Improvements in RWTH's system for off-line handwriting recognition

Michał Kozielski, Patrick Doetsch, Hermann Ney
Human Language Technology and Pattern Recognition Group
Chair of Computer Science 6
RWTH Aachen University, D-52056 Aachen, Germany
{kozielski,doetsch,ney}@i6.informatik.rwth-aachen.de

***Abstract*—In this paper we describe a novel HMM-based system for off-line handwriting recognition. We adapt successful techniques from the domains of large vocabulary speech recognition and image object recognition: moment-based image normalization, writer adaptation, discriminative feature extraction and training, and open-vocabulary recognition. We evaluate those methods and examine their cumulative effect on the recognition performance. The final system outperforms current state-of-the-art approaches on two standard evaluation corpora for English and French handwriting.**

I. Introduction

The methods used in today's state-of-the-art large vocabulary handwriting recognition systems can be divided into two groups. The first group comprises algorithms that have their roots in image processing and have been designed to explicitly deal with challenges introduced by handwritten documents. Those challenges include: differences in image contrast, cursive writing resulting in text being slanted, and variations in size and shape of characters. The second group represents general learning methods that have been successfully adopted from other domains, most notably from speech recognition; and are only loosely related to dealing with handwritten documents. Because the word and character segmentation is a nontrivial problem, most of the today's state-of-the-art systems utilize approaches that infer it implicitly; mainly HMM models or recurrent neural networks. The ability to adapt to a text coming from writers not seen in training is an another requirement put on the modern systems. Moreover the systems have to deal with large vocabularies containing hundreds of thousands of words composed of dozens of different characters.

In this paper we present a novel and robust system for off-line handwriting recognition. We introduce the relevant algorithms and measure the successive improvement in recognition performance. Sections II-A and II-B describe a moment-based scheme for preprocessing and feature extraction. We explain the refinements over the method published in [1]. Sections II-C and II-D introduce all maximum-likelihood–based components of the system: model length estimation and writer adaptation. We investigate how these methods work together with the moment-based preprocessing. Furthermore section II-E gives an overlook on discriminative feature extraction using neural networks, and discriminative training of HMM models. We observe how various combinations of discriminative learning work together with the writer adaptation method. Finally section II-F describes the principle behind the open-vocabulary recognition. In section III-B we report results on two standard evaluation corpora for English (IAM) and French (RIMES) handwriting. Section III-C contains a comparison to the best published results. Our systems outperforms other approaches and scores 13.3% and 13.7% word error rate on the evaluation set of the IAM and RIMES corpora respectively.

II. System description

A. Preprocessing

The information about the segmentation of text pages into text lines is provided in the corpora we use. The preprocessing starts with normalizing the contrast of gray-scale images. We use an algorithm that maps 70% of the lightest pixels to white and 5% of the darkest pixels to black. The rest of the pixels is normalized linearly. Then we correct the slant of images with a median of angle values estimated by three different deslanting algorithms [2][3][4]. The algorithms work by shearing the image with an angle from a certain range and evaluating those transformations with different objective functions. After deslanting we extract frames with an overlapping sliding window of width 30 pixels and shift 3 pixels. The height of the sliding windows is equal to the size of the original image. A horizontal cosine window is applied to each frame to smooth the image on its borders.

We compute the 1st- and 2nd-order moments for each frame independently. The 1st-order moments represent the center of gravity which is used to shift the content of the frame to the center of the image. The 2nd-order moments correspond to the weighted standard deviation of the distance between pixels in the frame and the center of gravity. They are used to compute the scaling factors. Figure 1 illustrates

(a) (b) (c)

Fig. 1: Original image (a) and the results of 1st-order (b) and 2nd-order (c) normalization.

1520-5363/13 $26.00 © 2013 IEEE
DOI 10.1109/ICDAR.2013.190

the normalization procedure on a selected frame. The moments are computed not from the gray-scale image itself, but from the gradient of that image, which is approximated by the magnitude of the vertical and horizontal Sobel operators. Each frame is normalized in such a way that the resulting moments of all frames are equal after the normalization. The resulting dimensions of the frame are 8×32 pixels. The aspect ratio is not kept during normalization because the vertical and horizontal moments are computed and normalized separately. For the definition of moments and normalization formulas please refer to [1].

B. Feature extraction

Every frame extracted with the sliding window is transformed into a single feature vector. The gray-scale values of all pixels in a frame are used as features and are further reduced by PCA to 20 components. The number of principal components of the PCA transformation has a small influence on the recognition performance. Note that the moment-based normalization procedure has a serious effect on the inter-class distances. Objects of different size but similar shape are closer to each other. This effect is magnified by the distortion caused by the changed aspect-ratio. In order to overcome this inter-class similarity we augment the feature vector by adding the original moments:

$$\left[\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} - p_y, \sqrt{\frac{\mu_{20}}{m_{00}}}, \sqrt{\frac{\mu_{02}}{m_{00}}}\right] \tag{1}$$

where p_y is the value m_{01}/m_{00} computed over an entire text line and gives a rough estimate of the vertical baseline. A negative value of the second dimension of (1) suggests an ascender, a positive one a descender. m_{pq} is a geometric moment of order p+q computed over pixels' gray-scale values. μ_{pq} is a geometric moment shifted by the center of gravity and is called a central moment. In this way we map the class-specific moment information, which was originally distributed over the whole image, to specific components of the feature vector. The final feature vector has 24 dimensions. For the purpose of training and recognition one segment for the decoder constitutes of a sequence of concatenated feature vectors that have been extracted from a single page.

C. HMM structure

Every character is modelled by a Viterbi-trained left-to-right HMM with loop and skip transitions. Those transitions (treaded as penalties in negative log scale) are constant and fixed across all models. The loop and skip penalties are equal to 3, and the forward transition is not penalized. There is an additional penalty in the decoder for exiting the HMM model, which controls the number of insertions during recognition. The HMM structure consist of segments, each segment consists of 2 states sharing the same emission distribution. This topology ensures that no emission distribution is omitted during training. The number of segments per character varies and is computed as follows from initial training alignment:

$$S_c = f_P \frac{N_{x,c}}{N_c} \tag{2}$$

where S_c is the estimated number of segments for character c, $N_{x,c}$ is the number of frames aligned to c, N_c is the number of occurrences of c in training, and f_P is a scaling factor. The technique of adjusting the number of HMM states per character is called model length estimation (MLE) [5]. A comparison of different methods for optimizing the number of states can be found in [6]. The initial alignments are created during the training of the system using a constant number of segments equal to 6. We model the emission distributions of our HMM-based system using Gaussian mixtures with 64 densities and a globally-pooled diagonal covariance matrix.

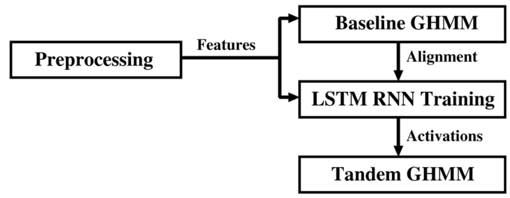

Fig. 2: Illustration of the LSTM-RNN tandem HMM system. A baseline HMM is applied in a forced alignment mode in order to generate frame-wise labels. The LSTM-RNN is trained with this labeling and the activations of a hidden layer are used as features to train a new HMM system afterwards.

D. Writer adaptation

A very common approach to writer adaptation for Gaussian mixture HMM models is the maximum likelihood linear regression (MLLR). The method works by normalizing means and eventually covariances of the mixture components to compensate for writer variations. In the constrained case (CMLLR) we force the mean transformations to be equal to the covariance transformations. This method has been described in detail in [7]. It is desirable to estimate transformation matrices for all emission distributions, but because of lack of training data the matrices are usually grouped into so called regression classes [8]. Here however we pool over all HMM states which results in one global transformation matrix for each writer. Moreover we apply the transformation in the feature-space instead of the parameter space, so no changes in the decoder implementation are needed. The frame-to-state alignments required for training of the matrices are obtained from a previous training iteration. We compute one transformation matrix for every writer in both training and recognition.

The recognition is performed in two passes. In the first pass we obtain transcriptions for the test set using a system that has been trained on non-transformed features. We use those transcriptions to create frame-to-state alignments and to compute transformation matrices. Because the writer information is not available in the test set, we assume that every text page comes from a separate writer. In the second pass we perform the final recognition using the transformed features.

E. Discriminative training

Discriminative HMM training and discriminative feature extraction methods have shown to improve system performance of HMM-based recognition systems [9]. In our system we perform a discriminative feature extraction using a neural-network–based tandem approach [10]. In particular we use the

TABLE I: Evaluation of the preprocessing algorithms on the IAM development set.

Method	WER [%]	CER [%]
No preprocessing	34.6	22.8
Contrast, deslanting	21.3	10.9
Moments	19.9	10.6
Contrast, deslanting, moments	14.6	5.5

TABLE II: Evaluation of the modeling techniques on the IAM development set.

Method	WER [%]	CER [%]
Standard	14.6	5.5
MLE	14.1	5.0
MLE, CMLLR	13.1	4.4

TABLE III: Evaluation of the discriminative training procedures on the IAM development set.

Method	WER [%]	CER [%]
Maximum likelihood	13.1	4.4
M-MPE	12.2	4.7
LSTM-RNN	12.2	3.6
LSTM-RNN, M-MPE	11.9	3.2

TABLE IV: Evaluation of the open-vocabulary recognition on the IAM development set.

Method	WER [%]	CER [%]
Maximum likelihood, OOV	10.7	3.8
LSTM-RNN, OOV	10.1	3.4
LSTM-RNN, M-MPE, OOV	9.5	2.7

Long-Short-Term-Memory recurrent neural network (LSTM-RNN) [11] which was successfully applied to several sequence learning problems including handwriting recognition [12]. The LSTM-RNN is trained on the frame level. The labeling of the frames is obtained by applying the HMM-based system described in the previous sections to the training data in a forced alignment mode. The overall training procedure is depicted in Figure 2.

In our experiments we trained a bidirectional LSTM-RNN with two hidden layers containing 100 and 200 hidden nodes respectively resulting in roughly 850k weights. The network was trained with Backpropagation-Through-Time [13]. Convergence was detected using a validation set consisting of roughly 20% of the training data. After convergence, the network was applied to all frames and the activation pattern of the first hidden layer was extracted. Through the bidirectional network topology a 200-dimensional feature vector was produced in this way, which was further normalized and reduced by PCA to 20 components. The resulting neural-network–based features were used to train a new HMM system using the same HMM topology as described in the previous section.

Although the extracted features already contain discriminative information about the classes it has been shown in [9] that discriminative HMM training with those features can improve recognition performance. In our experiments we used the modified minimum phone error criterion (M-MPE) [14][15]. Convergence of the iterative training procedure was detected by evaluating the system on the development set of the IAM database. In our experiments the training was very robust and the objective function reached its minimum after 50 iterations.

F. Language modeling

The recognition of Out-of-vocabulary words requires going beyond the standard word-level language models and being able to transcribe sub-word units, most desirably single characters. There have been numerous approaches to this problem with character-level and mixed language models [16][17][18]. In our system we use a combination of two language models [19]. Additionally to a standard word-level language model we use a separate n-gram character-level language model for out-of-vocabulary word detection and recognition. The probabilities assigned by those two models are combined into one Bayes decision rule. This approach clearly separates the word representation from character representation. The contexts of those two language models are separated so words and characters do not interfere with each other. The language models can also have different orders and can be created using different discounting methods. Moreover a very important word lexical constraint is retained as opposed to pure character-level language models.

III. Experiments

A. Databases and language models

The IAM database [20] consist of images of handwritten English text sentences. There are 747 paragraphs of text for training, 116 for development, and 336 for evaluation. A writer appearing in one set does not appear in any of the other sets. There are 283 different writers in the training set. The language models have been built upon the combined LOB [21], Brown [22], and Wellington [23] corpora. We have excluded the sentences appearing in the IAM development and evaluation sets for the purpose of training the language models. As word-level language model (LM) we use a standard 3-gram model with modified Kneser-Ney discounting built upon the training text source containing one sentence per line. The vocabulary consists of 50k most frequent words from the training set, which leads to a 4% OOV rate on the development set. The perplexity of the LM is 420. Our 10-gram character-level LM has been built upon a list of OOV words extracted from the training set. The character inventory contains 77 characters plus silence and noise. Because we recognize whole paragraphs of text, which contain multiple sentences, the LM has to be able to hypothesize the sentence boundary.

The RIMES Database [24] comes from the ICDAR 2011 block-recognition competition and consists of images of handwritten French text sentences. There are 1500 paragraphs for training and 100 for evaluation. We have built the word-level 4-gram LM upon the annotations of the training set. The perplexity of the LM is 26. There are 672 different writers in the training set. We have not applied the character-level LM because of insufficient amount of training data. The character inventory contains 96 characters plus silence.

TABLE V: Evaluation of the modeling techniques on RIMES.

Method	WER [%]	CER [%]
Standard	16.5	5.9
MLE	15.9	5.6
MLE, CMLLR	15.7	5.5

TABLE VI: Evaluation of the discriminative training procedures on RIMES.

Method	WER [%]	CER [%]
Maximum likelihood	15.7	5.5
LSTM-RNN	13.9	4.8
LSTM-RNN, M-MPE	13.7	4.6

B. Experimental results

Error rates are calculated using the Levenshtein distance between references and hypotheses. We evaluate the methods summarized in this paper on the development set of the IAM database. As evaluation metrics for our experiments we use the word error rate (WER) and character error rate (CER).

Table I summarizes the effect of the preprocessing methods described in Section II-A. Contrast normalization and deslanting are essential preprocessing steps improving the baseline system by 10.6% WER absolutely. Alternatively a system with only moment-based size normalization achieves an absolute improvement of 12% WER. Applying all the preprocessing steps together further improves the system by 5.3% WER. It is important to note that robust preprocessing is crucial for obtaining state-of-the-art results. A detailed comparison can be found in section III-C.

Table II summarizes the effect of modeling techniques explained in sections II-C and II-D. Estimating a different number of HMM states per character (MLE) reduces the WER further from 14.6% to 14.1%. The writer adaptation with CMLLR brings another reduction of the WER to 13.1%. This Maximum-likelihood trained system serves as a baseline for the discriminative feature extraction and training methods described in Section II-E.

The LSTM-RNN tandem system evaluates to a WER of 12.2% which could further be reduced to 11.9% WER with the discriminative HMM training using the M-MPE criterion. Applying the M-MPE training to the writer-adapted features evaluates to an WER 12.2%. The difference in WER between the LSTM-RNN tandem system and the system without LSTM-RNN training is only 0.3%. However, it is worth mentioning that the CER is reduced by 0.8% through the LSTM-RNN training. The discriminative training procedures are compared in Table III.

Table IV summarizes the results of the open-vocabulary recognition with different systems evaluated before. In case of our discriminative-trained system we obtain an improvement from 11.9% to 9.5%. We manage to recognize 32% of the OOV words correctly. The recognition accuracy of the in-vocabulary words also improves. The statistics are calculated using the Levenshtein alignment. On the evaluation set our final system reaches the word error rate of 13.3%.

TABLE VII: Comparison with results reported by other groups on the IAM development and evaluation sets.

Systems	Voc.	WER [%]		CER [%]	
		Dev.	Eval	Dev.	Eval
Our system	50k	9.5	13.3	2.7	5.1
España et al. [25]	50k	19.0	22.4	-	9.8
Toselli et al. [26]	9k	-	25.8	-	-
Graves et al. [12]	20k	-	25.9	-	18.2
Bertolami et al. [27]	20k	26.8	32.8	-	-

TABLE VIII: Comparison with results reported by other groups on RIMES evaluation set from the ICDAR 2011 competition.

Systems	WER [%]	CER [%]
Our system	13.7	4.6
A2IA [28]	15.2	7.2
Telecom ParisTech [29]	31.2	-

We trained the same system on the RIMES database. The advancements coming from the preprocessing and the model length estimation procedure were comparable to those experienced on the IAM database. However the improvement coming from writer adaptation was negligible, as seen on table V. The discriminative training procedures showed a similar performance to the one demonstrated on the IAM database, as seen on table VI. The final result of our system on the RIMES database is 13.7% word error rate.

C. Comparison with the state-of-the-art

Table VII shows the comparison of the results on the IAM database. We achieve a word error rate of 13.3% on the evaluation set, which is the best result published so far. In [25] neural networks were used to perform particular preprocessing steps and finally another neural network was used to estimate the state posterior probabilities. Toselli [26] used an HMM model with features composed of gray values and gradients. In [12] the authors used an LSTM recurrent neural network with a CTC output layer. In [27] a voting strategy was applied to a set of HMM models. Table VIII shows the comparison of the results on the RIMES database, where Menasri [28] used a combination of an HMM model and recurrent neural networks. We achieve a word error rate of 13.7% on the evaluation set, which is the best result published so far. Missing numbers in tables VII and VIII were not included in the referenced papers.

IV. Conclusions

We have shown that certain preprocessing, adaptation, and discriminative learning methods presented in this paper bring a cumulative improvement of the recognition results. Those methods can be successfully combined and used to build a state-of-the-art HMM-based system. On the IAM and RIMES evaluation sets our system outperforms other approaches and scores 13.3% and 13.7% word error rate respectively.

Acknowledgments. This work was partially supported by a Google Research Award and by the Quaero Programme, funded by OSEO, French State agency for innovation.

References

[1] M. Kozielski, J. Forster, and H. Ney, "Moment-based image normalization for handwritten text recognition," in *International Conference on Frontiers in Handwriting Recognition*, Bari, Italy, Sep. 2012, pp. 256–261.

[2] M. Pastor, A. Toselli, and E. Vidal, "Projection profile based algorithm for slant removal," in *Image Analysis and Recognition*, ser. Lecture Notes in Computer Science, A. Campilho and M. Kamel, Eds. Springer Berlin / Heidelberg, 2004, vol. 3212, pp. 183–190.

[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] M. P. i Gadea, A. H. Toselli, V. Romero, and E. Vidal, "Improving handwritten off-line text slant correction," in *Procc. of The Sixth IASTED international Conference on Visualization, Imaging, and Image Processing (VIIP 06)*, 2006.

[5] P. Dreuw, D. Rybach, C. Gollan, and H. Ney, "Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition," in *International Conference on Document Analysis and Recognition*, Barcelona, Spain, Jul. 2009, pp. 21–25.

[6] M. Zimmermann and H. Bunke, "Hidden markov model length optimization for handwriting recognition systems," in *Frontiers in Handwriting Recognition, 2002. Proceedings. Eighth International Workshop on*, 2002, pp. 369–374.

[7] M. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[8] C. Leggetter and P. Woodland, "Flexible speaker adaptation using maximum likelihood linear regression," in *Proc. ARPA Spoken Language Technology Workshop*. Morgan Kaufmann, 1995, pp. 110–115.

[9] 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 *IEEE International Conference on Image Processing*, Brussels, Belgium, Sep. 2011.

[10] H. Hermansky, D. P. W. Ellis, and S. Sharma, "Tandem connectionist feature extraction for conventional HMM systems," in *Acoustics, Speech, and Signal Processing (ICASSP), 2000 IEEE International Conference on*.

[11] M. Liwicki, A. Graves, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proceedings of the 9th International Conference on Document Analysis and Recognition (ICDAR)*, 2007.

[12] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 855–868, May 2009.

[13] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct. 1990.

[14] D. Povey and P. C. Woodland, "Minimum phone error and i-smoothing for improved discriminative training," in *Acoustics, Speech, and Signal Processing (ICASSP), 2002 IEEE International Conference on*.

[15] P. Dreuw, G. Heigold, and H. Ney, "Confidence- and margin-based MMI/MPE discriminative training for off-line handwriting recognition," *Int. J. Doc. Anal. Recognit.*, vol. 14, no. 3, pp. 273–288, Sep. 2011.

[16] A. Rastrow, A. Sethy, B. Ramabhadran, and F. Jelinek, "Towards using hybrid word and fragment units for vocabulary independent LVCSR systems," in *INTERSPEECH 2009, 10th Annual Conference of the International Speech Communication Association*, Brighton, United Kingdom, Sep. 2009, pp. 1931–1934.

[17] K. Vertanen, "Combining open vocabulary recognition and word confusion networks," in *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, Apr. 2008, pp. 4325–4328.

[18] I. Bazzi, R. Schwartz, and J. Makhoul, "An omnifont open-vocabulary OCR system for English and Arabic," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 21, no. 6, pp. 495–504, Jun. 1999.

[19] M. Kozielski, D. Rybach, S. Hahn, R. Schlüter, and H. Ney, "Open vocabulary handwriting recognition using combined word-level and character-level language models," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, May 2013, in press.

[20] U.-V. Marti and H. Bunke, "The IAM-database: an English sentence database for offline handwriting recognition," *International Journal on Document Analysis and Recognition*, vol. 5, pp. 39–46, Nov. 2002.

[21] S. Johansson, E. Atwell, R. Garside, and G. Leech, *The Tagged LOB Corpus: User's Manual*, Norwegian Computing Centre for the Humanities, 1986.

[22] W. Francis and H. Kucera, "Brown corpus manual, manual of information to accompany a standard corpus of present-day edited American English," Tech. Rep., 1979.

[23] L. Bauer, "Manual of information to accompany the Wellington corpus of written New Zealand English," Tech. Rep., 1993.

[24] E. Augustin, M. Carré, G. E., J. M. Brodin, E. Geoffrois, and F. Preteux, "Rimes evaluation campaign for handwritten mail processing," in *Proceedings of the Workshop on Frontiers in Handwriting Recognition*, 2006, pp. 231–235.

[25] S. España-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez, "Improving offline handwritten text recognition with hybrid HMM/ANN models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 4, pp. 767–779, Apr. 2011.

[26] A. H. Toselli, V. Romero, M. P. i Gadea, and E. Vidal, "Multimodal interactive transcription of text images," *Pattern Recognition*, vol. 43, no. 5, pp. 1814–1825, 2010.

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

[28] F. Menasri, J. Louradour, A.-L. Bianne-Bernard, and C. Kermorvant, "The A2iA French handwriting recognition system at the Rimes-ICDAR2011 competition." in *Document Recognition and Retrieval Conference*, ser. SPIE Proceedings, C. Viard-Gaudin and R. Zanibbi, Eds., vol. 8297. SPIE, 2012.

[29] E. Grosicki and H. El Abed, "ICDAR 2009 handwriting recognition competition," in *Document Analysis and Recognition, 2009. ICDAR '09. 10th International Conference on*, Jul. 2009, pp. 1398–1402.