2014 14th International Conference on Frontiers in Handwriting Recognition

Handwriting Normalization by Zone Estimation using HMM/ANNs

Joan Pastor-Pellicer*, Salvador España-Boquera*, Francisco Zamora-Martínez†, María José Castro-Bleda*
{*jpastor, sespana, mcastro*}*@dsic.upv.es*, francisco.zamora@uch.ceu.es
* *Departamento de Sistemas Informáticosy Computación*
Universitat Politècnica de València
Valencia, Spain
† *Dep. Ciencias Físicas, Matemáticas y de la Computación*
Universidad CEU Cardenal Herrera
Alfara del Patriarca, Valencia, Spain

***Abstract*—Offline handwritten text recognition requires several preprocessing stages. Many different preprocessing techniques have been proposed in the literature based either on geometrical heuristics or on statistical models. Unfortunately, these approaches usually fail when dealing with short sentences or isolated words. One statistical technique for text line preprocessing is based on the detection and classification of local extrema points, by means of neural networks, to determine the reference lines delimiting the different zones. This technique depends on a sufficient amount of local extrema and, relating its robustness, a single bad classified extrema point may lead to undesirable results.**

This paper proposes a novel method to normalize handwritten text lines based on a supervised statistical model which takes into account all pixels instead of just the local extrema. A Hidden Markov Model hybridized with an Artificial Neural Network is applied column-wise in order to segment each column of the handwritten line into three zones. The reference lines obtained in this way are used to normalize the image afterwards. The technique has been empirically tested on the IAM offline database.

***Keywords*-handwriting text line normalization; zone estimation; hybrid HMM/ANN**

I. Introduction

Offline handwritten text recognition remains a challenging pattern recognition task. One of the reasons is the high variability of writing styles. The preprocessing stage of automatic handwriting recognition systems usually comprise several steps in order to reduce variations in the handwritten texts as much as possible. Some of these steps are illustrated in Figure 1. Once the text line image is detected, this preprocessing typically relies on slope correction, slant correction and size normalization. With the slope correction, the handwritten word is horizontally rotated and translated such that the lower baseline is aligned to the horizontal axis of the image. Slant is the clockwise angle between the vertical direction and the direction of the vertical text strokes. Once the slant is corrected, size normalization tries to make the system invariant to the characters size.

Most preprocessing modules comprise the detection of the different zones of the cursive script depicted in Figure 2: the main body or core zone (zone between the upper baseline and the lower baseline), the zone of the ascenders, and the zone of the descenders. Traditional methods [1], [2], [3] obtain a rough estimate of the main body zone by horizontal density histograms [1], [2] or by applying the "Run-Length Smoothing Algorithm" [4].

Figure 2. Example of text line image with the different zones (zone of ascenders, zone of descenders, and main body or core zone) and the reference lines delimiting them (upper and lower baselines, and the lines of ascenders and descenders) of the cursive script.

These techniques are based on geometrical heuristics which state that there are more pixels in the main body zone than in the zone of ascenders or descenders. The use of local extrema for estimating the zones can be also found in the literature. For example, in [5] the vertical extreme values are used to estimate the baseline by selecting the subset of baseline points using regression analysis. Unfortunately, geometrical heuristics may fail in many cases, specially in short sentences or in isolated words. Since they are based on ink statistics, they may be confused in presence of too much or too few ascenders and/or descenders, and they can also be affected by the presence of long horizontal strokes.

It is possible to estimate the main body zone avoiding geometrical heuristics, relying instead on machine learning techniques. For instance, [6] uses neural networks to obtain a rough estimate of the main body zone. The use of neural networks to determine the reference lines by classifying local extrema points has already been proposed in [7], [8]. This method may fail when there are few local extrema, as is the case short sentences and isolated words, or in the presence of a sole bad classified point. Our goal is to obtain a more robust method.

This paper presents a new technique based on Hidden Markov Models (HMMs) to track the reference lines without requiring the detection and classification of local extrema

2167-6445/14 $31.00 © 2014 IEEE
DOI 10.1109/ICFHR.2014.111

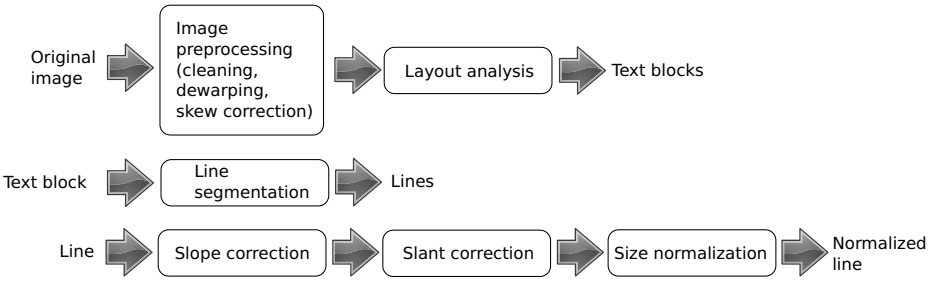


Figure 1. Bird's eye view of handwriting recognition preprocessing stages from the scanned document to preprocessed handwritten lines.

points. HMMs play an important role in handwriting recognition, but their role is mainly limited to the recognition of preprocessed lines. Although the use of HMMs in preprocessing is not new, it seems to have been limited to the segmentation of lines from printed documents [9] and, more recently, extended to cursive script [10].

Next section describes our approach, showing illustrative examples. The experimental framework including the description of the whole recognition system and the training of the system using the IAM corpus are presented in Section III. Some evaluation metrics are shown in Section IV. Finally, the conclusions and new proposals for future work are drawn in Section V.

II. Text Size Normalization by Zone Estimation

Text size normalization is closely related to the detection of the different zones of the cursive script. As stated in the previous section, most approaches are based either on geometrical heuristics or on machine learning techniques. The novel size normalization method proposed in this work is based on statistical machine recognition techniques and relies also on tracking reference lines. It differs from [7], [8] in the way those lines are obtained: Instead of joining local extrema associated to the same class, the pixels are classified into the different zones so that the reference lines are the frontiers between them.

A. Statistical Framework

The zone detection problem can be formulated as a joint pixel classification problem into three classes $\{A, B, D\}$ for zone of Ascenders, main Body and Descenders, respectively. This classification shows some restrictions: If we focus our attention on a given column, the pixels of the same zone are contiguous and the classes follow a vertical order (from top to bottom: A, B and D, as shown in Figure 2). The zone estimation, posed in this way, can be easily formulated as a statistical pattern recognition problem. More particularly, if this process is applied column-wise, the problem is a joint classification on sequences, which can be tackled by means of HMMs or also with Conditional Random Fields [11]. Indeed, both models provide the capability of combining syntactic restrictions with the estimation of likelihoods or posteriors of each pixel from some features. In this work, we have opted to use HMMs hybridized with Artificial Neural Networks (ANNs) since connectionist models have widely proven their suitability for image processing [12].

More formally, given a text line image of width w and height h, each one of the w image columns can be described as a sequence of pixels $X = (x_1 \dots x_h)$ and, under the statistical approach to pattern recognition [13], the goal is to find the likeliest zone sequence $Z^\star = (z_1 \dots z_h)$ maximizing the a posteriori probability:

$$Z^\star = \underset{Z \in \{A,B,D\}^h}{\operatorname{argmax}} \; P(Z|X) \ . \tag{1}$$

The application of the Bayes rule leads to a decomposition of $P(Z|X)$ into the model $P(X|Z)$ and the statistical model describing the a priori probability of zones $P(Z)$. The problem can then be reformulated as:

$$Z^\star = \underset{Z \in \{A,B,D\}^h}{\operatorname{argmax}} \; P(X|Z)P(Z) \ . \tag{2}$$

The emission probabilities of HMMs, usually estimated by means of Gaussian mixtures, are computed in this case by means of ANNs. Since these models estimate posteriors, we need to convert to emissions by applying Bayes rule [14].

B. Modeling

A very simple left-to-right with loops HMM topology, as depicted in Figure 3, suffices to model the a priori probability of zones given by the constraints about the possible sequence of zone labels. Each one of the three emitting states corresponds to one of the three zones. The fact that some lines do not have ascenders or descenders is modeled by allowing their respective states to be skipped.

The state emissions are estimated with a multilayer perceptron (MLP) which receives a centered window around the pixel to be classified and classifies the pixel in one of the three zones. To this end, the softmax activation function is used at the output layer. An alternative consists of determining whether the pixel belongs to the main body zone or not, without discriminating between ascender and descender zones. In this case, the emissions of the states A and D are tied and a single logistic output neuron suffices.

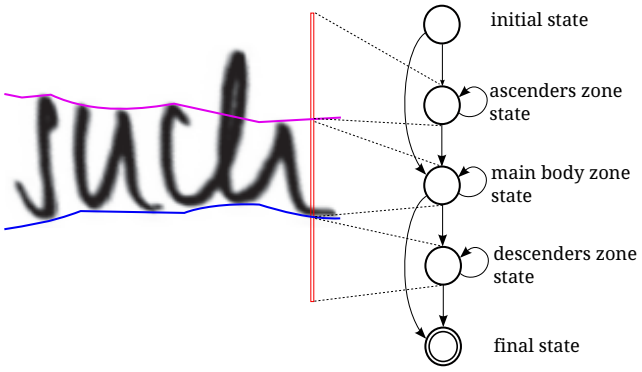


Figure 3. Scheme of the used HMM topology: one state for each zone $\{A, B, D\}$ and skips for ascenders and descenders.

C. Smoothing of Reference Lines

During training and for preprocessing, the HMM/ANN described above is independently applied to each column of the image to segment it into three zones. White-space columns, skipped during training, are also skipped in the preprocessing stage. The segmentation points of nearby columns are usually very similar since their input features are highly correlated. Nevertheless, some irregularities may be observed. In order to cope with these variations as well as to assign segmentation points in white-spaces columns, the upper and lower contours are interpolated and smoothed.

Note that the HMM/ANN only detects the main body zone, and the upper and lower contours are extracted from there. In order to estimate the reference line of ascenders and descenders, the local extrema of the pixels of the corresponding zones are used. The highest local maxima of the vertical upper contour are used for ascenders and the local minima of the lower contour for descenders. In addition some restrictions are applied:

- local extrema of the contours must be more than 5 pixels off the main body zone,
- only one local extrema of each class could appear in each column. In that case, the furthest point from the main body zone is taken.

The final reference lines are determined joining ascenders and descenders respectively by means of lineal interpolation as can be appreciated in Figure 2.

D. Text Size Normalization

Finally, once the reference lines are obtained, normalization is performed for each column of the image by linearly scaling the three zones to a fixed height. Ascender zone is reduced to 20% of the final image height and the descender zone is reduced to 10%. This produces a fixed height image suitable for the feature extraction described below. The image height has been fixed to 42 pixels as in [8]. Note that although the size normalization process does not preserve the aspect ratio, it is still possible to perform a width normalization by counting, for instance, the average number of changes of white-space/ink, in horizontal, in the main body zone rows and using this value, estimated in the training corpus and normalized per pixel, to scale the width.

III. Experimental Framework

A. Corpus

Since this work is focused on the text normalization step, the system has been tested with a corpus of segmented lines. A handwriting recognition experiment with the version 3.0 of the IAM database [15], using the standard training and test partitions, has been conducted. This version consists of 5 685 sentences comprising about 115 000 word instances produced by 657 writers, without restrictions on the writing style or the writing instrument used.

B. Image Preprocessing

The image cleaning, slope and slant removal are identical to [8], whereas the zone estimation, the tracking of reference lines and the text normalization applied afterwards are those described in previous section. An example of the preprocessing step is illustrated in Figure 4. The recognition experiments conducted with the resulting preprocessed and parametrized lines uses the same type of HMM/ANN models as in [8].

C. Size Normalization

The HMM/ANN are applied in this step to downsized images in order to reduce the input parameters of the MLP window and to speedup the process. The images have been downsized to 50% and the segmentation points are applied to the original images. We have also observed that applying the technique every two columns produces nearly the same results.

Let us now describe how the HMM/ANN parameters (MLPs, class prior probabilities and the HMM transition probabilities) have been estimated from the IAM database: In order to obtain labeled patterns (pixels belonging to one of the three zones to classify), the training and validation lines of the IAM database have been automatically segmented into zones by using the techniques described in [8]. White-space columns are not taken into account. Let us observe that, since this artificially generated ground-truth has been computed automatically, it may contain some mistakes so that the training may be biased. But note also that, as in [6], the generalization capability of MLPs may partially tackle this issue.

MLPs have been trained using backpropagation with momentum term, weight decay and bunch mode using the April-ANN toolkit [16]. Each training is stopped when the error measured on a validation set does not improve during more than 40 training epochs. Several MLPs have been trained varying different parameters using the technique of

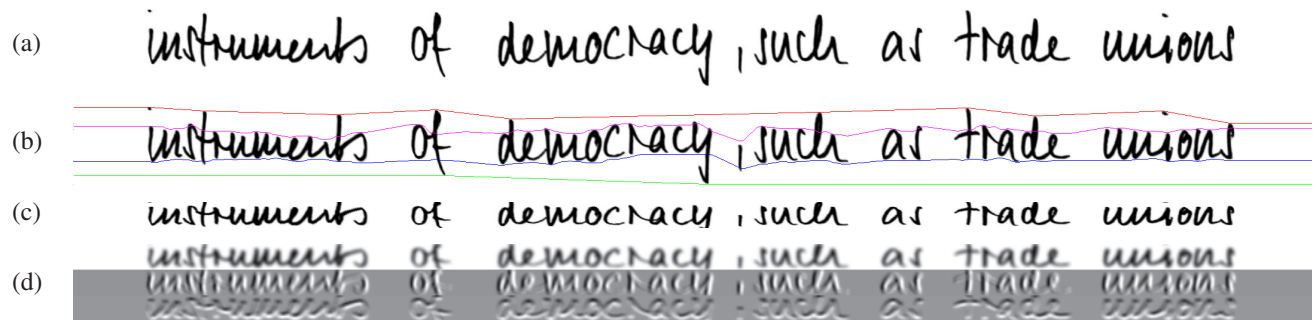


Figure 4. Example of a IAM text line image. From top to bottom: (a) original image, (b) image with reference lines computed with the proposed technique, (c) image after text size normalization, and (d) sequence of feature vectors.

hyper-parameter random optimization [17]. The explored parameters include:

- the learning rate and the momentum term are taken from a log-uniform distribution between 0.001 and 0.5,
- the loss function is chosen from cross-entropy and mean squared error,
- the weight decay is chosen from $\{10^{-5}, 10^{-6}, 10^{-7}\}$,
- the size of the first hidden layer (chosen from $\{64, 126, 256, 512\}$) and the second hidden layer (chosen from $\{16, 32, 64, 128\}$), with the restriction that the first must be greater than the second,
- the activation function of the hidden neurons is taken from hyperbolic tangent and logistic,
- the size of the bunch mode has been taken from $\{16, 32, 64\}$,
- the window size to model the input pixel window: the width and the heights are chosen independently to be $2 \times n + 1$ pixels, for $n \in \{10, 20, 25, 30, 35, 40\}$,
- two or three output classes.

The best configuration for this experimental setting was:

- the input layer receives a window of pixels of width $2 \times 35 + 1$ and height $2 \times 35 + 1$ centered at the pixel to be classified as well as the vertical distance to the upper and lower vertical contour of columns of a window of the same width,
- two hidden layers of sizes 256 and 64, respectively,
- learning rate 0.15, momentum term 0.2 and weight decay 10^{-6},
- the output layer comprises three softmax units.

The class priors $P(z)$ have been estimated from the relative frequencies of each class from the training data. The HMM transition probabilities has been estimated from the same data by simulating forced Viterbi alignment. Note that, in this case, it is not necessary to perform an expectation maximization procedure: the re-segmentation step is not required since the artificial ground-truth used is labeled at the pixel level and we are dealing with a joint classification at this level.

The segmentation points of each column, obtained from the downsized image, are scaled to the original line image and the points of each reference line are interpolated using the `interpolate.splrep` routine from Python Scipy [18] (linear interpolation with $k = 1$) and are smoothed afterwards by means of a convolution with a Hanning window (of width 15).

D. Feature Extraction

The preprocessed image is then transformed into a sequence of feature vectors following the approach described in [19]. A sliding window of square cells is applied on the image, and three values are extracted from each cell: normalized gray level, horizontal derivative of the gray level, and vertical derivative of the gray level. A window comprising 20 cells moving in steps of two pixels has been used, leading to sequences of 60-dimensional feature vectors. Figure 4(d) shows a feature extraction example.

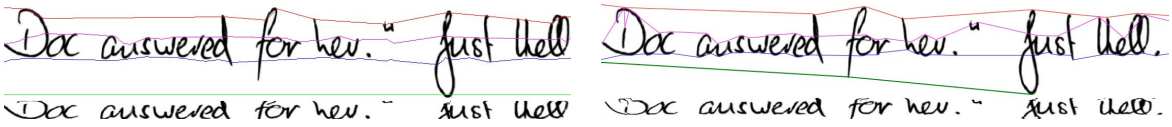
E. Recognition

The underlying recognition engine is based on HMM/ANNs similar to [8]. The HMM models graphemes have a 7-state left-to-right topology with loops and without skips. A MLP with two hidden layers of 256 and 128 units using the softmax activation in the output layer is used to estimate the posteriors which are converted into scaled emission probabilities, as described in [14]. The input of the MLP was composed by the current feature vector plus a context of 5 vectors on the left and 5 on the right. The HMM/ANNs are trained by means of EM procedure with forced Viterbi alignment.

The language model and lexicon used in this work are the same as in [25]. The language model is a Witten Bell smoothed 4-gram trained with the SRILM toolkit [26]. Three different text corpora were used: the LOB corpus [27] (excluding those sentences that contain some line from the test set or the validation set of the IAM task), the Brown corpus [28], and the Wellington corpus [29]. The lexicon has approximately 103K different words.

Table I
WER AND CER FOR THE TEST SET OF IAM DATABASE.

System	$\|\Omega\|$	WER (%)	CER (%)
Bertolami et al. (GHMM) [20]	20K	35.5	-
Bertolami et al. (HMMs) [20]	20K	32.8	-
Drew et al. (GHMM) [21]	50K	29.2	10.3
TU Dortmund (HMM) [22]	-	28.9	-
Drew et al. (MLP-GHMM + M-MPE) [23]	50K	28.8	10.1
Graves et al. (BLSTM NN/CTC) [24]	20K	25.9	-
España-Boquera et al. (HMM/ANN) [8]	20K	25.9	10.5
HMM Zone estimation (HMM/ANN)	**103K**	**24.4**	**10.6**
Zamora et al. (BLSTM) [25]	103K	22.2	10.5
Zamora et al. (HMM/ANN) [25]	103K	21.1	8.6



(a) (b)

Figure 5. Example of IAM image normalization: (a) applying HMM's/ANN zone detection, (b) detecting the reference lines using extrema points.

IV. EXPERIMENTAL RESULTS

The results are compared by means of the Word Error Rate (WER) and the Character Error Rate (CER). A WER of $18.6(\%)$ and a CER of $6.9(\%)$ were obtained for the validation set. For the test set, the obtained WER was 24.4 and the corresponding CER was 10.6. These results are shown in Table I along with the performance of other recognition systems reported in the literature. Let us remark that the results obtained in this work share the same lexicon and language model of [25]. As can be observed, the new results do not reach the best figures but they remain competitive with state-of-the-art systems.

Since the ground-truth used to train the zone estimation technique is obtained from the local extrema classification technique used in [8], it is expected that some mistakes had been inherited from it. This may explain that the proposed method does not reach the figures of merit from the models used to obtain the artificial ground-truth. Nevertheless, we have observed that the proposed zone estimation is able to recover some errors generated by the local extrema classification technique, as shown in Figure 5.

V. CONCLUSIONS AND FUTURE WORK

A new technique to normalize handwritten text line images by zone estimation using HMM/ANNs has been proposed. The estimation of the different zones (ascender, main body and descender zones) is computed pixel-wise by applying a HMM/ANN to each column of the text line image. This information is used to normalize the lines after correcting the slope and the slant using the techniques described in [7], [8]. Nevertheless, the application of the technique proposed in this work for slope normalization is straightforward. The proposed technique has been tested on the IAM offline database and the achieved results are competitive with state-of-the-art systems.

In the proposed technique, each column is labeled into zones independently of their neighbors. A smoothing has been applied afterwards. It would be desirable to investigate methods taking the column correlation into account in the optimization process. Other lines of research are the use convolutional neural networks [30], [31] for the estimation of the posterior probabilities and the use of Conditional Random Fields [11], [32] instead of HMMs. Additionally, more complex HMM topologies could be used to better model the distribution of vertical lengths of each zone.

As a future work, we plan to combine this new technique with other approaches such as the classification of local extrema by neural networks [7] in order to override the weakness of both approaches and obtaining a more robust system. Also, we intend to apply the zone estimation based on HMMs proposed in this paper to perform line extraction and line normalization simultaneously following the ideas from [9], [10]. The basic idea consists of including a loop in the HMM and to apply the model to the columns of the entire image document.

ACKNOWLEDGMENT

This work has been supported by the Spanish Government under project TIN2010-18958.

REFERENCES

[1] D. J. Burr, "A normalizing transform for cursive script recognition," in *Proc. 6th Int. Conf. Pattern Recognition*, Munich, Germany, 1982, pp. 1027–1030.

[2] R. M. Bozinovic and S. N. Srihari, "Off-line cursive script word recognition," *IEEE Trans. on PAMI*, vol. 11, no. 1, pp. 68–83, 1989.

[3] A. Vinciarelli and J. Luettin, "A new normalization technique for cursive handwritten words," *Pattern Recognition Letters*, vol. 22, no. 9, pp. 1043–1050, 2001.

[4] K. Y. Wong, R. G. Casey, and F. M. Wahl., "Document Analysis system," *IBM Journal of Research and Developement*, vol. 26, no. 6, pp. 647–655, 1982.

[5] T. Caesar, J. Gloger, and E. Mandler, "Estimating the baseline for written material," in *Proc. 3rd Int. Conf. Document Analysis and Recognition*, vol. 1, 1995, pp. 382–385.

[6] R. Seiler, M. Schenkel, and F. Eggimann, "Off-line Cursive Handwriting Recognition Compared with On-line Recognition," in *Proc. 13th Int. Conf. on Pattern Recognition*, Vienna, Austria, 1996, pp. 505–509.

[7] J. Gorbe-Moya, S. España-Boquera, F. Zamora-Martínez, and M. J. Castro-Bleda, "Handwritten Text Normalization by using Local Extrema Classification," in *Proc. 8th Int. Workshop on Pattern Recognition in Information Systems*, Barcelona, Spain, 2008, pp. 164–172.

[8] S. España-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models," *IEEE Trans. on PAMI*, vol. 33, no. 4, pp. 767–779, 2011.

[9] Z. Lu, R. Schwartz, and C. Raphael, "Script-independent, HMM-based text line finding for OCR," in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 551–554.

[10] V. Bosch, A. Toselli, and E. Vidal, "Statistical Text Line Analysis in Handwritten Documents," in *Proc. Int. Conf. Frontiers in Handwriting Recognition*, Bari, Italy, 2012, pp. 201–206.

[11] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," in *Proc. 18th Int. Conf. Machine Learning*, San Francisco, CA, USA, 2001, pp. 282–289.

[12] S. Marinai, M. Gori, and G. Soda, "Artificial Neural Networks for Document Analysis and Recognition," *IEEE Trans. on PAMI*, vol. 27, no. 1, pp. 23–35, 2005.

[13] L. Rabiner and B. H. Huang, *Fundamentals of Speech Recognition*. Prentice-Hall, 1993.

[14] H. Bourlard and N. Morgan, *Connectionist speech recognition—A hybrid approach*, ser. Series in engineering and computer science. Kluwer Academic, 1994, vol. 247.

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

[16] F. Zamora-Martínez *et al.*, "April-ANN toolkit, A Pattern Recognizer In Lua, Artificial Neural Networks module," 2013, https://github.com/pakozm/april-ann.

[17] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *The Journal of Machine Learning Research*, vol. 13, pp. 281–305, 2012.

[18] E. Jones, T. Oliphant, and P. Peterson, "Scipy: Open source scientific tools for python," *http://www. scipy. org/*, 2001.

[19] A. H. Toselli *et al.*, "Integrated Handwriting Recognition and Interpretation using Finite-State Models," *Int. Journal of Pattern Recognition and Artificial Intelligence*, vol. 18, no. 4, pp. 519–539, 2004.

[20] R. Bertolami and H. Bunke, "Hidden markov model-based ensemble methods for offline handwritten text line recognition," *Patt. Recognition*, vol. 41, no. 11, pp. 3452 – 3460, 2008.

[21] P. Dreuw, G. Heigold, and H. Ney, "Confidence and Margin-Based MMI/MPE Discriminative Training for Online Handwriting Recognition," *Int. Journal of Document Analysis and Recognition*, vol. 14, no. 3, pp. 273–288, 2011.

[22] T. Pltz and G. Fink, "Markov models for offline handwriting recognition: a survey," *Int. Journal of Document Analysis and Recognition*, vol. 12, pp. 269–298, 2009.

[23] P. Dreuw, P. Doetsch, C. Plahl, and H. Ney, "Hierarchical Hybrid MLP/HMM or rather MLP Features for a Discriminatively Trained Gaussian HMM: A Comparison for Offline Handwriting Recognition," in *Proc. Int. Conf. on Image Processing*, 2011, pp. 3541–3544.

[24] A. Graves *et al.*, "A Novel Connectionist System for Unconstrained Handwriting Recognition," *IEEE Trans. on PAMI*, vol. 31, no. 5, pp. 855–868, 2009.

[25] F. Z. Martínez *et al.*, "Neural network language models for off-line handwriting, recognition," *Pattern Recognition*, vol. 47, no. 4, pp. 1642–1652, 2014.

[26] A. Stolcke, "SRILM: an extensible language modeling toolkit," in *Proc. Int. Conf. on Spoken Language Processing (ICSLP)*, 2002, pp. 901–904.

[27] S. Johansson, E. Atwell, R. Garside, and G. Leech, "The Tagged LOB Corpus: User's Manual," Norwegian Computing Centre for the Humanities, Bergen, Norway, Tech. Rep., 1986.

[28] W. Francis and H. Kucera, "Brown Corpus Manual, Manual of Information to accompany A Standard Corpus of Present-Day Edited American English," Dep. of Linguistics, Brown University, Providence, Rhode Island, US, Tech. Rep., 1979.

[29] L. Bauer, "Manual of Information to Accompany The Wellington Corpus of Written New Zealand English," Dep. of Linguistics, Victoria University, Wellington, New Zealand, Tech. Rep., 1993.

[30] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. 21st Int. Conf. Pattern Recognition*, Tsukuba, Japan, 2012, pp. 3304–3308.

[31] P. H. O. Pinheiro and R. Collobert, "Recurrent Convolutional Neural Networks for Scene Parsing," *CoRR*, vol. abs/1306.2795, 2013.

[32] J. Peng, L. Bo, and J. Xu, "Conditional Neural Fields," in *Proc. Advances in Neural Information Processing Systems 22*, 2009, pp. 1419–1427.