introduction

Automatic reading of unconstrained handwriting images that contain sentences written in natural language is still considered widely unsolved [1]. The large variety in character shapes, the large number of words occurring in natural language, and the inability to segment connected handwriting into characters before recognition lead to high error rates and slow recognition speed. If no full text transcription is required, keyword spotting has been proposed as a less demanding alternative to locate specific search terms in scanned documents [2]. Applications include the daily processing of handwritten letters sent to companies, e.g. to perform a triage based on keywords like "urgent" [3], and the processing of large collections of historical manuscripts in order to make their textual content searchable in digital libraries [4].

Traditionally, keyword spotting is approached with template matching methods that compare template images with the target documents [4]–[6]. General drawbacks of this approach include the necessity to collect templates for each keyword and the low generalization capability for new writing styles unseen in the training data. Learning-based methods, on the other hand, are able to incorporate some variance in writing style [3], [7], [8]. In particular, learning character models [7], [8] instead of word models offers the possibility to search for arbitrary keywords, even if they do not appear in the training data. We pursued this approach in our previous work [9] using hidden Markov models (HMMs) and demonstrated its high performance when compared with standard template matching.

In contrast to typical full text transcription systems, the method proposed in [9] does not require a lexicon of words for keyword spotting. By reducing the pattern space from a large number of lexicon words to a small number of alphabet characters, keyword spotting is accomplished magnitudes faster. Also, the method can be applied to old languages in historical manuscripts even if no lexicon of words is available. However, a drawback of the lexicon-free approach is that no word language model can be used to support the spotting system, which only takes the text appearance into account. Language models (LMs) have proven very useful in the past for handwriting recognition and have become a standard system component, foremost in form of word n-gram LMs [10].

In this paper, we extend our previous spotting method [9] with a language model component. Still following the lexicon-free approach, character n-gram LMs are integrated into the spotting system. On the modern IAM database, which includes English texts from different writers, as well as the historical George Washington database, we demonstrate that character LMs significantly improve the spotting performance. Furthermore, we provide a discussion of the remaining spotting errors and comment on recently published related work [7], [8].

The remainder of this paper is organized as follows. First, the HMM-based keyword spotting system is briefly reviewed in Section II. Then, the integration of character LMs is presented in Section III. Experimental results are discussed in Section IV and, finally, conclusions are drawn in Section V.

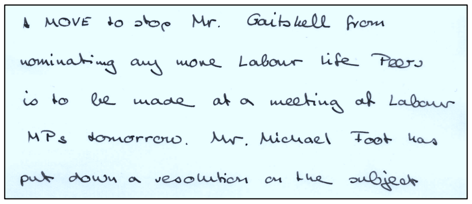
II. HMM-Based Keyword Spotting

In this section, the baseline system for keyword spotting with character hidden Markov models (HMMs) proposed in [9] is briefly reviewed. First, the data sets considered in this paper are described in Section II-A. Then, image preprocessing and feature extraction is addressed in Section II-B. Finally, the HMM-based spotting system is presented in Section II-C.

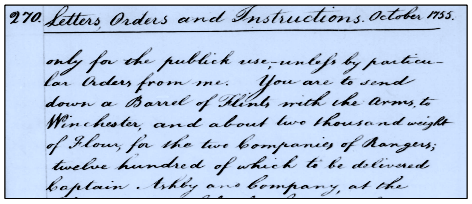
A. Data Sets

The first data set used in this paper is the IAM database (IAMDB)[1] [10] which consists of 1,539 pages of handwritten modern English text, written by 657 writers. An exemplary document image is shown in Figure 1a.

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

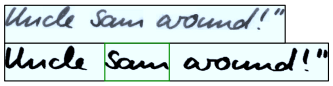

A MOVE to stop Mr. Gaitskell from
nominating any more Labour life Peers
is to be made at a meeting of Labour
MPs tomorrow. Mr. Michael Foot has
put down a resolution on the subject

(a) IAMDB

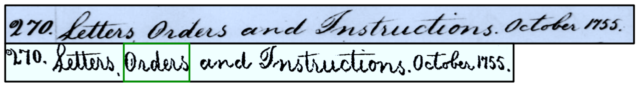

270. Letters Orders and Instructions. October 1755.
only for the publick use—unless by particu-
lar Orders from me. You are to send
down a Barrel of Flints with the Arms, to
Winchester, and about two thousand weight
of Flour, for the two Companies of Rangers;
twelve hundred of which to be delivered
Captain . Ashby and Company, at the

(b) GWDB

Fig. 1. Original data set images.


Uncle Sam around!"
Uncle Sam around!"

(a) IAMDB


270. Letters Orders and Instructions. October 1755.
270. Letters Orders and Instructions. October 1755.

(b) GWDB

Fig. 2. Text line image preprocessing; spotting "Sam" and "Orders" in the normalized images.

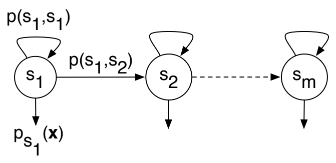
The second data set is the George Washington database (GWDB)[2] [9] which includes 20 pages of letters written by George Washington and his associates. An exemplary image is provided in Figure 1b.

B. Image Preprocessing and Feature Extraction

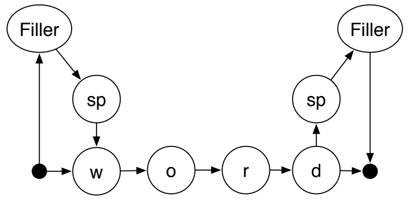
The keyword spotting system operates at text line level, that is no segmentation of text line images into words or characters is required. We ignore errors stemming from layout analysis and text line segmentation and start directly with grayscale text line images. Then, image preprocessing includes binarization, correction of the skew, i.e. the inclination of the baseline, correction of the slant, i.e. the inclination of the characters, separation of the writing region into an upper, middle, and lower part, and normalization of the width. The effect of normalization is illustrated in Figure 2a for the IAMDB and in Figure 2b for the GWDB.

In contrast to printed documents, handwritten text cannot be segmented reliably into characters before recognition. Instead, an over-segmentation is performed with a sliding window of one pixel width moving from left to right over the image. At each position, nine geometrical features are extracted including the fraction of black pixels, the first and second order moments, the contour positions, the deviation at the contours, and the number of black-white transitions.

[2]http://www.iam.unibe.ch/fki/databases/iam-historical-document-database/washington-database



(a) character model



(b) keyword text line model

Fig. 3. Hidden Markov models at character and text line level.

The image is thus represented by a feature vector sequence $\mathbf{x} = x_1, \dots, x_N$ where $x_i \in \mathbb{R}^9$ and N is the image width.

For more details on image preprocessing and feature extraction, we refer to [10].

C. Keyword Spotting

The keyword spotting system is based on character HMMs. Each character model consists of a sequence $s_1, \dots, s_m$ of hidden states as illustrated in Figure 3a. States change with probability $p(s_i, s_j)$ and emit observable feature vectors based on the probability density function $p_{s_i}(x)$ given by a mixture of Gaussians with diagonal covariance matrices. The feature distributions and state transition probabilities are trained with the Baum-Welch algorithm [11] based on transcribed text line images. For recognition, character models are concatenated to a text line HMM and processed by the Viterbi algorithm [11] which returns the best alignment of the hidden states and the HMM likelihood with respect to the trained feature distributions and state transition probabilities.

The goal of keyword spotting is to identify search terms within the text line images as illustrated in Figures 2a and 2b. In the proposed system this is achieved by calculating the likelihood ratio between two text line models. First, the general filler model F that consists of an arbitrary sequence of characters and, secondly, the keyword model K shown in Figure 3b which is further constrained to contain the search term either at the beginning, in the middle, or at the end of the text line, separated by the space character "sp". With respect to the number of states $|w|_K$ assigned to the keyword w during Viterbi recognition, the spotting confidence is given by

$$c(\mathbf{x}, w) = \frac{\log p(\mathbf{x}|K) - \log p(\mathbf{x}|F)}{|w|_K} \qquad (1)$$

The maximum value $c(\mathbf{x}, w) = 0$ is achieved if the filler model recognizes a sequence of characters that contain the keyword. In this case, the difference of the two models is zero.

Finally, the spotting confidence is compared with a threshold T and the text line image is returned as a positive match if $c(\mathbf{x}, w) \geq T$. For more details on the HMMs and the spotting system, we refer to [9].

III. Character Language Model Integration

In the spotting approach presented in Section II, the filler model represents general handwriting as an arbitrary sequence of characters, taking only the observed appearance features into account. However, characters do not appear arbitrarily in natural language. In this paper, statistical language models (LMs) are integrated into the spotting system in order to provide an additional language context.

In the following, Section III-A provides an adapted confidence function for keyword spotting with character LMs. Then, statistical n-gram LMs are discussed in Section III-B. Finally, Section III-C describes Viterbi recognition with bigram LMs.

A. Spotting Confidence

Without taking into account LMs, the original likelihood $p(\mathbf{x}|M)$ of some text line model M, for instance the keyword model in Figure 3b, can be expressed as

$$\log p(\mathbf{x}|M) = \max_{\mathbf{c} \in \mathcal{C}_M} (\log p(\mathbf{x}|\mathbf{c})) \tag{2}$$

where $\mathbf{c} = c_1, \ldots, c_{|\mathbf{c}|}$ is a character sequence of length $|\mathbf{c}|$, $\mathcal{C}_M$ is the domain of character sequences allowed by M, and $p(\mathbf{x}|\mathbf{c})$ is the HMM likelihood.

Character LMs assign the probability $p(\mathbf{c})$ to the character sequence $\mathbf{c}$ and are integrated with a modified model likelihood

$$\log p_{LM}(\mathbf{x}|M) = \max_{\mathbf{c} \in \mathcal{C}_M} (\log p(\mathbf{x}|\mathbf{c}) + \log p(\mathbf{c})) \tag{3}$$

where not only the appearance features $\mathbf{x}$ are taken into account to find the best character sequence $\mathbf{c} \in \mathcal{C}_M$, but also the language model.

In order to balance the influence of the language model, we consider two parameters which are frequently used for HMM-based handwriting recognition with statistical LMs [12], namely a grammar scale factor $\alpha \geq 0$ and an insertion penalty $\beta \in \mathbb{R}$ that controls the length of the character sequence. Taking into account these parameters, which are optimized with respect to the spotting performance on a validation set, the model likelihood amounts to

$$\log p_{\alpha,\beta}(\mathbf{x}|M) = \max_{\mathbf{c} \in \mathcal{C}_M} (\log p(\mathbf{x}|\mathbf{c}) + \alpha \log p(\mathbf{c}) + \beta|\mathbf{c}|) \tag{4}$$

By replacing the model likelihood in Equation 1 with the modified expression, we obtain the final spotting confidence

$$c_{\alpha,\beta}(\mathbf{x}, w) = \frac{\log p_{\alpha,\beta}(\mathbf{x}|K) - \log p_{\alpha,\beta}(\mathbf{x}|F)}{|w|_K} \tag{5}$$

B. Statistical n-Gram LMs

A frequently pursued approach to language modeling for handwriting recognition are statistical n-gram LMs [12] that estimate the probability of n text patterns, usually words or characters, to appear in a row in natural language. Assuming that the character probability is only dependent on its history of $n-1$ preceding characters, the probability

$$p(\mathbf{c}) = \prod_{i=1}^{|\mathbf{c}|} p(c_i|c_{i-n+1}, \ldots, c_{i-1}) \tag{6}$$

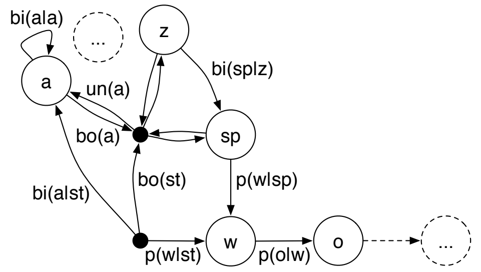


Fig. 4. Keyword text line model with smoothed character bigrams.

is assigned to the character sequence $\mathbf{c} = c_1, \ldots, c_{|\mathbf{c}|}$. The character probability $p(c_i|c_{i-n+1}, \ldots, c_{i-1})$ is estimated from text corpora, such as electronic books and newspapers.

In case of bigrams, that is $n = 2$, the maximum likelihood estimate of the character probability is

$$p_{MLE}(c_i|c_{i-1}) = \frac{\#(c_{i-1}, c_i)}{\#(c_{i-1})} \tag{7}$$

where $\#(\mathbf{c})$ is the number of occurrences of the character sequence $\mathbf{c}$ in the text corpus. In order to avoid zero probability for unseen character bigrams with $\#(c_{i-1}, c_i) = 0$, smoothing is applied [13]. Using the backoff strategy, unseen bigrams are approximated with unigram estimates $un(c_i)$ and are scaled with backoff weights $bo(c_{i-1})$ to ensure that the probabilities add up to one. The smoothed bigrams are then given by

$$p(c_i|c_{i-1}) = \begin{cases} bi(c_i|c_{i-1}), & \text{if } \#(c_{i-1}, c_i) > 0 \\ bo(c_{i-1}) \cdot un(c_i), & \text{if } \#(c_{i-1}, c_i) = 0 \end{cases} \tag{8}$$

where $bi(c_i|c_{i-1})$ is a discounted bigram estimate, allowing some leftover probability that is assigned to the unseen bigrams. In this paper, we use Witten-Bell discounting for which a detailed description can be found in [13].

The quality of a language model is typically measured by its perplexity $\mathcal{P}(\mathbf{c})$ on an independent text corpus

$$\mathcal{P}(\mathbf{c}) = p(\mathbf{c})^{-\frac{1}{|\mathbf{c}|}} \tag{9}$$

which reflects the quality of the text constraints that are imposed by the model [10]. For alphabet size A, a uniform language model with character probability $\frac{1}{A}$ achieves a perplexity of A. More accurate language models achieve lower scores $1 \leq \mathcal{P}(\mathbf{c}) \leq A$.

C. Viterbi Recognition with Bigram LMs

The original spotting confidence in Equation 1 is based on the likelihoods of two text line HMMs, that is the keyword model K and the filler model F. The likelihoods are obtained by the Viterbi algorithm (see Section II-C). In order to evaluate the modified spotting confidence in Equation 5 for smoothed bigram LMs (see Section III-B), an efficient approach is to integrate the LMs as additional transition probabilities between characters in the text line HMMs. In this case, the unmodified Viterbi algorithm can be used for conjoint optimization of $p(\mathbf{x}|\mathbf{c})$ and $p(\mathbf{c})$.

An example is shown in Figure 4 for the first part of the keyword text line model. Discounted bigram estimates

Database	Train	Valid	Test	Keywords	A	$\mathcal{P}$
IAMDB	6161	920	929	882	81	11.3
GWDB	328	164	164	84	83	10.8

TABLE I. DATABASE STATISTICS

System	IAMDB	GWDB
Reference	47.75	71.47
Bigram	55.05	74.32

TABLE II. MAP RESULTS ON THE TEST SET

$bi(c_i|c_{i-1})$ are added directly between two characters with the transition log probability $\alpha \log bi(c_i|c_{i-1}) + \beta$. In Figure 4, examples include the transition from the sentence start state "st" to the character "a" and from "z" to the space character "sp", assuming that those bigrams were seen in the text corpus. For unseen bigrams, a special backoff state is used that assigns a total log probability of $\alpha \log bo(c_{i-1}) + \alpha \log un(c_i) + \beta$ to the transition, for example between the sentence start state "st" and "z". The transitions between keyword characters are either seen or unseen bigrams and contribute a log probability of $\alpha \log p(c_i|c_{i-1}) + \beta$. Note that both the start and the backoff state are special non-emitting states with no feature distribution. They are only connecting emitting character states.

IV. Experimental Evaluation

In an experimental evaluation on the IAMDB and GWDB (see Section II-A), we compare the proposed language model extension with the original keyword spotting system. The effect is demonstrated for smoothed bigram character models (see Sections III-B and III-C).

A. Setup

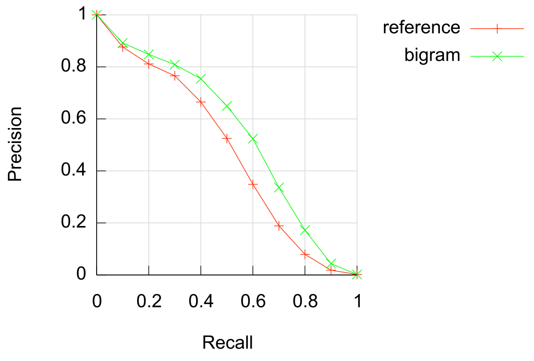
First, the text line images of the data sets are split into disjoint sets for training the character HMMs, validating system parameters, and testing the final system.[3] We spot all non-stop words that appear in all three sets. For the IAMDB, the same setting is used as in [9]. For the GWDB, we have changed the text encoding such that punctuation marks are treated as individual words, same as for the IAMDB, and are excluded from the keywords. This reduces the number of keywords and improves the results. Table I lists the data set statistics.

HMM system parameters, which are optimized on the validation set with respect to the spotting performance, include the number of states and the number of Gaussian mixtures. They are adopted from previous work [9]. Additional language model parameters include the grammar scale factor with $\alpha \in \{0, 10, \ldots, 100\}$ and the character insertion penalty with $\beta \in \{-200, -180, \ldots, 200\}$. The HTK toolkit[4] is used for Baum-Welch training and Viterbi recognition.

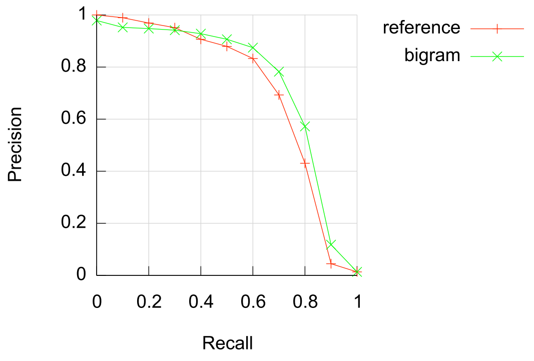
Strictly following the lexicon-free approach to keyword spotting we did not use external text corpora for language modeling which might not be available for historical languages. Instead, the character bigram LMs are estimated on the available training and validation sets. The perplexity $\mathcal{P}$ of the LMs on the test set is indicated in Table I and put into context with the alphabet size A. The SRILM toolkit[5] is used for Witten-Bell discounting.

For evaluation in a global threshold scenario, a ranked list of pairs $(\mathbf{x}, w)$ is created based on their spotting confidence

[3]For the GWDB, the mean results of a fourfold cross-validation are reported.
[4]http://htk.eng.cam.ac.uk/
[5]http://www.speech.sri.com/projects/srilm/



(a) IAMDB



(b) GWDB

Fig. 5. Spotting performance on the test set.

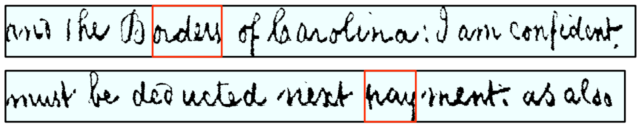
(see Section II-C). Then, beginning with the top rank, the pairs are added successively to the list of spotting results and the number of true positives (TP), false positives (FP), and false negatives (FN) are recorded at each step. This creates a recall ($\frac{TP}{TP+FN}$) and precision ($\frac{TP}{TP+FP}$) curve which captures the system performance for all possible threshold values. As a single performance value, we consider the mean average precision (MAP), that is the area under curve. The `trec_eval` software[6] is used for performance evaluation.

B. Results

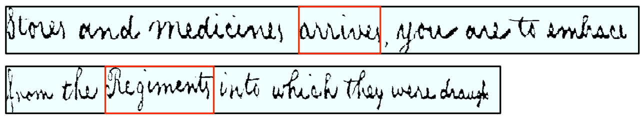
The MAP results are listed in Table II. On the IAMDB, the character LMs improve the performance by $+7.3\%$ and achieve a MAP of 55.05%. On the GWDB, the improvement is $+2.85\%$ resulting in a MAP of 74.32%. Both improvements are statistically significant (t-test over all keyword queries with $\alpha = 0.05$) and demonstrate the benefit of using character language context for HMM-based keyword spotting.

The recall–precision curves are shown in Figures 5a and 5b. While the precision is consistently improved for the IAMDB, a loss in precision is observed for low recall values, that is for top ranks, on the GWDB. It demonstrates a possible drawback

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


and the Borders of Carolina: I am confident
must be deducted next payment: as also

(a) Subwords "orders" and "pay" spotted within larger words.


Stores and medicines arrive, you are to embrace
from the Regiments into which they were draught

(b) Singular words "arrive" and "Regiment" spotted on plural words.

Fig. 6. Typical false positive errors in the top ranks for the GWDB.

of character LMs: since the character recognition is guided towards valid words with respect to the language, additional false positives may occur. The reason why this effect is only observed on the GWDB might be an overfitting of the LMs to the small training and validation sets, which together contain only 492 text lines.

C. Errors

A general weakness of the proposed keyword spotting approach is the high number of false positives in the top ranks. While character LMs improve the general spotting performance, they cannot convincingly alleviate this drawback. On the GWDB we observed, on the contrary, more false positives when using character LMs. Figures 6a and 6b show two typical error cases in the top ranks of this database.

The false positives were recently criticized by Wshah et al. in [8]. The authors of [8] propose a similar but lexicon-based keyword spotting system that derives its spotting confidence from lexicon-related alternative models instead of the filler model used in this paper. We believe this is a promising direction to address the false positives in the future.

In general, the transformation of HMM likelihoods into solid confidence measures is far from trivial. In the neural network based approach to keyword spotting we pursued in [7], the posterior probabilities directly returned by the discriminative classifier were more reliable for keyword spotting. We assume this is one of the reasons for the high spotting performance when compared with HMMs.

V. Conclusions

In this paper, we extend our previous HMM-based keyword spotting method with a language model component. Still following a lexicon-free approach, character n-gram language models are integrated into the spotting confidence.

The effect of character language models is experimentally demonstrated for smoothed bigram models. On the modern IAM database, which contains English texts from different writers, as well as on the historical George Washington database, the character language models significantly increase the spotting performance. We report a mean average precision of 55.05% ($+7.3\%$) on the IAM database, and 74.32% ($+2.85\%$) on the George Washington database.

Besides a comparison with higher order n-gram and other language models, future work includes the investigation of alternative filler models in order to reduce the number of false positives in the top ranks.

Acknowledgments

This work has been supported by the Swiss National Science Foundation fellowship project PBBEP2_141453, the Spanish project TIN2009-14633-C03-03, and the Spanish MICINN under the MIPRCV "Consolider Ingenio 2010" CSD2007-00018 project.

References

[1] G. Lorette, "Handwriting recognition or reading? What is the situation at the dawn of the 3rd millenium?" *Int. Journal on Document Analysis and Recognition*, vol. 2, no. 1, pp. 2–12, 1999.

[2] R. Manmatha, C. Han, and E. Riseman, "Word spotting: A new approach to indexing handwriting," in *Proc. Int. Conf. on Computer Vision and Pattern Recognition*, 1996, pp. 631—637.

[3] J. Rodriguez and F. Perronnin, "Handwritten word-spotting using hidden Markov models and universal vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[4] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *Int. Journal on Document Analysis and Recognition*, vol. 9, pp. 139–152, 2007.

[5] K. Terasawa and Y. Tanaka, "Slit style HOG features for document image word spotting," in *Proc. 10th Int. Conf. on Document Analysis and Recognition*, vol. 1, 2009, pp. 116–120.

[6] M. Rusinol, D. Aldavert, R. Toledo, and J. Llados, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *Proc. 11th Int. Conf. on Document Analysis and Recognition*, 2011, pp. 63–67.

[7] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *IEEE Trans. PAMI*, vol. 34, no. 2, pp. 211–224, 2012.

[8] S. Wshah, G. Kumar, and V. Govindaraju, "Multilingual word spotting in offline handwritten documents," in *Proc. 21th Int. Conf. on Pattern Recognition*, 2012, pp. 310–313.

[9] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Lexicon-free handwritten word spotting using character HMMs," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 934–942, 2012.

[10] U.-V. Marti and H. Bunke, "Using a statistical language model to improve the performance of an HMM-based cursive handwriting recognition system," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, pp. 65–90, 2001.

[11] 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, 1989.

[12] M. Zimmermann and H. Bunke, "N-gram language models for offline handwritten text recognition," in *Proc. 9th Int. Workshop on Frontiers in Handwriting Recognition*, 2004, pp. 203–208.

[13] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," *Computer Speech and Language*, vol. 13, pp. 359–394, 1999.