Adapting BLSTM Neural Network based Keyword Spotting trained on Modern Data to Historical Documents

Volkmar Frinken, Andreas Fischer, Horst Bunke
Institute of Computer Science and Applied Mathematics
University of Bern
Bern, Switzerland
{frinken, afischer, bunke}@iam.unibe.ch

R. Manmatha
Department of Computer Science
University of Massachusetts
Amherst, MA 01003-9264, USA
manmatha@cs.umass.edu

***Abstract*—Being able to search for words or phrases in historic handwritten documents is of paramount importance when preserving cultural heritage. Storing scanned pages of written text can save the information from degradation, but it does not make the textual information readily available. Automatic keyword spotting systems for handwritten historic documents can fill this gap. However, most such systems have trouble with the great variety of writing styles. It is not uncommon for handwriting processing systems to be built for just a single book. In this paper we show that neural network based keyword spotting systems are flexible enough to be used successfully on historic data, even when they are trained on a modern handwriting database. We demonstrate that with little transcribed historic text, added to the training set, the performance can further be enhanced.**

***Keywords*-Keyword Spotting, Historical Data, Handwriting Recognition, Neural Networks, Adaptation**

I. Introduction

The automatic processing of handwritten text, such as letters, manuscripts, or books has been the focus of research for several decades [1], [2]. Recently, an increasing interest in historical documents can be observed [3]. Making historical handwritten texts available for searching and browsing is of tremendous value in the context of preserving mankind's cultural heritage. Libraries all over the world store huge numbers of handwritten books and many of them would like to open the contents to the public. Searching handwritten data is a promising way to achieve that goal.

Transcribing the entire text of a handwritten document for searching is not only inefficient as far as computational costs are concerned, but it may also result in poor performance, since misrecognized words cannot be found. Therefore, techniques especially designed for the task of keyword spotting have been developed.

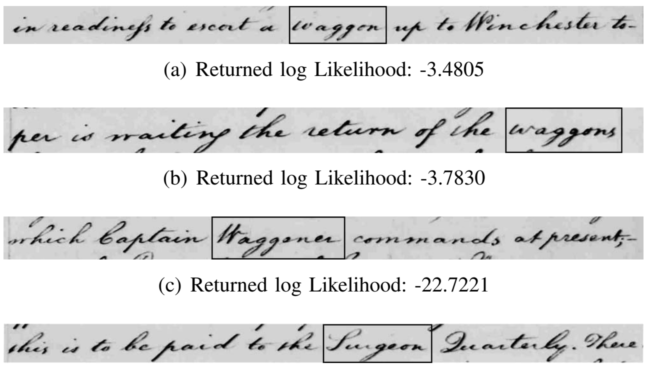
Current approaches to word spotting can be split into two categories, viz. query-by-example (QBE) and query-by string (QBS). With the former approach, all instances of the search word in the training set are compared with all word images in the test set. Among the most popular approaches in this category are dynamic time warping (DTW) [4], [5], [6] and classification using global features [7], [8]. Algorithms based on QBE suffer from the drawback that they can only find words appearing in the training set. The latter approach of QBS models the key words according to single characters in the training set and searches for sequences of these characters in the test set [9], [10]. Recently, keyword spotting systems that are modified versions of handwriting recognition systems have received increasing attention. In [10], [11], [12], hidden Markov models are used to find the words to be searched. In [13], a novel approach using bidirectional long short-term (BLSTM) neural networks is proposed. However, the performance of the neural network based keyword spotting system depends crucially on the amount of training data. Unlike modern handwritten data, a lack of neatly transcribed handwritten text is often encountered with historical handwritten data.

When dealing with handwritten historic data, certain challenges have to be faced. Ancient books or letters embody diverse writing styles and it is very common to construct recognizers for just a single book, e.g. [14]. Under such a scenario, where different documents have unique writing styles, a single database containing sufficient training data for all historic texts does not exist.

Keyword spotting systems based on underlying techniques that require a learning phase perform generally very well [15], [13], but they require large amounts of transcribed text for training. This limits their suitability for historic data. Furthermore, transcribing hundreds of lines of text to train a word spotting system is tedious and expensive, since it has to be done for every source of text.

In [16] the authors demonstrate that a HMM-based keyword spotting system for handwritten text can be improved by training certain parameters of the HMM model on printed fonts while other parameters are still trained on the handwritten text. Similarly, we propose to use modern handwriting data to train an initial spotting system based on Neural Networks. This initialized system can then be successfully adopted to historic data. We demonstrate that only a small amount of transcribed text is necessary to create a powerful keyword spotting system that reaches or even surpasses the performance of sophisticated systems specifically created for

978-0-7695-4221-8/10 $26.00 © 2010 IEEE
DOI 10.1109/ICFHR.2010.61


in readiness to escort a waggon up to Winchester to
per is waiting the return of the waggons
which Captain Waggoner commands at present
this is to be paid to the Surgeon Quarterly. There

(a) Returned log Likelihood: -3.4805

(b) Returned log Likelihood: -3.7830

(c) Returned log Likelihood: -22.7221

(d) Returned log Likelihood: -23.8951

Figure 1. Search results for the word “waggon”.

that data.

The rest of the paper is structured as follows. In Section II, the BLSTM neural networks and the preprocessing of the data are described. Details of the data and observed challenges are given in Section III, an experimental evaluation is presented in Section IV, and conclusions are drawn in Section V.

II. BLSTM Neural Network based Word Spotting

Keyword spotting refers to the process of retrieving all instances of a given word in a document. In this paper, we focus on historic handwritten letters. Without transcribing the data, a user should still be able to search for any word, just like using a search engine. How the results of such a search may look like can be seen in Fig. 1. Note that the base system just returns a likelihood of the word being found. Afterwards, this likelihood can be compared to a threshold to decide whether or not this is a true match.

A. Preprocessing

We consider complete text lines as input units for our keyword spotting system. The texts used in the experiments come from the IAM off-line database[1] [17] and George Washington DB[2] [18]. See Fig. 2 for samples of the data. After binarizing the image with a threshold on the gray scale value, the slant and skew of each text line are corrected and the width and height are normalized. Then features are extracted using a horizontally sliding window. A window with a width of one pixel is used to extract nine geometric features at each position, three global and six local ones. The global features are the 0^{th}, 1^{st} and 2^{nd} moment of the black pixels’ distribution within the window. The local features are the position of the top-most and that of the bottom-most black pixel, the inclination of the top and bottom contour of the word at the actual window position, the number of vertical black/white transitions, and the average gray scale value between the top-most and bottom-most black pixel. For details on the binarization, normalization and feature extraction steps, we refer to [19].

B. BLSTM Neural Networks

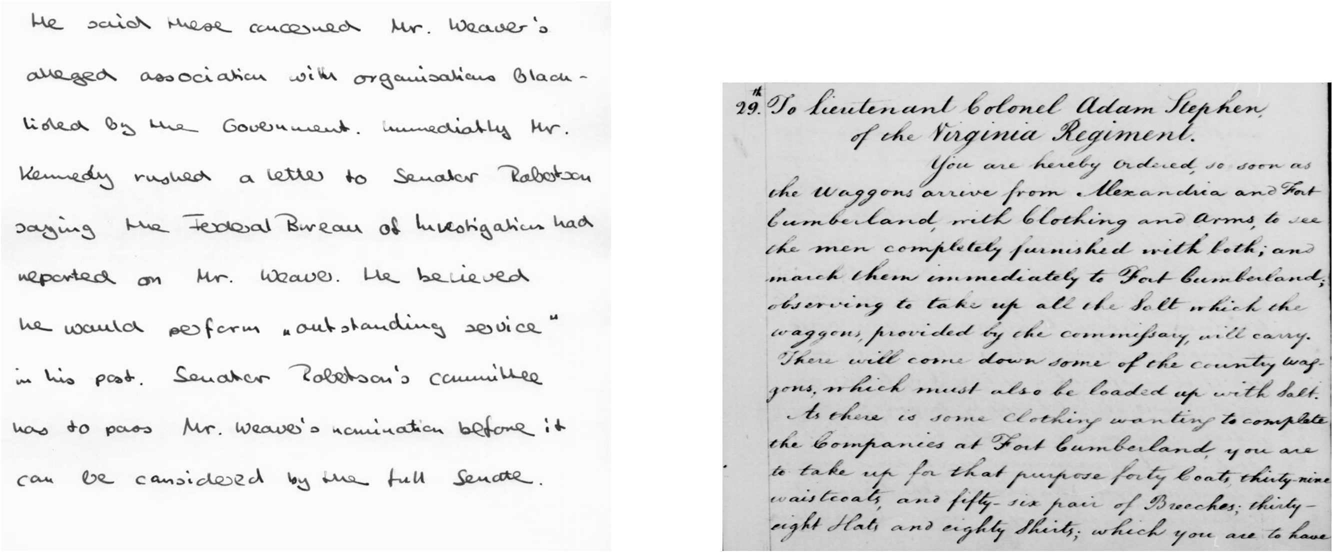
The recognizer used in this paper is a recently developed recurrent neural network, termed *bidirectional long-short term memory* (BLSTM) neural network [20]. Instead of simple nodes, the hidden layers are made up of so-called *long short-term memory* blocks. These memory blocks are specifically designed to address the *vanishing gradient problem*, which describes the exponential increase or decay of values as they cycle through recurrent network layers. This is done by nodes that control the information flow into and out of each memory block.

The input layer contains one node for each of the nine geometrical features and is connected with two distinct recurrent hidden layers. The hidden layers are both connected to the output layer. The network is *bidirectional*, i.e. a sequence of feature vectors is fed into the network in both the forward and the backward mode. The input layers consist of one node for each feature. One input and one hidden layer deal with the forward sequence, and the other input and hidden layer with the backward sequence. At each position k of the input sequence of length t, the output layer sums up the values coming from the hidden layer that has processed positions 1 to k, and the hidden layer that has processed positions t down to k. The output layer contains one node for each possible character in the sequence plus a special ε node, to indicate “no character”. At each position, the output activations of the nodes are normalized so that they sum up to 1, and are treated as a probability vector for each letter at this position. For more details about BLSTM networks, we refer to [20], [21].

The sequence of probability vectors returned by the neural network can be efficiently used for word and text line recognition as well as for word spotting [13], where the Connectionist Temporal Classification (CTC) Token Passing algorithm [20] is utilized for the latter task. In short, the probability sequence is extended by an additional entry representing the *any* character ($'*'$), having always the value 1. By adding a symbol, representing the *any* character, at the beginning and to the end of the word w to be spotted, the CTC algorithm finds the best path that passes through the *any* character, then through the word w, and then again through the *any* character. This means that the path traverses through the letters of the word w where it fits best while the rest of the text line has no influence. Then, the product of all probability values along this path is computed and divided by the keyword’s length (the number of letters in the word). The result can be interpreted as the likelihood

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

[2] 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



He said these concerned Mr. Weaver's
alleged association with organisations Black-
listed by the Government. Immediatly Mr.
Kennedy rushed a letter to Senator Robertson
saying the Federal Bureau of Investigation had
reported on Mr. Weaver. He believed
he would perform „outstanding service"
in his post. Senator Robertson's committee
has to pass Mr. Weaver's nomination before it
can be considered by the full Senate.

(a) IAM database

29th To Lieutenant Colonel Adam Stephen,
of the Virginia Regiment.
You are hereby Ordered, so soon as
the Waggons arrive from Alexandria and Fort
Cumberland with Clothing and Arms, to see
the men completely furnished with both; and
march them immediately to Fort Cumberland;
observing to take up all the Salt which the
waggons, provided by the commiſsary, will carry.
There will come down some of the county wag-
gons, which must also be loaded up with Salt.
As there is some Clothing wanting to complete
the Companies at Fort Cumberland, you are
to take up for that purpose forty Coats, thirty-nine
waistcoats, and fifty-six pair of Breeches; thirty-
eight Hats and eighty Shirts; which you are to have

(b) GW database

Figure 2. Samples from the two databases used in the experiments.

that this word is contained in the considered text line. For more details about the keyword spotting algorithm we refer to [13].

III. Adaptation

When different data sets are used for training and testing, several problems occur. This is especially true when the data sets originate from different geographic locations or periods of times, like the IAM and GW database. Not only the writing style is different, but also different characters can be observed. Among writing style differences are the positions of the ordinal indicator like 'st' in '1st', which may occur on the base line, as a superscript or above the number. See Fig. 3(a) for samples from the GW database where ordinal indicators are written above the number. A character that frequently appears in historic texts but which is not used any more is the 'long s'. An example of the word "possible' from the GW and IAM database can be seen in Fig. 3(b). Another obstacle are signatures, abbreviations or symbols. Fig. 3(c) gives an example of the abbreviation "&c." for "etc." and Washington's signature.

The way we handle these special cases is by endowing the neural network with a 'garbage' output node. When the network is trained on the IAM-DB, infrequent characters, such as '#' or '*', are mapped to the 'garbage'-model. Then, for adaptation, all unrecognizable characters mentioned above are mapped to the garbage model. Large differences on the morphology of some keywords do not constitute a problem, as long as nodes for the each character of the keyword exist. The system can be seen as being bootstrapped using modern handwritten data and refined using historical data. As demonstrated in this paper, only a small amount of the historical data is needed for that process.

Another point worth mentioning regards feature normalization. The activation function of the input nodes of the neural network require all features to have a mean of 0 and a variance of 1. Both, mean and variance have to be recomputed on the historical dataset.

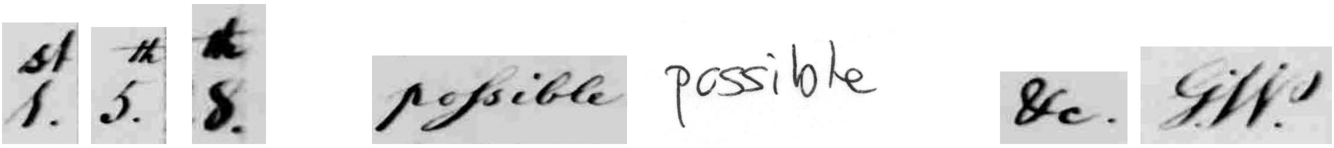
IV. Experimental Evaluation

A. Setup

The experiments we conducted involved modern handwritten data and historic data. In a first set of experiments we analyzed the keyword spotting performance of neural networks that were trained on the IAM database and tested without modifications on the GW database. We trained 50 neural networks using a training set of 6161 text lines and a writer independent validation set of 920 text lines. Due to the random initialization of the neural networks, a great variance in the networks' performance can be observed. Hence, the validation set and several thousand key words were used to identify the best network. This network is not necessarily the best one on the test set, but we have shown in [13] that usually a good selection can be made this way.

Afterwards we explored the application of a second training phase, using different amounts of training data. In the first adaptation experiment, two pages of transcribed text are necessary, one page that acts as a training set and the other as a validation set. The second adaptation experiment requires five pages of transcribed historic data. Two pages were used as the training and three as the validation set.

To make test results coherent and comparable, we used 4-fold cross validation. The GW database consists of 20 pages,



(a) Ordinal indicator above numbers

(b) The ‘long s’ is not used any more

(c) Abbreviations and signatures

Figure 3. Special characters

which are divided into four parts of five pages each. We used one part for training and validation and the remaining 15 pages for testing. The average results are reported.

Finally the results of two reference systems are given as well. The first one is a BLSTM NN based keyword spotting system trained entirely on the historic data. The other reference system is a HMM based keyword spotting system which was also trained on historic data exclusively [15]. Note that for both reference system, 10 pages were used for training, 5 for validation and 5 for testing, also in a four fold cross validation. That means that ten, resp. three times as much transcribed text was used.

For testing we aimed at spotting every word contained in the GW database that is not a stop word. Stop words are words that do not contribute much valuable information and are used more for structuring the text than carry information, like “the”, “a” or “although”. We used the stop word list[3] from the SMART project [22]. All in all, the list of keywords to be spotted includes 1067 entries.

B. Results

Each word tested on a text line returns a probability. The word spotting algorithm compares this probability against a global threshold to decide whether or not it is a match. We used all returned values as a global threshold in oder to make the results as precise as possible. For each of these thresholds, we computed the number *true positives* (TP), *true negatives* (TN), *false positives* (FP), and *false negatives* (FN). These number were then used to plot *recall-precision* values. *Precision* is defined as number of relevant objects found by the algorithm divided by the number of all objects found $\frac{TP}{TP+FP}$, while *recall* is defined as the number of relevant objects found divided by the number of all relevant objects in the test set $\frac{TP}{TP+FN}$. Due to the high number of tested keywords and hence different thresholds, the scatter plot can be considered as a continuous curve.

In Fig. 4, three *recall-precision* curves can be seen. These curves are the average over all cross validations runs of the performance of the best network, as determined on the validation set. The bottom-most curve displays the performance of the initial experiment, when no data from the GW DB

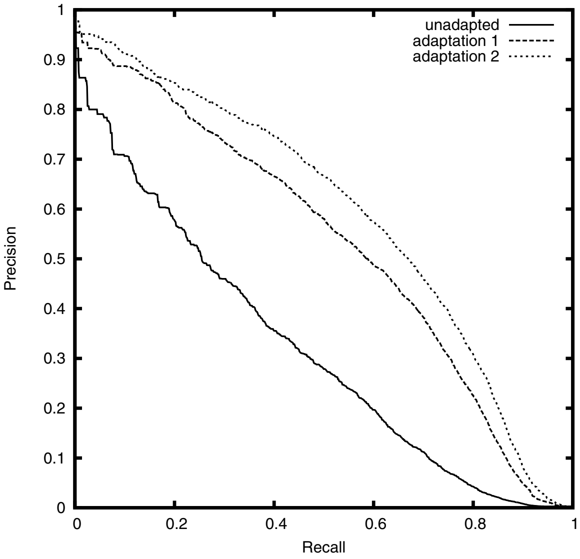


Figure 4. *recall-precision* curve of the different adaptation approaches

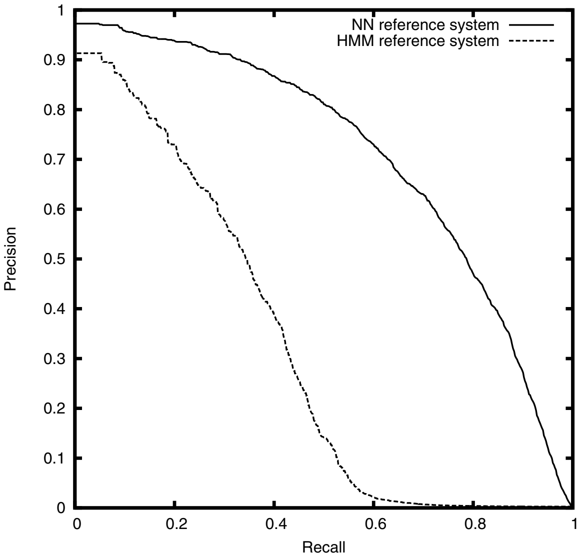


Figure 5. *recall-precision* curve of the reference systems

[3] http://jmlr.csail.mit.edu/papers/volume5/lewis04a/a11-smart-stop-list/english.stop

is used for training. The curve in the middle displays the performance of the adaptation approach that requires two pages of transcribed data and the top-most curve displays the performance of the other adaptation approach that requires five transcribed pages. The *recall-precision* curve of the two reference systems can be seen in Fig. 5.

A common measure to compare different *recall-precision* curves is to consider the mean average precision (*map*), which is the mean of the areas under the curves. The following table lists the results using this measure. Note that 'best *map*' means the mean average precision for the network that performed best on the validation set.

setup	average *map*	best *map*
initial experiment	0.28	0.31
adaptation 2 pages	0.50	0.53
adaptation 5 pages	0.57	0.59
HMM reference	0.32	
NN reference	0.60	0.71

The neural network trained on the IAM database performed only slightly inferior to the HMM based keyword spotting system, trained entirely on the GW Database. Unsurprisingly, the more data is used for adapting the neural networks to the current writing style, the better they perform. The average performance of networks trained entirely on the GW database is nearly met when using five pages of historic data for the second training phase. Both adaptation methods clearly outperform the HMM reference system.

V. Conclusion

We have shown in this paper that it is possible for neural network based keyword spotting systems to be trained on modern handwriting data, even when they are used on a completely different, historic data set. We have explored the possibility to adapt the networks to the historic data by using a very small portion of transcribed data. A system created this way outperforms one of our reference systems, even though the reference system was trained entirely on the historic data set.

We have proven that, due to their flexibility, BLSTM based keyword spotting system can be very useful to spot keywords when little or no transcription of the historic data or the specific writing style is available. In the future, we are looking into unsupervised adaptation techniques in the form of self-learning, to further explore the applicability of this keyword spotting approach.

Acknowledgments

This work has been supported by the Swiss National Center of Competence in Research (NCCR) on Interactive Multimodal Information Management (IM2) and the the Swiss National Science Foundation (Project CRSI22_125220/1). We thank Alex Graves for kindly providing us with the BLSTM Neural Network source code.

References

[1] A. Vinciarelli, "A Survey On Off-Line Cursive Word Recognition," *Pattern Recognition*, vol. 35, no. 7, pp. 1433–1446, 2002.

[2] R. Plamondon and S. N. Srihari, "On-Line and Off-Line Handwriting Recognition: A Comprehensive Survey," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 22, no. 1, pp. 63–84, 2000.

[3] A. Antonacopoulos and A. C. Downton, "Special issue on the analysis of historical documents," *IJDAR*, vol. 9, no. 2–4, pp. 75–77, 2007.

[4] A. Kołcz, J. Alspector, M. F. Augusteijn, R. Carlson, and G. V. Popescu, "A Line-Oriented Approach to Word Spotting in Handwritten Documents," *Pattern Analysis and Applications*, vol. 3, pp. 153–168, 2000.

[5] R. Manmatha and T. M. Rath, "Indexing of Handwritten Historical Documents - Recent Progress," in *Symposium on Document Image Understanding Technology*, 2003, pp. 77–85.

[6] T. M. Rath and R. Manmatha, "Word Image Matching Using Dynamic Time Warping," in *Computer Vision and Pattern Recognition*, vol. 2, 2003, pp. 521–527.

[7] E. Ataer and P. Duygulu, "Matching Ottoman Words: An Image Retrieval Approach to Historical Document Indexing," in *6th Int'l Conf. on Image and Video Retrieval*, 2007, pp. 341–347.

[8] Y. Leydier, F. Lebourgeois, and H. Emptoz, "Text Search for Medieval Manuscript Images," *Pattern Recognition*, vol. 40, pp. 3552–3567, 2007.

[9] H. Cao and V. Govindaraju, "Template-free Word Spotting in Low-Quality Manuscripts," in *6th Int'l Conf. on Advances in Pattern Recognition*, 2007.

[10] J. Edwards, Y. Whye, T. David, F. Roger, B. M. Maire, and G. Vesom, "Making Latin Manuscripts Searchable using gHMM's," in *Advances in Neural Information Processing Systems (NIPS) 17*. MIT Press, 2004, pp. 385–392.

[11] H. Jiang and X. Li, "Incorporating training errors for large margin hmms under semi-definite programming framework," *Int'l Conf. on Acoustics, Speech and Signal Processing*, vol. 4, pp. 629–632, April 2007.

[12] F. Perronnin and J. Rodriguez-Serrano, "Fisher Kernels for Handwritten Word-spotting," in *10th Int'l Conf. on Document Analysis and Recognition*, vol. 1, 2009, pp. 106–110.

[13] V. Frinken, A. Fischer, and H. Bunke, "A Novel Word Spotting Algorithm Using Bidirectional Long Short-Term Memory Neural Networks," in *4th Workshop on Artificial Neural Networks in Pattern Recognition*, 2010.

[14] V. Romero, A. H. Toselli, L. Rodrguez, and E. Vidal, "Computer assisted transcription for ancient text images," in *Proc. of 4th Int'l Conf. on Image Analysis and Recognition*, ser. LNCS, vol. 4633, 2007, pp. 1182–1193.

[15] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "HMM-Based Word Spotting in Handwritten Documents Using Subword Models," in *20th Int'l Conf. on Pattern Recognition, accepted for publication*, 2010.

[16] J. Rodríguez-Serrano, F. Perronnin, J. Lladós, and G. Sánchez, "A Similarity Measure Between Vector Sequences with Application to Handwritten Word Image Retrival," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009, pp. 1722–1729.

[17] U.-V. Marti and H. Bunke, "The IAM-Database: An English Sentence Database for Offline Handwriting Recognition," *Int'l Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, 2002.

[18] T. M. Rath and R. Manmatha, "Word spotting for historical documents," *Int'l Journal of Document Analysis and Recognition*, vol. 9, pp. 139–152, 2007.

[19] U.-V. Marti and H. Bunke, "Using a Statistical Language Model to Improve the Performance of an HMM-Based Cursive Handwriting Recognition System," *Int'l Journal of Pattern Recognition and Artificial Intelligence*, vol. 15, pp. 65–90, 2001.

[20] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A Novel Connectionist System for Unconstrained Handwriting Recognition," *IEEE Transaction on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[21] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist Temporal Classification: Labelling Unsegmented Sequential Data with Recurrent Neural Networks," in *23rd Int'l Conf. on Machine Learning*, 2006, pp. 369–376.

[22] G. Salton, *The SMART Retrieval System—Experiments in Automatic Document Processing*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1971.