RWTH OCR: A Large Vocabulary Optical Character Recognition System for Arabic Scripts

Philippe Dreuw, David Rybach, Georg Heigold, and Hermann Ney

Abstract We present a novel large vocabulary OCR system, which implements a confidence- and margin-based discriminative training approach for model adaptation of an HMM based recognition system to handle multiple fonts, different handwriting styles, and their variations. Most current HMM approaches are HTK based systems which are maximum-likelihood (ML) trained and which try to adapt their models to different writing styles using writer adaptive training, unsupervised clustering, or additional writer specific data. Here, discriminative training based on the Maximum Mutual Information (MMI) and Minimum Phone Error (MPE) criteria are used instead. For model adaptation during decoding, an unsupervised confidence-based discriminative training within a two-pass decoding process is proposed. Additionally, we use neural network based features extracted by a hierarchical multi-layer-perceptron (MLP) network either in a hybrid MLP/HMM approach or to discriminatively retrain a Gaussian HMM system in a tandem approach. The proposed framework and methods are evaluated for closed-vocabulary isolated handwritten word recognition on the IfN/ENIT Arabic handwriting database, where the word-error-rate is decreased by more than 50% relative compared to a ML trained baseline system. Preliminary results for large-vocabulary Arabic machine printed text recognition tasks are presented on a novel publicly available newspaper database.

RWTH Aachen University
Human Language Technology and Pattern Recognition
Ahornstr 55, D-52056 Aachen, Germany
Tel.: +49-241-80-21613
Fax: +49-241-80-22219
e-mail: <surname>@cs.rwth-aachen.de

1 Introduction

In this work, we describe our novel large vocabulary optical character recognition (OCR) system. Our hidden Markov model (HMM) based RWTH OCR system is based on a publicly available state-of-the-art large vocabulary continuous speech recognition (LVCSR) framework which has been designed for the special requirements of research applications and supports for grid-computing.

The aim of this work is to analyze for Arabic handwriting and machine printed text recognition tasks the effect of discriminative MMI/MPE training and the incorporation of a margin and a confidence term into discriminative criteria. Therefore none of the preprocessing steps commonly applied in handwriting recognition like binarization, deskewing, deslanting or size-normalization are used.

The focus of this work shall be on offline handwriting recognition of closed-vocabulary isolated Arabic words and large open-vocabulary machine-printed Arabic text recognition tasks in combination with n-gram language models. More explicitly, the novelties of our investigation are as follows:

1. Conversion of a state-of-the-art large vocabulary speech recognition framework for handwritten and machine-printed OCR.
2. Analysis of offline handwritten and machine-printed Arabic text recognition.
3. Direct evaluation of the utility of the margin term in MMI/MPE based training. Ideally, we can turn on/off the margin term in the optimization problem.
4. Direct evaluation of the utility of an additional confidence term. Ideally, we improve over the best trained system by retraining the system with unsupervised labeled test data.
5. Evaluation on state-of-the-art systems. Ideally, we directly improve over the best discriminative system, e.g. conventional (i.e., without margin) MMI/MPE for handwriting recognition.
6. Evaluation of hybrid MLP/HMM and discriminatively retrained MLP-GHMM tandem approaches.

The remainder of this chapter is structured as follows: First, the background in described in Section 2. Next, Section 3 gives a system overview, whereas the RWTH OCR software framework is presented in Section 4. The datasets we used for evaluating the proposed framework are explained in Section 5 where especially our ongoing work in creating a publicly available database for Arabic machine-printed text recognition is presented in Section 5.2. Experimental results are presented in Section 6, and finally, the chapter is concluded in Section 7.

2 Background

From a system point of view, many approaches for Arabic handwriting recognition [3] in the past were HMM based systems using the Hidden Markov Model Toolkit (HTK) [78]. BBN's Glyph HMM system "Byblos" [46, 51, 67], has been extended to "PLATO" [52] within the MADCAT [57] project, and is used for handwriting and machine-printed OCR tasks. SIEMENS [70] showed how to convert a Latin OCR system to Arabic handwritings. Other projects like OCRopus[1] or Tesseract[2] currently do not support the recognition of Arabic scripts, and apparently only a few commercial applications like Readiris[3] and NovoDynamics VERUS[4] can support those cursive scripts.

Many commercial machine-printed OCR products or systems described in the published literature developed their recognition algorithms on isolated characters [45]. These systems usually assumed that characters can be segmented accurately as a first step, and made hard decisions at each stage which resulted in an accumulation of errors, thus broken and touching characters were responsible for the majority of errors. Obviously these assumptions are too strong for degraded or handwritten documents, or font-free approaches [37].

Such approaches were surpassed by late-decision systems, e.g. tools developed by the speech recognition community, such as hidden Markov models (HMMs). In these systems, multiple hypotheses about both segmentations and identities are maintained, and the final decisions are made at the end of an observation sequence by tracing back the local decisions which led to the best global hypothesis [34]. Similar to the framework presented in [67, 51] our novel RWTH OCR system is able to recognize Arabic handwritten *and* machine printed text.

State-of-the-art speech recognition systems are based on discriminative Gaussian HMMs (GHMMs), where major points of criticism of this conventional approach are the indirect parameterization of the posterior model, the nonconvexity of the conventional training criteria, and the insufficient flexibility of the HMMs to incorporate additional dependencies and knowledge sources [28]. State-of-the-art handwritten text recognition systems are usually based on HMMs too [7, 22, 70], but are typically trained using the maximum-likelihood (ML) criterion. Hybrid neural network based systems like RNN / CTC [27], MLPs / HMM [21] or tandem based approaches like MLP-GHMM [71] were recently very successful in online and offline handwriting recognition. However, most of the tandem based approaches use an ML based training criterion to retrain the GHMMs.

Typical training criteria for string recognition like for example minimum phone error (MPE) and maximum mutual information (MMI) in speech recognition are based on a (regularized) loss function. In contrast, large margin classifiers - the defacto standard in machine learning - maximize the separation margin. An additional loss term penalizes misclassified samples.

The MMI training criterion has been used in [55] to improve the performance of an HMM based offline Thai handwriting recognition system for isolated characters. They propose a feature extraction based on a block-based PCA and composite image features, which are reported to be better at discriminating Thai confusable

[1] http://code.google.com/p/ocropus/

[2] http://code.google.com/p/tesseract-ocr/

[3] http://www.irislink.com/readiris/

[4] http://www.novodynamics.com/

characters. In [8], the authors apply the Minimum Classification Error (MCE) criterion to the problem of recognizing online unconstrained-style characters and words, and report large improvements on a writer-independent character recognition task when compared to an ML trained baseline system.

Similar to systems presented in [54, 12, 52], we apply the MMI/MPE criterion, but modified by a margin term. This margin term can be interpreted as an additional observation-dependent prior weakening the true prior [35], and is identical with the support vector machine (SVM) optimization problem of log-linear models [30].

The most common way for unsupervised adaptation is the use of the automatic transcription of a previous recognition pass without the application of confidence scores. Many publications in automatic speech recognition (ASR) have shown that the application of confidence scores for adaptation can improve recognition results. However, only small improvements are reported for maximum likelihood linear regression (MLLR) adaptation [25, 60, 62] or confidence-based constrained MLLR (CMLLR) adaptation [5]. In addition to the margin concept, the MMI/MPE training criteria are extended in this work by an additional confidence term [15] to allow for novel unsupervised model adaptation.

3 System Overview

In offline handwriting recognition, we are searching for an unknown word sequence $w_1^N := w_1, \dots, w_N$, for which the sequence of features $x_1^T := x_1, \dots, x_T$ fits best to the trained models. We maximize the posterior probability $p(w_1^N|x_1^T)$ over all possible word sequences w_1^N with unknown number of words N. This is modeled by Bayes' decision rule:

$$x_1^T \to \hat{w}_1^N(x_1^T) = \arg\max_{w_1^N} \left\{ p^{\kappa}(w_1^N) p(x_1^T|w_1^N) \right\} \tag{1}$$

with κ being a scaling exponent of the language model.

Especially in Arabic handwriting with its position dependent glyphs [44], large white-spaces can occur between isolated-, beginning-, and end-shaped characters (see Figure 1 (a)). As a specific set of characters is only connectable from the right side, such words have to be cut into parts (Part of Arabic Word (PAW)). Due to ligatures and diacritics in Arabic handwriting, the same Arabic word can be written in several writing variants, depending on the writer's handwriting style.

In this work, we use a writing variant model refinement [19] of our visual model

$$\begin{aligned} p(x_1^T|w_1^N) = \\ \max_{v_1^N|w_1^N} \left\{ p^{\alpha}_{\Lambda_\mathrm{v}}(v_1^N|w_1^N) p^{\beta}_{\Lambda_\mathrm{e,t}}(x_1^T|v_1^N, w_1^N) \right\} \end{aligned} \tag{2}$$

with v_1^N a sequence of unknown writing variants, α a scaling exponent of the writing variant probability depending on a parameter set Λ_v, and β a scaling exponent of the visual model depending on a parameter set $\Lambda_\mathrm{e,t}$ for emission and transition model.

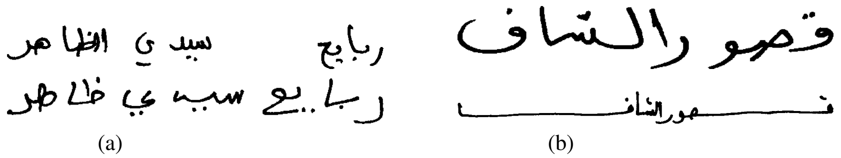

Fig. 1 Two examples where each column shows the same Tunisian town name: large white-spaces (a) and a large stretching of long drawn-out characters (b) occurs often in Arabic handwriting. Therefore an adequate modeling of white-spaces and state-transition penalties are important parts to be tuned in an HMM based Arabic handwriting recognition system.

During training, a corpus and lexicon with supervised writing variants instead of the commonly used unsupervised writing variants can be used, during decoding, the writing variants can only be used in an unsupervised manner. Obviously, the supervised writing variants in training can lead to better trained glyph models only if the training corpora have a high annotation quality. Usually, the probability $p(v|w)$ for a variant v of a word w is considered as uniformly distributed [17]. Here we use the count statistics as probability

$$p(v|w) = \frac{\mathsf{N}(v,w)}{\mathsf{N}(w)} \tag{3}$$

where the writing variant counts $\mathsf{N}(v,w)$ and the word counts $\mathsf{N}(w)$ are estimated from the corresponding training corpora, and represent how often these events were observed. Note that $\sum_{v'} \frac{\mathsf{N}(v',w)}{\mathsf{N}(w)} = 1$. The scaling exponent α of the writing variant probability of Equation 2 can be adapted in the same way as it is done for the language model scale κ in Equation 1.

3.1 Feature Extraction

The images are scaled down to a fixed height while keeping their aspect ratio. We extract simple appearance-based image slice features x'_t at every time step $t = 1,\ldots,T$ which are augmented by their spatial derivatives in horizontal direction $\Delta = x'_t - x'_{t-1}$. Note that many systems divide the sliding window itself into several sub-windows and extract different features within each of the sub-windows [6,36,55,70]

In order to incorporate temporal and spatial context into the features, we concatenate 7 consecutive features in a sliding window with maximum overlap, which are later reduced by a PCA transformation matrix to a feature vector x_t of dimension 30 (see Figure 2).

Without any preprocessing of the input images, the simple appearance-based image slice features $x_t = [x'_t, \Delta]$ together with their corresponding state alignments can then be processed by a hierarchical MLP framework originally described in [76].

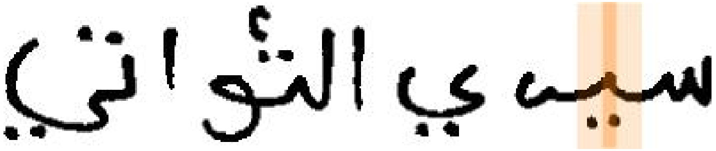

Fig. 2 Right-to-left sliding PCA window over input images without any preprocessing for Arabic handwriting.

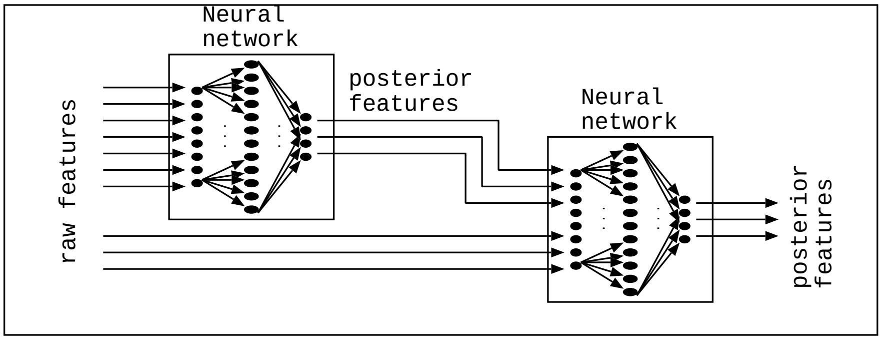

Fig. 3 Hierachical MLP network for discriminative feature extraction in OCR

Depending on the MLP hierarchy, preprocessing, and postprocessing operations, several feature sets can be generated. In order to incorporate temporal and spatial context into the features, we concatenate consecutive features in a sliding window, where the MLP outputs are later reduced by a PCA or LDA transformation (cf. Figure 3). Two different MLPs are trained, RAW and TRAP-DCT networks, where network details are given in Section 6.

Instead of using log-PCA/LDA reduced MLP posterior features for retraining a Gaussian HMM system, log-posterior features can be directly used without any reduction in a hybrid MLP/HMM framework [9], as briefly described in Section 3.2.

3.2 Visual Modeling

Arabic handwriting. Depending on the position in an Arabic word, most of the 28 characters can have up to 4 different shapes [44]. Here we use position dependent glyph models to model the different presentation forms, and due to ligatures, a total of 120 glyph models and one white-space model have to be estimated for the IfN/ENIT tasks (see Section 6). Additionally, a large stretching of long drawn-out glyphs occurs often in Arabic handwriting (see Figure 1 (b)). Therefore, we use very low loop penalties but higher skip penalties for our HMM state transitions (see Figure 4 (a)).

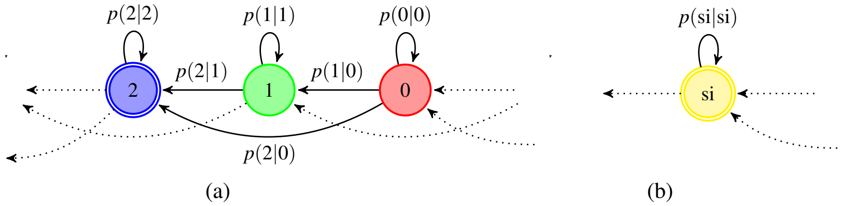

Fig. 4 Different HMM topologies and transition probabilities are used for character models (a) and white-space models (b) in Arabic and Latin handwriting recognition.

Arabic machine-printed text. As for Arabic handwriting, there are no distinct upper and lower case letter forms in machine-printed texts. Both printed and written Arabic are cursive. Unlike cursive writing based on the Latin alphabet, the standard Arabic style has substantially different shapes depending on the glyph context. Standard Arabic Unicode character encodings do typically not indicate the form each character should take in context, so it is left to the rendering engine to select the proper glyph to display for each character.

The basic Arabic range encodes mainly the standard letters and diacritics. For our novel large vocabulary Arabic machine-printed text database described in Section 5.2, about 200 position dependent glyph models have to be trained.

GHMM. Our hidden Markov model (HMM) based OCR system is Viterbi trained using the maximum-likelihood (ML) training criterion and a lexicon with multiple writing variants as proposed in [17, 19].

Each glyph is modeled by a multi-state left-to-right HMM with skip transitions and a separate Gaussian mixture models (GHMMs) with globally pooled variances. The parameters of all Gaussian mixture models (GMMs) are estimated with the ML principle using an expectation maximization (EM) algorithm, and to increase the number of densities in the mixture densities, successive splitting of the mixture densities is applied. Different HMM topologies and transition probabilities are used for glyph models (cf. Figure 4(a)) and white-space models (cf. Figure 4(b)) in Arabic text recognition, where the white-space model itself is always modeled by a single GMM in all systems.

The ML trained GMMs are refined using a discriminative training approach based on the margin-based M-MMI/M-MPE criteria [32] as briefly presented in Section 3.3.

Hybrid MLP/HMM. The MLP posterior probabilities $p(s_t|x_t)$ are divided by the prior state probabilities $p(s_t)$ in order to approximate the observation probabilities of an HMM, i.e. $p(x_t|s_t) \approx \frac{p(s_t|x_t)}{p(s_t)}$ as described in [9].

MLP-GHMM. The MLP-GHMM system is trained from scratch using the MLP log-posterior features as described in Section 3.1 (also known as tandem approach [71]). Again, ML/M-MMI/M-MPE training criteria can be used for GMM training. Note that the MLP network itself can also be trained using different alignments generated by the correspondingly trained GHMM systems.

3.3 Discriminative Training: Incorporation of the Margin and Confidence Term

In this work, we use a discriminative training approach based on the Maximum Mutual Information (MMI) and Minimum Phone Error (MPE) criteria as presented in [30, 31, 29]. In addition to the novel confidence-based extension of the margin-based MMI training presented in [15], the confidence concept has been incorporated in the margin-based MPE criterion in this work.

The proposed approach takes advantage of the generalization bounds of large margin classifiers while keeping the efficient framework for conventional discriminative training. This allows us to directly evaluate the utility of the margin term for OCR. So, our approach combines the advantages of conventional training criteria and of large margin classifiers.

This section briefly reviews how the MMI/MPE training criteria can be extended to incorporate the margin concept, and that such modified training criteria are smooth approximations to support vector machines with the respective loss function [30].

In OCR, the two-dimensional representation of an image is turned into a string representation $X = x_1, \ldots, x_T$ where x_t is a fixed-length array assigned to each column in the image (see Section 3.1 for further details). The word sequence $W = w_1, \ldots, w_N$ is represented by a character string.

Assume the joint probability $p_\Lambda(X,W)$ of the features X and the symbol string W. The model parameters are indicated by Λ. The training set consists of $r = 1, \ldots, R$ labeled sentences, $(X_r, W_r)_{r=1,\ldots,R}$. According to Bayes rule, the joint probability $p_\Lambda(X,W)$ induces the posterior

$$p_{\Lambda,\gamma}(W|X) = \frac{p_\Lambda(X,W)^\gamma}{\sum\limits_V p_\Lambda(X,V)^\gamma}. \tag{4}$$

The likelihoods are scaled with some factor $\gamma > 0$, which is a common trick in speech recognition to scale them to the "real" posteriors [31]. The approximation level γ is an additional parameter to control the smoothness of the criterion.

Let $p_\Lambda(X,W)$ be the joint probability and L a loss function for each training sample r:

$$L[p_\Lambda(X_r,\cdot), W_r] \tag{5}$$

with $\cdot$ representing all possible hypotheses W for a given lexicon, and W_r representing the correct transcription of X_r.

The general optimization problem can be formulated as a minimization of the total loss function:

$$\hat{\Lambda} = \arg\min_{\Lambda} \left\{ C||\Lambda - \Lambda_0||_2^2 + \sum_{r=1}^{R} L[p_\Lambda(X_r,\cdot), W_r] \right\} \tag{6}$$

and includes an ℓ_2 regularization term $||\Lambda - \Lambda_0||_2^2$ (i.e. a prior over the model parameters), where the constant C is used to balance the regularization term and the loss term including the log-posteriors. Here, the ℓ_2 regularization term is replaced by I-smoothing [66], which is a useful technique to make MMI/MPE training converge without over-training, and where the parameter prior is centered for initialization at a reasonable ML trained model Λ_0 (see Section 3.2).

3.3.1 Maximum Mutual Information

In automatic speech recognition (ASR), maximum mutual information (MMI) commonly refers to the maximum likelihood (ML) for the class posteriors. For MMI, the loss function to be minimized is described by:

$$L^{(\text{MMI})}[p_\Lambda(X_r,\cdot),W_r] = -\log \frac{p_\Lambda(X_r,W_r)^\gamma}{\sum\limits_V p_\Lambda(X_r,V)^\gamma}. \tag{7}$$

This criterion has proven to perform reasonably as long as the error rate on the training data is not too low, i.e., generalization is not an issue.

3.3.2 Margin-Based Maximum Mutual Information

Conventional MMI is based on the true posteriors in Equation 4. The margin-based MMI (M-MMI) loss function to be minimized is described by:

$$L_\rho^{(\text{M-MMI})}[p_\Lambda(X_r,\cdot),W_r] = -\log \frac{[p_\Lambda(X_r,W_r)\exp(-\rho A(W_r,W_r))]^\gamma}{\sum\limits_V [p_\Lambda(X_r,V)\exp(-\rho A(V,W_r))]^\gamma}, \tag{8}$$

which has an additional margin-term including the word accuracy $A(\cdot,W_r)$ based on the approximate word error [66]. Note that the additional term can be interpreted as if we had introduced a new posterior distribution. In a simplified view, we interpret this as a pseudo-posterior probability which is modified by a margin term.

Compared with the true-posterior in Equation (4), the M-MMI loss function includes the margin term $\exp(-\rho A(V,W_r))$, which is based on the string accuracy $A(V,W_r)$ between the two strings V,W_r. The accuracy counts the number of matching symbols of V,W_r and will be approximated for efficiency reasons (see Section 3.3.5).

As explained in [31], the accuracy is generally scaled with some $\rho > 0$, and this term weighs up the likelihoods of the competing hypotheses compared with the correct hypothesis [65]. On the contrary, this term can be equally interpreted as a margin term.

3.3.3 Minimum Phone Error

The Minimum Phone Error (MPE) criterion is defined as the (regularized) posterior risk based on the error function $E(V,W)$ like for example the approximate phone error [64], which is probably the training criterion of choice in Large Vocabulary Continuous Speech Recognition (LVCSR). For MPE, the loss function to be minimized is described by:

$$L^{(\text{MPE})}[p_\Lambda(X_r,\cdot),W_r] = \sum_{W\in\cdot} E(W,W_r)\frac{p_\Lambda(X_r,W_r)^\gamma}{\sum\limits_V p_\Lambda(X_r,V)^\gamma}, \tag{9}$$

In OCR, a phoneme unit usually corresponds to a glyph if words are modeled by glyph sequences.

3.3.4 Margin-Based Minimum Phone Error

Analogously, the margin-based MPE (M-MPE) loss function to be minimized is described by:

$$L_\rho^{(\text{M-MPE})}[p_\Lambda(X_r,\cdot),W_r] = \sum_{W\in\cdot} E(W,W_r)\frac{[p_\Lambda(X_r,W_r)\exp(-\rho A(W,W_r))]^\gamma}{\sum\limits_V [p_\Lambda(X_r,V)\exp(-\rho A(V,W_r))]^\gamma}, \tag{10}$$

It should be noted that due to the relation $E(V,W) = |W| - A(V,W)$ where $|W|$ denotes the number of symbols in the reference string, the error $E(V,W)$ and the accuracy $A(V,W)$ can be equally used in Equation 9 and Equation 10. The accuracy for MPE and for the margin term do not need to be the same quantity [29].

Finally, it should be pointed out that other posterior-based training criteria (e.g. MCE as used in [8]) can be modified in an analogous way to incorporate a margin term (for more details cf. [30, 31]).

3.3.5 Optimization

In [30] it is shown that the objective function $\mathscr{F}_\gamma^{(\text{MMI})}(\Lambda)$ converges pointwise to the SVM optimization problem using the hinge loss function for $\gamma \rightarrow \infty$, similar to [79]. In other words, $\mathscr{F}_\gamma^{(\text{M-MMI})}(\Lambda)$ is a smooth approximation to an SVM with hinge loss function which can be iteratively optimized with standard gradient-based optimization techniques like Rprop [30, 79].

In this work, the regularization constant C, the approximation level γ, and the margin scale ρ are chosen beforehand and then kept fixed during the complete optimization. Note that the regularization constant C and the margin scale ρ are not completely independent of each other. Here, we kept the margin scale ρ fixed and tuned the regularization constant C. Previous experiments in ASR have suggested that the performance is rather insensitive to the specific choice of the margin [30], and the results in [16] furthermore suggest that the choice of the I-smoothing constant C has less impact in an Rprop based optimization than in an Extendend Baum Welch (EBW) environment [66]. An I-smoothing regularization constant $C = 1.0$ is used in all results presented in Section 6.

In large vocabulary OCR, word lattices restricting the search space are used to make the summation over all competing hypotheses (i.e. sums over W) efficient. The exact accuracy on character or word level cannot be computed efficiently due to the Levenshtein alignments in general, although feasible under certain conditions as shown in [29]. Thus, the approximate character/word accuracy known from MPE/MWE [64] is used for the margin instead. With this choice of accuracy, the margin term can be represented as an additional layer in the common word lattices such that efficient training is possible. More details about the transducer-based implementation used in this work can be found in [29].

As in ASR, were typically a weak unigram language model is used for discriminative training [72, 73], we use a unigram language model in our proposed discriminative training criteria.

3.3.6 Confidences for Unsupervised Discriminative Model Adaptation

Sentence or word confidences can be incorporated into the training criterion by simply weighing the segments with the respective confidence. This is, however, not possible for state-based confidences. Instead of rejecting an entire sentence or word the system can use state confidence scores to select state-dependent data in an unsupervised manner. State confidence scores are obtained from computing arc posteriors from the lattice output from a previous decoder pass.

Rprop is a gradient-based optimization algorithm. The gradient of the training criterion under consideration can be represented in terms of the state posteriors $p_{rt}(s|x_1^{T_r})$. These posteriors are obtained by marginalization and normalization of the joint probabilities $p_\Lambda(x_1^{T_r}, s_1^T, w_1^{N_r})$ over all state sequences through state s at frame t. These quantities can be calculated efficiently by recursion, cf. forward/backward probabilities. Then, the state-based confidences $c_{r,s,t}$ are incorporated by multiplying the posteriors with the respective confidence before the accumulation. In summary, each frame t contributes $\cdot p_{rt}(s|x_1^{T_r}) \cdot c_{r,s,t} \cdot x_t$ to the accumulator acc_s of state s.

Another way to describe the incorporation of the confidence term into the margin pseudo-posteriors is from a system point of view. The accumulator acc_s of state s can be described by

$$\mathrm{acc}_s = \sum_{r=1}^{R} \sum_{t=1}^{T_r} \omega_{r,s,t} \cdot x_t ,$$

where the weight $\omega_{r,s,t}$, which is equal to $\delta(s_t, s)$ in ML training, is replaced for the proposed M-MMI-conf / M-MPE-conf criteria (with $\rho \neq 0$) by the margin modified pseudo-posteriors of the corresponding loss functions. The additional confidence term for the proposed M-MMI-conf criterion can be described as follows:

$$\omega_{r,s,t} := \frac{\sum\limits_{s_1^{T_r}:s_t=s} [p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(W_r)\cdot e^{-\rho\delta(W_r,W_r)}]^\gamma}{\underbrace{\sum_V \sum_{s_1^{T_r}:s_t=s} [p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(V)}_{\text{posterior}}\cdot \underbrace{e^{-\rho\delta(V,W_r)}}_{\text{margin}}]^\gamma} \cdot \underbrace{\delta(c_{r,s,t} \geq c_{\text{threshold}})}_{\text{confidence selection}} \qquad (11)$$

Here, the selector function $\delta(c_{r,s,t} > c_{\text{threshold}})$ with the parameter $c_{\text{threshold}}$ controls the amount of adaptation data. The M-MPE-conf criterion can be defined in a similar manner. Note that due to the quality of the confidence metric, thresholding the confidence scores after feature selection can often result in an improved accuracy, as reported in [25]. On the one hand, the experimental results for word-confidences in Figure 9 and state-based confidences in [16] suggest that the confidences are helpful, but on the other hand that the threshold itself has little impact due the proposed M-MMI-conf / M-MPE-conf approaches, which are inherently robust against outliers.

Analogously, the weight $\omega_{r,s,t}$ would correspond to the true posterior (cf. Equation 4) in an MMI-conf / MPE-conf criterion. According to [16,14,13] these criteria lead to no robust improvements, i.e. only the combination of margin *and* confidences makes the proposed approaches robust against outliers.

3.4 Writer Adaptive Training

Writer variations are compensated by writer adaptive training (WAT) [19] using constrained maximum likelihood linear regression (CMLLR) [23] to train writer dependent models. The available writer labels of the IfN/ENIT database are used in training to estimate the writer dependent CMLLR feature transformations. The parameters of the writer adapted Gaussian mixtures are trained using the CMLLR transformed features. During decoding, unsupervised writer clustering with Bayesian information criterion based stopping condition for a CMLLR based feature adaptation during a two-pass decoding process is used to cluster different handwriting styles of unknown test writers (cf. Section 3.5.2). It can be seen from the writer statistics in Table 1 that the number of different writers in set *e* is higher than in all other subsets, and thus the variation of handwriting styles. In machine-printed text recognition, the same approach could be applied to font labels available in the RAMP-N corpora (cf. Section 5.2).

3.5 Decoding Architecture

The recognition is performed in multiple passes. For model adaptation towards unknown data or unknown writing styles, the output of the first recognition pass (best word sequences or word lattices) can be either used for discriminative model adaptation (cf. Section 3.5.1) or writer adaptation (cf. Section 3.5.2). Although the automatically generated transcript may contain errors, adaptation using that transcript generally results in accuracy improvements [25]. The adaptation techniques used are explained in the following sections.

3.5.1 Discriminative Model Adaptation

The model adaptation can be carried out by discriminatively training writer dependent models using the word sequences obtained by the first recognition pass. Additionally, the confidence-alignments generated during the first-pass decoding can be used on a sentence-, word-, or state-level to exclude the corresponding features from the discriminative training process for unsupervised model adaptation.

Out-of-vocabulary (OOV) words are also meant to be harmful for adaptation [62] but even when a word is wrong, the pronunciation or most of the pronunciation can still be correct, suggesting that a state-based and confidence-based adaptation should be favored in such cases.

Word Confidences

As we are dealing with isolated word recognition on the IfN/ENIT database, the sentence and word confidences are identical. The segments to be used in the second-pass system are first thresholded on a *word-level* by their word confidences: only complete word *segments* aligned with a high confidence by the first-pass system are used for model adaptation using discriminative training.

State Confidences

Instead of rejecting an entire sentence or word, the system can use state confidence scores to select state-dependent data (cf. Section 3.3.6). State confidence scores are obtained from computing arc posteriors from the lattice output of the decoder. The arc posterior is the fraction of the probability mass of the paths that contain the arc from the mass that is represented by all paths in the lattice. The posterior probabilities can be computed efficiently using the forward-backward algorithm as, for example, described in [39]. Then, the word frames to be used in the second-pass system are first thresholded on a *state-level* by their state confidences: only word *frames* aligned with a high confidence by the first-pass system, are used for model adaptation using discriminative M-MMI-conf/M-MPE-conf training (see Section 3.3).

An example for a word-graph and the corresponding 1-best state alignment is given in Figure 5: during the decoding, the ten feature frames (the squares) can be aligned to different words (long arcs) and their states. In this example, the word-confidence of the 1-best alignment is $c = 0.7$ (upper arc). The corresponding state-

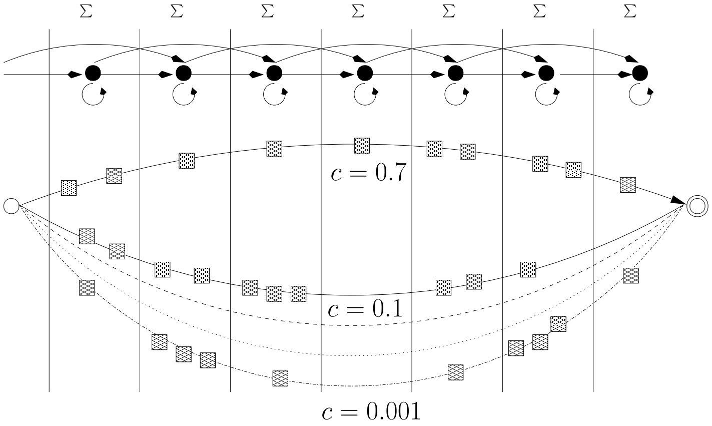

Fig. 5 Example for a word-graph and the corresponding 1-best state alignment: word-confidence of the 1-best alignment is $c = 0.7$. The corresponding state-confidences are calculated by accumulating state-wise over all other word alignments

confidences are calculated by accumulating state-wise over all competing word alignments (lower arcs), i.e. the state-confidence of the 1-best alignment's fourth state would stay 0.7 as this state is skipped in all other competing alignments, all other state-confidences would sum up to 1.0.

3.5.2 Writer Adaptation

The decoding in the second pass can be carried out using CMLLR transformed features. The segments to be recognized are first clustered using a generalized likelihood ratio clustering with Bayesian Information Criterion (BIC) based stopping condition [10]. The segment clusters act as writer labels required by the unsupervised adaptation techniques. The CMLLR matrices are calculated in pass two for every estimated writer cluster and are used for a writer dependent recognition system, which uses the models from the writer adaptive training of Section 3.4.

4 RWTH OCR Software Framework for Large Vocabulary OCR

The RWTH OCR software framework[5] is based on the RWTH Aachen University Open Source Speech Recognition System [68], short *RWTH ASR*. RWTH ASR has been designed for the special requirements of research applications. On the one hand it should be very flexible, to allow for rapid integration of new methods, and on the other hand it has to be efficient, so that new methods can be studied on real-life

[5] `http://www.hltpr.rwth-aachen.de/rwth-ocr/`

tasks in reasonable time and system tuning is feasible. The flexibility is achieved by a modular design, where most components are decoupled from each other and can be replaced at runtime. The API is subdivided into several modules and allows for an integration of (high and low level) methods in external applications.

The applicability of the toolkit to real-life speech recognition tasks has been proven by building several large vocabulary systems in recent international research projects, for example TC-STAR [43] (European English and Spanish), GALE [69, 63] (Arabic and Chinese), and QUAERO [56] (English, French, German, and Spanish).

A good example for the flexibility of the toolkit is the expeditious development of systems for continuous sign language recognition using video input [18] and for handwriting recognition [19, 15]. Only the feature extraction had to be replaced to adapt the system to these tasks. In the following sections, we will focus on the parts of the framework, which are relevant for OCR.

An important aspect for developing a system for a large vocabulary task is the support for grid-computing. Nearly all processing steps for training and decoding can be distributed in a cluster computer environment. The parallelization scales very well, because we divide the computations on the segment level, which requires synchronization only at the end of the computation.

The toolkit is published under an open source license, called "RWTH ASR License" and publicly available[6]. This RWTH ASR License grants free usage including re-distribution and modification for non-commercial use.

4.1 Feature Extraction

The feature extraction is implemented in a generic framework for data processing, called *Flow*. The data flow is modeled by links connecting several nodes to a network. Each node performs some type of data manipulation including loading, storing, and caching of data.

The networks are created at runtime based on a network definition in XML documents, which makes it possible to implement or modify data processing tasks without modifying and re-compiling the software. The individual nodes can be either instances of a C++ class or a subnetwork of other nodes.

By using cache nodes, data types sent through the network can be written to disk at any point in the network. The stored data can be read afterwards without repeating the computations of the nodes before the cache node.

Flow networks are used to compute feature vectors as well as to generate and process data alignments, i.e. mappings from feature vectors to HMM states. Using the caching nodes, features and alignments can be re-used in processing steps requiring multiple iterations.

[6] `http://www.hltpr.rwth-aachen.de/rwth-asr/`

4.2 Visual Modeling

A word is modelled by a sequence of glyph models. The writing variant model gives for each word in the vocabulary a list of glyph model sequences together with a probability of the variant's occurrence. The toolkit supports context dependent modeling of subunits (glyphs for OCR, phones for ASR) using decision trees for HMM state model tying. However, context dependent modeling has not been used so far for our OCR systems.

The toolkit supports strict left-to-right HMM topologies, each representing a (potentially context dependent) sub-word unit. All HMMs consist of the same number of states, except for a dedicated white-space (or silence) model. The transition model implements loop, forward, and skip transitions with globally shared transition probabilities.

The emission probability of an HMM state is represented by a Gaussian mixture model (GMM). By default, globally pooled variances are used. However, several other tying schemes, including density-specific diagonal covariance matrices are supported.

For the unsupervised refinement or re-estimation of model parameters the toolkit supports the generation and processing of confidence weighted state alignments. Confidence thresholding on state level is supported for unsupervised training as well as for unsupervised adaptation methods. The toolkit supports different types of state confidence scores, most described in [25]. The emission model can be re-estimated based on the automatically annotated observations and their assigned confidence weights, as presented in [26, 15].

4.3 Model Adaptation

The software framework supports maximum likelihood linear regression (MLLR) and feature space MLLR (fMLLR) (also known as constrained MLLR, CMLLR) for writer adaptive modeling.

The fMLLR consists of normalizing the feature vectors by the use of a maximum likelihood estimated affine transform, as described in [23]. As an extension, the estimation of dimension reducing affine transforms, as described in [42], is supported. fMLLR is implemented in the feature extraction front-end, allowing for use in both recognition and in training, thus supporting writer adaptive training [19].

For Maximum Likelihood Linear Regression (MLLR) [40] affine transforms are applied to the means of the visual model. A regression class tree approach [41] is used to adjust the number of regression classes to the amount of adaptation data available. As a variation, it is possible to do adaptation using only the offset part (and not the matrix part) of the affine transform.

The adaptation methods can be utilized both for unsupervised and supervised adaptation. The transformation estimation can make use of weighted observations allowing for confidence based unsupervised adaptation.

4.4 Language Modeling

The toolkit does not include tools for the estimation of language models. However, the decoder supports N-gram language models in the ARPA format, produced e.g. by the SRI Language Modeling Toolkit [75]. The order of the language model is not limited by the decoder. Class language models, defined on word classes instead of words, are supported as well. Alternatively, a weighted finite state automaton representing a (weighted) grammar can be used.

4.5 Decoder

The decoder included in our toolkit is based on the word conditioned tree search [53]. Word conditioned tree search is a one-pass dynamic programming algorithm which uses a pre-compiled lexical prefix tree as representation of the writing variants dictionary. When using a tree lexicon, the word identity is not known until a leaf node is reached. Therefore, the language model (LM) probability can only be applied at the word end, although an early incorporation of the LM can be achieved using LM look-ahead. To make the application of the dynamic programming principle possible, the search space has to be structured by introducing separate copies of the lexical tree for each preceding word sequence. The length of this word sequence depends on the order of the language model used, e.g. for a bigram language model only the direct predecessor word is required.

The search space would be too large to be constructed as a whole, instead only the active portions are constructed dynamically in combination with a beam search. The beam search strategy retains for every time step only the most promising hypotheses. Hypotheses with a too low score compared to the best state hypothesis are eliminated by *state pruning*. The beam width, i.e. the number of surviving hypotheses, is defined by a threshold. *Language model pruning* is applied to the word start hypotheses after applying the language model, which limits the number of active tree copies. In addition, histogram pruning restricts the absolute number of active hypotheses.

The state pruning can be refined by incorporating the language model probabilities as early as possible using a language model look-ahead [59]. The anticipated language model probability for a certain state in the tree is approximated by the best word end reachable. This probability is incorporated in the pruning process by combining it with the probability of the state hypothesis.

The decoder can also generate a word graph (also called lattice) which is a compact representation of the set of alternative word sequences with corresponding word boundaries [58]. This word graph can be used in later processing steps. Our system produces word graphs as finite state automata with attached word boundaries or alternatively in the HTK standard lattice format.

The computation of emission probabilities can be optionally accelerated by the use of SIMD instructions provided by modern processors [38]. The feature vectors as well as the means of the Gaussian mixture models are then transformed to integers using a scalar quantization. The following computations on these quantized vectors are performed using MMX or SSE2 instructions.

4.6 Documentation

The documentation is divided into two parts: usage documentation and source code documentation. While the source code documentation is helpful for extending the software, the usage documentation is more comprehensive and more relevant for the normal user.

The usage documentation is organized in a wiki and covers all steps of the model training, multi-pass recognition, and describes the common concepts of the software and the used file formats. Emerging questions can be asked in a support forum.

5 Datasets

In the following we describe the corpora we used for closed-vocabulary isolated handwritten word recognition, and our novel Arabic newspaper corpus for open-vocabulary machine-printed text recognition tasks.

5.1 IfN/ENIT Arabic Handwriting Database

The IfN/ENIT database is divided into four training subsets with an additional fold for testing [49]. The current database version (v2.0p1e) contains a total of 32492 Arabic words handwritten by about 1000 writers, and has a vocabulary size of 937 Tunisian town names. Here, we follow the same evaluation protocol as for the ICDAR 2005, 2007, 2009, and ICFHR 2010 competitions [48, 50]. The corpus statistics for the different subsets can be found in Table 1.

It should be noted that all experiments with this database in the following sections were done without any pruning, and thus the improvement of the system accuracy is due to the proposed refinement methods only.

5.2 The RWTH Arabic Machine-Print Newspaper Corpus

In 1995, the DARPA Arabic machine-print (DAMP) corpus was collected by SAIC [11, 52]. It consists of 345 images from newspapers, books, magazines, etc., but is not publicly available.

Table 1 Corpus statistics for the IfN/ENIT Arabic handwriting sub-corpora.

Subsets	#Observations [k]			
	Writers	Words	Characters	Frames
a	0.1	6.5	85.2	452
b	0.1	6.7	89.9	459
c	0.1	6.5	88.6	452
d	0.1	6.7	88.4	451
e	0.5	6.0	78.1	404
f	n.a.	8.6	64.7	n.a.
s	n.a.	1.5	11.9	n.a.

The synthetic APTI database [74] for Arabic machine-printed documents offers many synthetically rendered fonts but seems unsuitable for large vocabulary and domain specific OCR tasks.

In [2] a Multi-Modal Arabic Corpus (MMAC)[7] containing a list of six million Arabic words is presented, which may be used as a lexical lookup table to check the existence of a given word. However, no large amounts of image segments with corresponding ground-truth annotations to be used in OCR experiments are currently provided. Recently, the PATDB [4] has been presented, which will be interesting for future work, but which is not yet available.

The objective of the MADCAT [57] project is to produce a robust, highly accurate transcription engine that ingests documents of multiple types, especially Arabic scripts, and produces English transcriptions of their content. Some parts of the Arabic handwriting data, which was created by the Linguistic Data Consortium (LDC) and used in previous MADCAT evaluations [52], has been recently used for the OpenHaRT 2010 [1] competition. However no machine-printed documents have been provided so far.

Therefore we started in 2010 with the generation of the large vocabulary RWTH Arabic Machine-Print Newspaper (RAMP-N) corpus[8] suitable for OCR research, by collecting more than 85k PDF pages of newspaper articles from the following websites:

- `http://www.addustour.com` (Lebanon)
- `http://www.albayrakonline.com` (Jordan)

In our current collection (cf. Table 2), the newspaper data in the training corpus ranges from April to May 2010, development corpus from May 2010, and the evaluation corpora were collected in September 2010.

We automatically generate ground-truth annotations with the freely available PDFlib Text Extraction Toolkit (TET)[9], which reliably extracts Unicode text, im-

[7] `http://www.ashrafraouf.com/mmac`

[8] `http://www.hltpr.rwth-aachen.de/~dreuw/arabic.php`

[9] `http://www.pdflib.com/products/tet/`

Table 2 RAMP-N corpora statistics

	Train	Dev	Eval a	Eval b	Eval c	LM Training
Running words	1,483,136	7,775	20,042	17,255	15,290	228,492,763
Running Characters	5,970,997	30,884	72,358	64,293	62,065	989,494,230
Text lines	222,421	1,155	3,480	2,439	2,224	22,910,187
Pages	409	2	5	4	4	85,316
Fonts	20	5	12	7	6	-
OOV Rate	1.90%	2.79%	2.21%	2.90%	2.75%	-

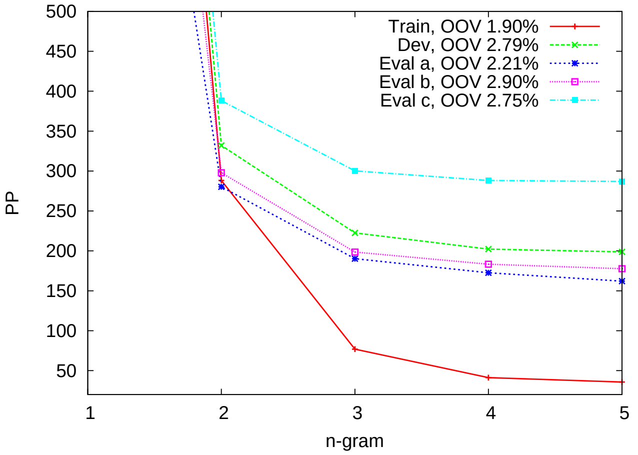

Fig. 6 Perplexities (PP) for different n-gram contexts using modified Kneser-Ney smoothing and a vocabulary size of 106k words

ages and metadata from PDF documents. Additionally, detailed glyph and font information as well as the position on the page can be extracted.

In addition to the 28 Arabic base forms, and after filtering out texts with Latin glyphs, the Arabic texts in our current collection include 33 ligatures, 10 Arabic-Indian digits, and 24 punctuation marks. They are modeled by 95 position independent or by 197 position dependent glyph HMMs [67, 17]. The position dependent glyph transcriptions have been created by a rule based approach based on the six Arabic characters, which have only an isolated or final form [44].

Text Corpora. About 228M running words have been collected for domain specific language model (LM) estimation. As vocabulary we currently use the 106k most frequent words of the 228M LM data corpus, resulting in about 126k writing variants due to ligatures, an average out-of-vocabulary (OOV) rate of 2.5% (cf. Table 2), and a 0% out-of-glyph rate. None of the segments in the development or evaluation corpora belong to the LM training data. The resulting perplexities, which are relatively high due to the rich morphology in Arabic, for different n-gram language models using modified Kneser-Ney smoothing are presented in Figure 6.

6 Experimental Results

The proposed approach is applied to isolated Arabic handwritten and continuous Arabic machine-printed texts. The experiments for isolated word recognition are conducted on the IfN/ENIT database [61] using a closed lexicon, experiments for continuous line recognition on the novel large vocabulary RWTH Arabic Machine-Print Newspaper (RAMP-N) corpus.

6.1 First Pass Decoding

In this section we compare our ML trained baseline systems (cf. Section 3.2 for visual model details) to our discriminatively trained systems using the MMI and MPE criteria and their margin-based extensions.

Each of the 120 glyph models in our Arabic handwriting recognition base system is modeled by a 3-state left-to-right HMM with three separate GMMs. The position dependent glyph model of our ML trained baseline system includes 361 mixtures with 36k Gaussian densities with globally pooled diagonal variances.

The discriminative training is initialized with the respective ML trained baseline model and iteratively optimized using the Rprop algorithm (cf. Section 3.3). For isolated Arabic word recognition on the IfN/ENIT database, we compare our ML trained baseline system with MMI/M-MMI criteria only.

6.1.1 Discriminative GHMMs

In general, the number of Rprop iterations and the choice of the regularization constant C have to be chosen carefully (cf. optimization in Section 3.3), and were empirically optimized in informal experiments to 30 Rprop iterations and $C = 1.0$ (cf. detailed Rprop iteration analysis and convergence without over-training in Figure 8).

The results in Table 3 show that the discriminatively trained models clearly outperform the ML trained baseline models, especially the models trained with the additional margin term. The strong decrease in word error rate (WER) for experiment setup *abd-c* might be due to the training data being separable for the given configurations, whereas the strong improvement for experiment *abcde-e* was expected because of the test set *e* being part of the training data.

In the following experiments, we additionally use glyph dependent lengths (GDL) as described in [17, 19], resulting in ML trained baseline model with 216 glyph models, 646 mixtures, and up to 55k densities (cf. Section 3.2). The necessity of this glyph dependent model length estimation is exemplified by visualizing the state alignment in Figure 7. Different background colors are used for the respective HMM states.

By estimating glyph dependent model lengths, the overall mean of glyph length changed from 7.89px (i.e. 2.66 px/state) to 6.18px (i.e. 2.06px/state) when down-

Table 3 Comparison of ML trained baseline systems, and discriminatively trained systems using MMI and M-MMI criteria after 30 Rprop iterations on the IfN/ENIT database.

Train	Test	WER[%]		
		ML	MMI	M-MMI
abc	d	10.88	10.59	**8.94**
abd	c	11.50	10.58	**2.66**
acd	b	10.97	10.43	**8.64**
bcd	a	12.19	11.41	**9.59**
abcd	e	21.86	21.00	**19.51**
abcde	e	11.14	2.32	**2.95**

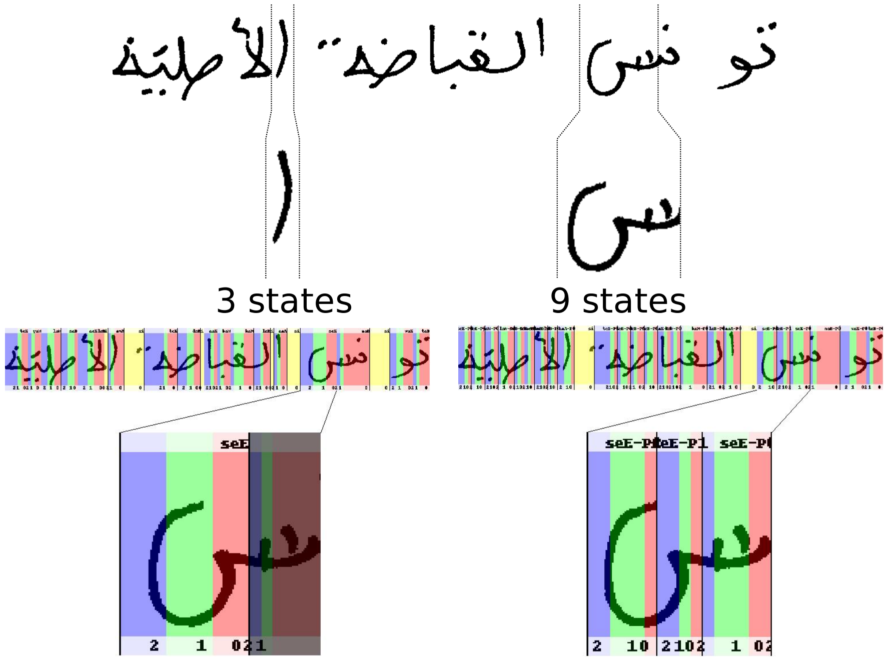

Fig. 7 Top: more complex characters should be represented by more states. Bottom: using GDL glyph models, frames previously aligned to a wrong neighboring glyph model (left, black shaded) are aligned to the correct glyph model (right).

scaling the images to 16px height while keeping their aspect-ratio. Thus every state of an GDL glyph model has to cover less pixels due to the relative reduction of approx. 20% pixels.

In Figure 8 detailed WER and character error rate (CER) plots over M-MMI training iterations are shown. It can be observed that both WER and CER are smoothly and almost continuously decreasing with every Rprop iteration, and that about 30 Rprop iterations are optimal for the considered datasets.

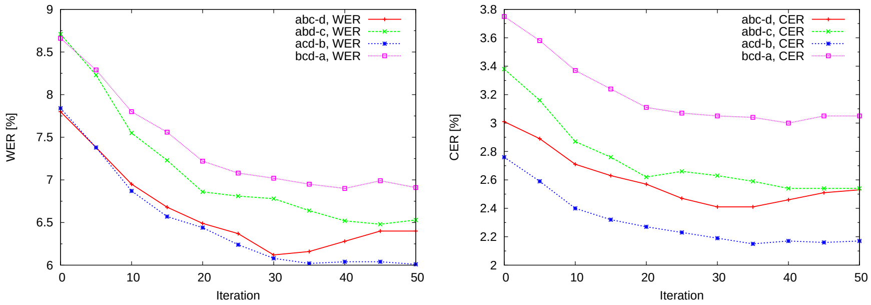

Fig. 8 Decreasing word error rates (WER) character error rates (CER) and for all different training subsets of the IfN/ENIT database over M-MMI Rprop iterations (baseline with glyph dependent length (GDL) estimation).

Table 4 Results for margin-based M-MMI criterion after 30 Rprop iterations on the IfN/ENIT database using glyph dependent lengths (GDL).

Train	Test	WER[%]			
		ML	GDL	+MMI	+M-MMI
abc	d	10.88	7.83	7.4	**6.12**
abd	c	11.50	8.83	8.2	**6.78**
acd	b	10.97	7.81	7.6	**6.08**
bcd	a	12.19	8.70	8.4	**7.02**
abcd	e	21.86	16.82	16.4	**15.35**

The final results for discriminative GHMM training with additional glyph dependent lengths estimation are presented in Table 4.

6.1.2 Hybrid MLP/HMM vs. Tandem MLP-GHMM

Due to a position and glyph-dependent length modeling of the 28 base Arabic characters [17], we finally model the Arabic words in the IfN/ENIT database by 216 different glyph models (i.e., 215 glyphs and one white-space model). The system described in [15] (cf. also M-MMI column in Table 7) is used to generate an initial alignment of the features to the 216 labels. Our discriminative GHMM baseline system (cf. Table 5) uses 3 mixtures per glyph label, resulting in up to 646 mixtures with 55k densities. The MLP networks have been trained on raw pixel column features from the sets *a*, *b*, and *c* only.

RAW MLP Features. The hierarchical system uses at the first level no windowing of the input features, a single hidden layer with 2000 nodes, and 216 output nodes, which are reduced by a log-PCA transformation to 32 components. The second network concatenates these features in addition to the raw features, and uses a window size of 9 consecutive features The 576-dimensional features (i.e. 32 × 2 × 9 fea-

Table 5 System comparison: MLP-GHMM performs best, both GHMM and MLP-GHMM systems are M-MMI trained

Train	Test	GHMM		MLP/HMM		MLP-GHMM	
		WER[%]	CER[%]	WER[%]	CER[%]	WER[%]	CER[%]
abc	d	6.12	2.41	4.54	1.70	3.47	1.50
abd	c	6.78	2.63	2.64	0.93	1.38	0.75
acd	b	6.08	2.19	2.70	0.87	2.52	0.98
bcd	a	7.02	3.05	3.11	1.32	2.60	1.09
abcd	e	15.35	6.14	11.57	4.54	7.26	3.03

tures) are forwarded to a single hidden layer with 3000 nodes, and reduced by a log-PCA transformation to 32 components.

TRAP-DCT MLP Features. The system uses a TRAP-DCT [33] preprocessing of the raw pixel input features. The TRAP-DCT preprocessing for sliding window image patches can be interpreted as a modular block-based DCT of the patches at image row level. The hierarchical system uses at the first level a spatio-temporal TRAP-DCT window to augment the 32-dimensional raw pixel input feature vectors to a 256-dimensional vector. Again, the first level hierarchical network uses a single hidden layer with 1500 nodes, and 216 output nodes, which are reduced by a log-LDA transformation to 96 components. The second network concatenates these features in addition to the raw features, and uses a window size of 5 consecutive log-LDA network features, and a window size of 9 consecutive raw input features to account for different spatio-temporal information. The 768-dimensional features (i.e. $96 \times 5 + 32 \times 9$ features) are forwarded to a single hidden layer with 3000 nodes, and finally reduced by a log-LDA transformation to 36 components.

We empirically optimized RAW, TRAP-DCT, and feature combinations on the different IfN/ENIT training subsets, which showed no significant difference. The TRAP-DCT log-posterior features are used in Table 5 for the hybrid MLP/HMM approach, which turned out to perform slightly better than the RAW features in these informal experiments. Furthermore, we observed that a discriminative MLP-GHMM system is about 25% relative better than a generatively trained one, especially in combination with the concatenated RAW+TRAP-DCT features. The comparison in Table 5 shows a significant advantage of the retrained MLP-GHMM system over the hybrid MLP/HMM and the GHMM baseline. The achieved 7.26 % WER on evaluation *set e* is about 50% relatively better than the M-MMI trained baseline system, and to the best of the authors knowledge, outperforms all error rates reported in the literature.

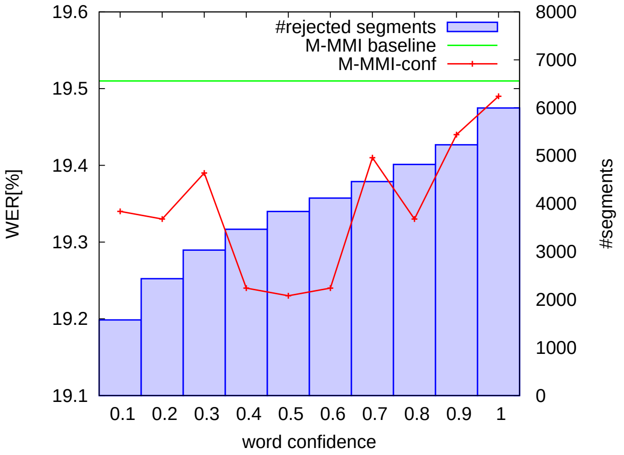

Fig. 9 Results for word-confidence based M-MMI-conf training on the on the evaluation setup *abcd-e* of the IfN/ENIT database using different confidence thresholds and their corresponding number of rejected segments (baseline without glyph dependent length (GDL) estimation).

6.2 Second Pass Decoding and Unsupervised Model Adaptation

In this section we evaluate our discriminative training for unsupervised model or writer adaptation during a second pass decoding step.

6.2.1 Confidence-Based Discriminative GHMMs

In a first experiment we used the complete first-pass output of the M-MMI system for an unsupervised model adaptation. The results in Table 6 show that the M-MMI based unsupervised adaptation without confidences cannot improve the system accuracy. With every Rprop iteration, the system is even more biased by the relatively large amount of wrong transcriptions in the adaptation corpus.

The discriminative M-MMI-conf training is initialized with the respective M-MMI trained model and iteratively optimized using the Rprop algorithm (cf. Section 3.3). Using the word-confidences for M-MMI-conf based model adaptation of our first-pass alignment to reject complete word segments (i.e. feature sequences X_1^T) from the unsupervised adaptation corpus, the results in Table 6 show a slight improvement only in comparison to the M-MMI trained system. Figure 9 shows the resulting WER for different confidence threshold values and the corresponding number of rejected segments. For a confidence threshold of $c = 0.5$, more than 60% of the 6033 segments of set e are rejected from the unsupervised adaptation corpus, resulting in a relatively small amount of adaptation data.

Using the state-confidences for M-MMI-conf based model adaptation of our first-pass alignment to decrease the contribution of single frames (i.e. features x_t) during

Table 6 Results for M-MMI-conf model adaptation on the evaluation setup *abcd-e* of the IfN/ENIT database after 30 Rprop iterations (baseline without glyph dependent length (GDL) estimation).

Training/Adaptation	WER[%]	CER[%]
ML	21.86	8.11
M-MMI	19.51	7.00
+ unsupervised adaptation	20.11	7.34
+ supervised adaptation	2.06	0.77
M-MMI-conf (word-confidences)	19.23	7.02
M-MMI-conf (state-confidences)	**17.75**	**6.49**

Table 7 Results for confidence-based M-MMI-conf model adaptation after 15 Rprop iterations on the IfN/ENIT database using glyph dependent lengths (GDL), and margin-based M-MMI criterion after 30 Rprop iterations.

Train	Test	WER[%]				
		1st pass				2nd pass
		ML	GDL	+MMI	+M-MMI	M-MMI-conf
abc	d	10.88	7.83	7.4	**6.12**	**5.95**
abd	c	11.50	8.83	8.2	**6.78**	**6.38**
acd	b	10.97	7.81	7.6	**6.08**	**5.84**
bcd	a	12.19	8.70	8.4	**7.02**	**6.79**
abcd	e	21.86	16.82	16.4	**15.35**	**14.55**

the iterative M-MMI-conf optimization process (cf. optimization in Section 3.3), the number of features for model adaptation is reduced by approximately 5% for a confidence threshold of $c_{\text{threshold}} = 0.5$: 375 446 frames of 396 416 frames extracted from the 6033 test segments are considered during the optimization, only 20 970 frames are rejected based on state-confidence thresholding (cf. also Figure 5). Note that also the CER is decreased to 6.49%.

Interestingly, the supervised adaptation on test set *e*, where only the correct transcriptions of set *e* are used for an adaptation of the model trained using set *abcd*, can again decrease the WER of the system down to 2.06%, which is even better than an M-MMI optimization on the full training set *abcde* (cf. Table 3).

Table 7 shows the final results of our Arabic handwriting recognition system with additional glyph dependent lengths (GDL) as described in [19]. Again, the WER of the GDL based system can be decreased by our proposed M-MMI training during both decoding passes down to 14.55%.

In Figure 10 a combined WER/CER plot over M-MMI-conf training iterations on the evaluation setup *abcd-e* (cf. initialization plots) is shown. It can be observed that both WER and CER are slightly decreasing with every Rprop iteration, and that between 10 and 15 Rprop iterations are optimal for the considered small amount of unsupervised labeled test datasets. Due to the robustness of the confidence- and margin-based M-MMI-conf criterion against outliers, the proposed unsupervised

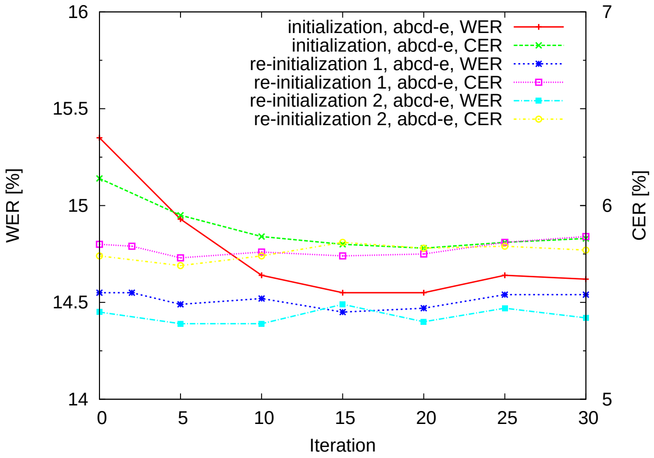

Fig. 10 Evaluation of iterative M-MMI-conf model adaption on the evaluation setup *abcd-e* of the IfN/ENIT database : text transcriptions are updated in an unsupervised manner after 15 Rprop iterations. The performance remains robust even after several re-initializations.

and text dependent model adaptation can even be applied in an iterative manner by a re-initialization of the text transcriptions. In Figure 10, we re-initialize 2 times the model adaptation process after 15 Rprop iterations. The results in Figure 10 show the robustness of our approach, leading to a slightly improved WER of 14.39%.

6.2.2 Writer Adaptation

The writer adaptive trained (WAT) models (cf. Section 3.4) can also be used as a first pass decoding system. The results in Table 8 show that the system performance cannot be improved without any writer clustering and adaptation of the features during the decoding step.

To show the advantage of using CMLLR based writer adapted features in combination with WAT models, we estimate in a first *supervised* experiment the CMLLR matrices directly from the available *writer labels* of the test subsets. The matrices are calculated for all writers in pass two and are used for a writer dependent recognition system, which uses the WAT models from Section 3.4. Note that the decoding itself is still unsupervised!

In the unsupervised adaptation case, the unknown writer labels of the segments to be recognized have to be estimated first using BIC clustering. Again, the CMLLR matrices are calculated in pass two for every estimated cluster label and are used for a writer dependent recognition system, which uses the WAT models from Section 3.4.

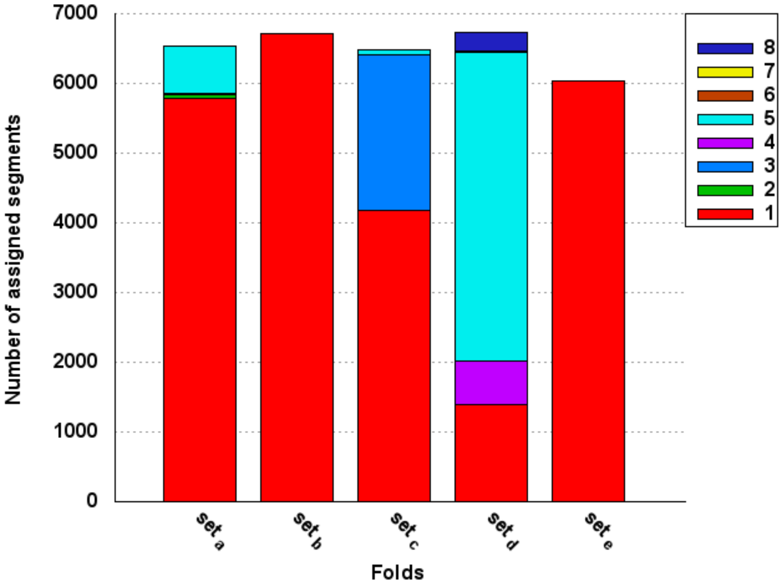

Fig. 11 Histograms for unsupervised clustering over the different test subsets and their resulting unbalanced segment assignments.

Table 8 shows that the system accuracy could be improved by up to 33% relative in the supervised-CMLLR adaptation case. In the case of unsupervised writer clustering, the system accuracy is improved in one fold only.

If we look at the cluster histograms in Figure 11 it becomes clear that the unsupervised clustering is not adequate enough. Each node in our clustering process as described in [10] is modeled as a multivariate Gaussian distribution $\mathcal{N}(\mu_i, \Sigma_i)$, where μ_i can be estimated as the sample mean vector and Σ_i can be estimated as the sample covariance matrix. The estimated parameters are used within the criterion as distance measure, but more sophisticated features than the PCA reduced sliding window features seem necessary for a better clustering, which will be interesting for future work.

Opposed to the supervised estimation of 505 CMLLR transformation matrices for the evaluation setup with training sets *abcd* and test set *e* (cf. Table 1), the unsupervised writer clustering could estimate only two clusters being completely unbalanced, which is obviously not enough to represent the different writing styles of 505 writers. Due to the unbalanced clustering and only a small number of clusters, all other cases are similar to the usage of the WAT models only (cf. Table 8).

However, the supervised-CMLLR adaptation results show that a good writer clustering can bring the segments of the same writer together and thus improve the performance of the writer adapted system.

Table 8 Comparison of GDL, WAT, and CMLLR based feature adaptation using unsupervised and supervised writer clustering.

Train	Test	WER[%]				
		1st pass			2nd pass	
		ML	+GDL	+WAT	WAT+CMLLR	
					unsup.	sup.
abc	d	10.88	**7.83**	7.54	7.72	**5.82**
abd	c	11.50	**8.83**	9.09	9.05	**5.96**
acd	b	10.97	**7.81**	7.94	7.99	**6.04**
bcd	a	12.19	**8.70**	8.87	8.81	**6.49**
abcd	e	21.86	**16.82**	17.49	17.12	**11.22**

6.3 Visual Inspections

The visualizations in Figure 12 show training alignments of Arabic words to their corresponding HMM states. The upper rows show the alignment to the ML trained model, the lower rows to the M-MMI trained models. We use R-G-B background colors for the 0-1-2 HMM states, respectively, from right-to-left. The position dependent glyph model names (cf. Section 3.2) are written in the upper line, where the white-space models are annotated by 'si' for 'silence'; the state numbers are written in the bottom line. Thus, HMM state-loops and state-transitions are represented by no-color-changes and color-changes, respectively.

It can be observed in the left column of Figure 12 that especially the white-spaces, which can occur between compound words and Parts of Arabic Words (PAWs) [17], help in discriminating the isolated- (A), beginning- (B), or end-shaped (E) glyphs of a word w.r.t. the middle-shaped (M) glyphs, where usually no white-spaces occur on the left or right side of the character (cf. [61, 44] for more details about A/B/M/E shaped characters). The frames corresponding to the white-space part of the words are aligned in a more balanced way in Figure 12(a) and Figure 12(b) using the M-MMI modeling (lower rows) opposed to ML modeling (upper rows): the proposed M-MMI models learned that white-spaces help to discriminate different glyphs. This can even lead to a different writing variant choice without any white-space models [17] (see Figure 12(c)). Note that we cannot know in advance in training if a white-space is used or not, and if so, how large it is, as it is not transcribed in the corpora and depends on the writer's handwriting style (e.g. cursive style used in Figure 12(a)).

In the right column of Figure 12, unsupervised test alignments are compared. The upper rows show incorrectly recognized words by unsupervised alignments to the ML trained model, the lower rows correctly recognized words by unsupervised alignments to the M-MMI trained models. Due to the discriminatively trained glyph models, the alignment in Figure 12(d) to the M-MMI model is clearly improved over the ML model, and the system opts for the correct compound-white-space writing variant [17]. In Figure 12(e), again the alignment is improved by the dis-

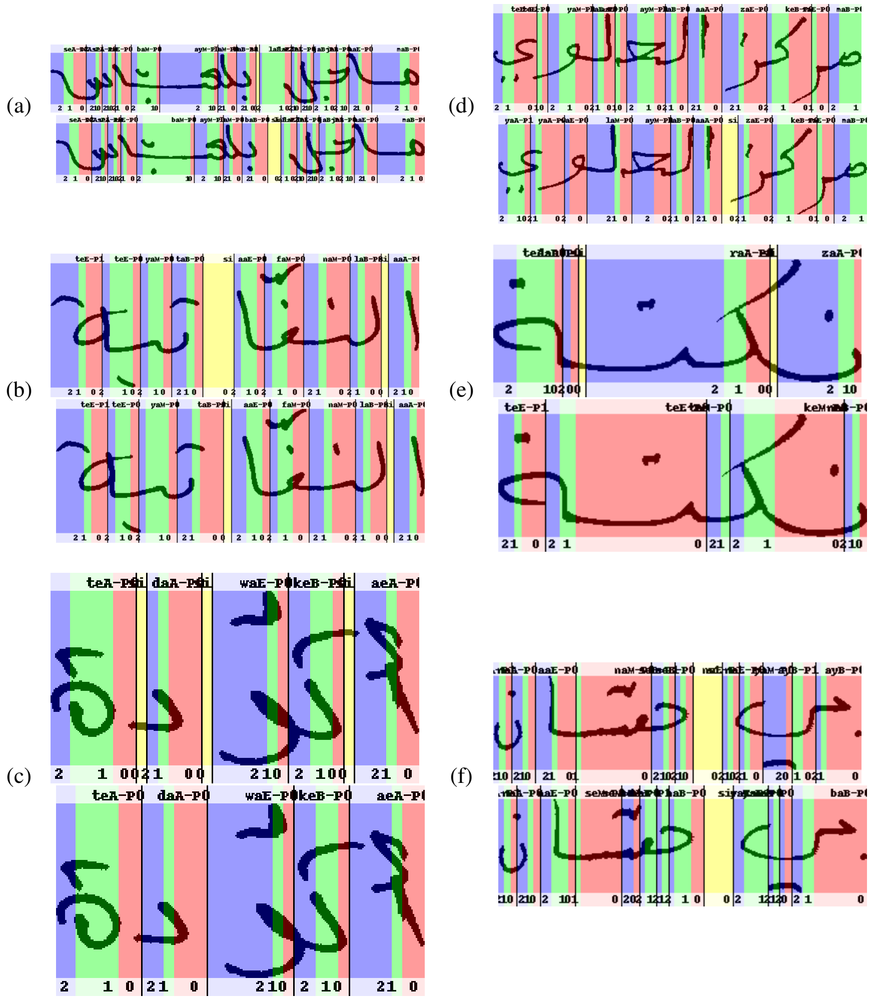

Fig. 12 Left column: Supervised training alignment comparisons - The upper rows show alignments to the maximum-likelihood (ML) trained model, the lower rows to the margin-based maximum mutual information (M-MMI) trained models. Right column: Unsupervised test alignment comparisons: The upper rows show incorrect unsupervised alignments to the ML trained model, the lower rows correct unsupervised alignments to the M-MMI trained models.

criminatively trained white-space and glyph models. Figure 12(f) shows a similar alignment to the white-space model, but a clearly improved and correct alignment to the discriminatively trained glyph models.

6.4 Comparisons with other Systems

IfN/ENIT Competitions at ICDAR/ICFHR. In Table 9 we compare or own evaluation results on the ICDAR 2005 [49] setups (without any tuning on test data as explained in Section 6.2) and ICDAR 2007/2009 and ICFHR 2010 [48, 50] setups. It should be noted that the result for the *abcd-e* condition is the best known error rate in the literature [20].

The ICDAR 2009 test datasets which are unknown to all participants were collected for the tests of the ICDAR 2007 competition. The words are from the same lexicon as those of the IfN/ENIT database and written by writers, who did not contribute to the data sets before, and are separated into set f and set s. Our results (externally calculated by TU Braunschweig) in Table 9 ranked third at the ICDAR 2009 competition and are among the best purely HMM based systems, as the A2iA and MDLSTM systems are hybrid system combinations or full neural network based systems, respectively. Also note that our single HMM based system is better than the independent A2iA systems (cf. [48] for more details). In particular, our proposed M-MMI-conf based approach for unsupervised model adaptation even generalizes well on the *set s*, which has been collected in the United Arabic Emirates and represents significantly different handwriting styles.

Note the 36% relative improvement in Table 9 we achieved in the recent ICFHR 2010 Arabic handwriting competition [50] with the proposed M-MMI training framework and an MLP based feature extraction. Our system ranked second and used again no system combinations. Interesting is the result of the UPV PRHLT group who significantly improved their relatively simple baseline system due to a vertical centroid normalization of sliding window based features [50, 24]. Note that our MLP-GHMM does not perform any preprocessing.

6.5 Machine-Printed Arabic Text Recognition

In a first set of experiments we optimized the feature extraction parameters and compared position independent and dependent glyph models, using single-density models only. In both cases we used glyph HMMs with 6 states with skip transitions and 3 separate GMMs with a globally pooled covariance matrix. The results for the development set of the RAMP-N database (cf. Section 5.2) in Figure 13 show an error rate reduction of about 50% relative for position dependent glyph models compared to a position independent glyph modeling. Note that we empirically optimized the PCA reduction to 30 components, and that the feature extraction parameters are similar to those used in handwritten text recognition.

Some examples of the professional ArabicXT fonts[10] occurring in the RAMP-N corpus, which are widely used by newspapers, magazines, or book publishers, are shown in Figure 14.

[10] `http://www.layoutltd.com/arabicxt.php`

Table 9 Comparison to ICDAR/ICFHR Arabic handwriting recognition competition results on the IfN/ENIT database

Competition	Group	WER [%]			
		abc-d	abcd-e	abcde-f	abcde-s
ICDAR 2005 [49]	UOB	15.00	24.07		
	ARAB-IFN	12.06	25.31	-	-
	ICRA (Microsoft)	11.05	34.26	-	-
ICDAR 2007 [47]	SIEMENS [70]	-	18.11	12.78	26.06
	MIE (DP)	-	-	16.66	31.60
	UOB-ENST (HMM)	-	-	18.07	30.07
ICDAR 2009 [48]	MDLSTM	-	-	**6.63**	18.94
	A2iA (combined)	-	-	10.58	23.34
	(MLP/HMM)	-	-	14.42	29.56
	(HMM)	-	-	17.79	33.55
	RWTH OCR (this work,M-MMI)	6.12	15.35	14.49	28.67
	RWTH OCR (this work, M-MMI-conf)	5.95	14.55	14.31	27.46
ICFHR 2010 [50]	UPV PRHLT (HMM)	7.50	12.30	7.80	**15.38**
	RWTH OCR (this work, MLP-GHMM)	**3.47**	**7.26**	9.12	18.94
	UPV PRHLT (HMM, w/o vert. norm.)	-	-	12.09	21.55
	CUBS-AMA (HMM)	-	-	19.68	32.10
Other results	BBN [51]	10.51	-	-	-

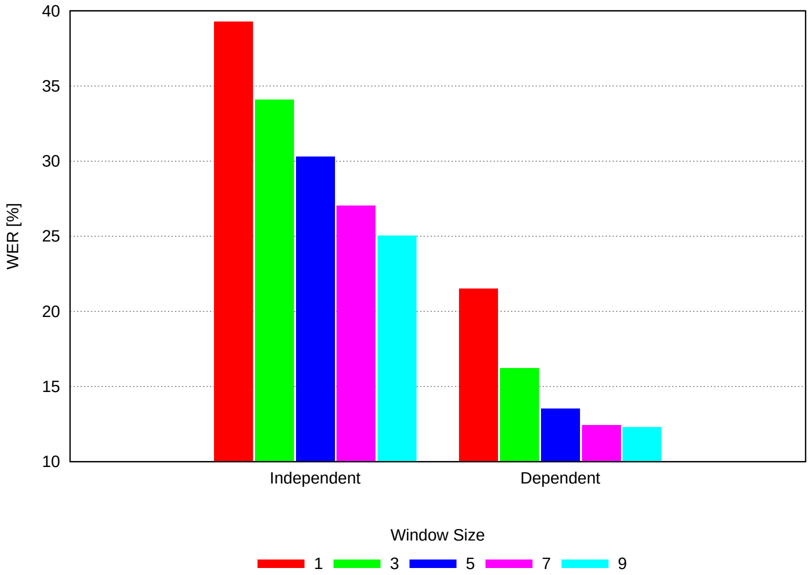

Fig. 13 Comparison of position independent and dependent glyph modeling on the RAMP-N development corpus, using single-density models, and PCA reduced appearance-based sliding window features.

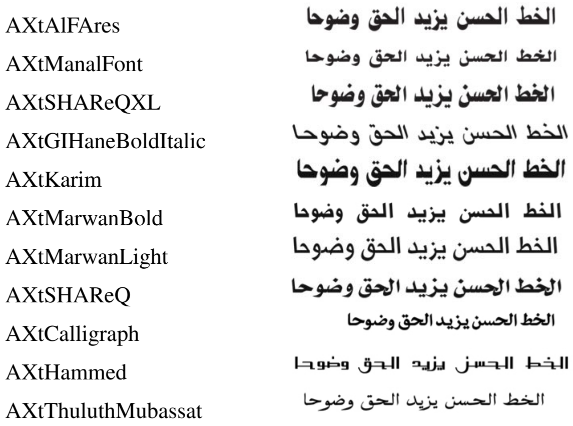

Fig. 14 Some examples of various professional newspaper fonts used in the RAMP-N corpora (example images taken from `http://www.layoutltd.com/`)

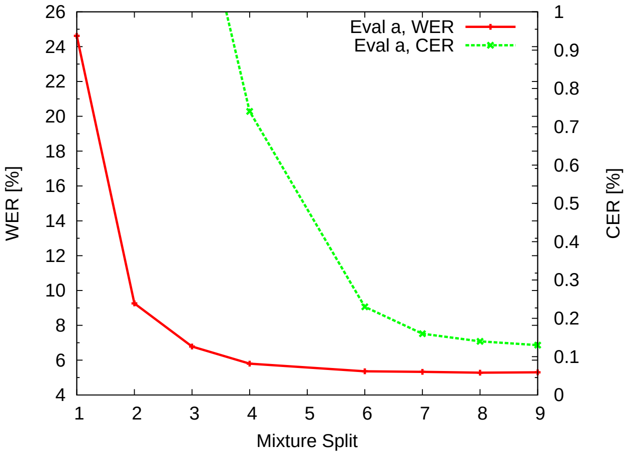

Fig. 15 Results for position dependent GMMs on the RAMP-N subset Eval a

Experiments with Gaussian mixture models (GMMs) instead of single densities in Figure 15 improve the WER/CER as expected, as they implicitly model the up to 20 different font appearances in the corpora. Note that glyph dependent length (GDL) models as e.g. successfully used for handwriting in [17, 15, 24] (also cf. Figure 7) lead only to small improvements so far for machine-printed text recognition.

The results in Table 10 show detailed results for each font appearing in the RAMP-N subset Eval a: high WER but low CER are due to OOV words, which are often recognized as a sequence of PAWs instead of a single word, resulting in one substitution and many insertion errors, but zero edits at the character level. Simply replacing those word sequences between white-space blocks can further reduce

Table 10 Font-wise results for ML trained GMMs on the RAMP-N subset Eval a.

Font	Lines	Errors	Words	OOV	WER[%]	Errors	Glyphs	CER[%]
AXtAlFares	2	10	2	2	500.00	0	19	0.00
AXtCalligraph	1	0	8	0	0.00	0	21	0.00
AXtGIHaneBoldItalic	15	19	129	4	14.73	12	591	2.03
AXtHammed	3	0	5	0	0.00	0	31	0.00
AXtKaram	9	2	83	0	2.41	4	300	1.33
AXtManal	1	0	2	0	0.00	0	4	0.00
AXtManalBlack	5	5	27	1	18.52	11	112	9.82
AXtMarwanBold	109	46	385	18	11.95	13	2002	0.65
AXtMarwanLight	3261	828	18963	405	4.37	79	83091	0.10
AXtShareQ	5	10	64	0	15.62	7	299	2.34
AXtShareQXL	68	35	371	13	9.43	10	1973	0.51
AXtThuluthMubassat	1	0	3	0	0.00	0	13	0.00
Total (Eval a)	3480	955	20042	443	4.76	136	88456	0.15

Table 11 Results for ML trained GMMs using rendered and scanned data of the RAMP-N subset Eval a.

Layout Analysis	Rendered		Scanned	
	WER	CER	WER	CER
Supervised	4.76	0.15	5.79	0.64
OCRopus	-	-	17.62	3.79

the WER. Interesting for future work will therefore remain larger lexica or character and PAW language models to further reduce the effect OOVs. Due to unbalanced font frequencies a re-rendering of the training data in other fonts might further reduce the error rates in future works.

The results in Table 11 show the difference between rendered and scanned results, where we additionally compared supervised layout and unsupervised layout analysis using OCRopus[11]. The scans were generated by printing and scanning the PDFs in their original size, i.e. DIN-A2 at 600dpi. It can be seen that the main performance decrease is due to OCRopus' layout analysis problems and not due to the scan quality.

As it is often observed that discriminative GHMM training performs better with fewer Gaussian mixture densities, we use a split-6 ML trained model to initialize our M-MPE training (cf. Λ_0 in Section 3.3). The results in Figure 16 show again a significant reduction in terms of WER and CER. Note that BBN's Glyph HMM system PLATO [52] reported similar relative improvements for position dependent glyph models and discriminative MMI/MPE training.

In Figure 17 an unsupervised alignment example is shown for a line segment of RAMP-N subset Eval a, which seems suitable for postprocessing steps such as syntax highlighting or reCAPTCHA-like [77] processes. We used an ML trained GHMM model resulting in zero word/character errors.

[11] `http://code.google.com/p/ocropus/`

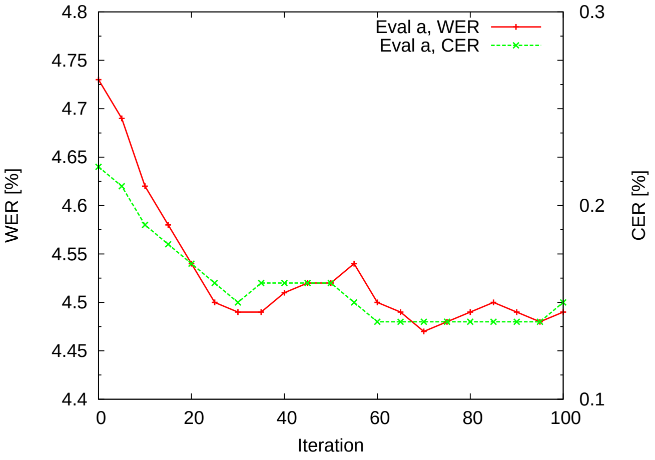

Fig. 16 Results for M-MPE training on RAMP-N corpus Eval a

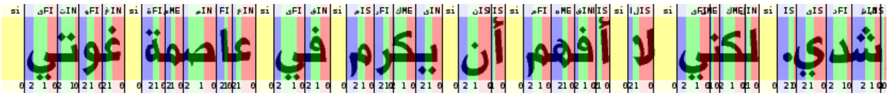

Fig. 17 Example of an unsupervised alignment on RAMP-N corpus Eval a

7 Conclusions

We presented our hidden Markov model (HMM) based RWTH OCR system which represents a unique framework for large vocabulary optical character recognition (OCR). The advantages of confidence- and margin-based discriminative training using a MMI/MPE training criterion for model adaptation using an HMM based multi-pass decoding system were shown for Arabic handwriting on the IfN/ENIT corpus (isolated word recognition), and preliminary results were shown for Arabic machine-printed text on the RAMP-N corpus (open-vocabulary, continuous line recognition). More details are presented in r [13].

We discussed an approach how to modify existing training criteria for handwriting recognition like for example MMI and MPE to include a margin term. The modified training criterion M-MMI was shown to be closely related to existing large margin classifiers (e.g. SVMs) with the respective loss function. This approach allows for the direct evaluation of the utility of the margin term for handwriting recognition. As expected, the benefit from the additional margin term clearly depends on the training conditions. The proposed discriminative training approach could outperform the ML trained systems on all tasks.

The impact of different writing styles was dealt with a novel confidence-based discriminative training for model adaptation, where the usage of state-confidences during the iterative optimization process based on the modified M-MMI-conf criterion could decrease the word-error-rate on the IfN/ENIT database by 33% relative in comparison to an ML trained system.

Interesting for further research will remain hybrid HMM/ANN approaches [27, 21], combining the advantages of large and non-linear context modeling via neural networks while profiting from the Markovian sequence modeling. This is also supported by the 36% relative improvement we could achieve in the ICFHR 2010 Arabic handwriting competition [50] with the proposed discriminative GHMM framework but an MLP based feature extraction.

We proposed an approach to automatically generate large corpora for machine-printed text recognition. The preliminary results on the novel RAMP-N database showed that our framework is able to recognize Arabic handwritten *and* machine-printed texts. Future work will focus on using more visual training data, larger lexica, higher order n-gram language models, and character or PAW based language models as e.g. successfully used in [52].

Acknowledgements. We would like to thank Christian Plahl, Stefan Hahn, Simon Wiesler, Patrick Doetsch, Robert Pyttel, Stephan Jonas, Jens Forster, Christian Gollan, and Thomas Deselaers for their support.

This work was partly realized as part of the Google Research Award "Robust Recognition of Machine Printed Text", and as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

References

1. NIST 2010 open handwriting recognition and translation evaluation plan, version 2.8, February 2010.
2. Ashraf AbdelRaouf, Colin Higgins, Tony Pridmore, and Mahmoud Khalil. Building a multi-modal Arabic corpus (MMAC). *International Journal on Document Analysis and Recognition (IJDAR)*, 13:285–302, 2010. 10.1007/s10032-010-0128-2.
3. Haikal El Abed and Volker Märgner. ICDAR 2009 – Arabic handwriting recognition competition. *International Journal on Document Analysis and Recognition (IJDAR)*, 1:1433–2833, 4 2010.
4. Amin G. Al-Hashim and Sabri A. Mahmoud. Printed arabic text database (patdb) for research and benchmarking. In *Proceedings of the 9th WSEAS international conference on Applications of computer engineering*, ACE'10, pages 62–68, Stevens Point, Wisconsin, USA, 2010. World Scientific and Engineering Academy and Society (WSEAS).
5. T. Anastasakos and S.V. Balakrishnan. The use of confidence measures in unsupervised adaptation of speech recognizers. In *International Conference on Spoken Language Processing (ICSLP)*, Sydney, Australia, 1998.
6. I. Bazzi, R. Schwartz, and J. Makhoul. An omnifont open-vocabulary OCR system for English and Arabic. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(6):495–504, 1999.
7. R. Bertolami and H. Bunke. Hidden markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, 41(11):3452–3460, November 2008.
8. Alain Biem. Minimum classification error training for online handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(7):1041–1051, 2006.
9. H. Bourland and N. Morgan. Connectionist speech recognition: A hybrid approach. *Series in engineering and computer science. Kluwer Academic Publishers*, 247, 1994.

10. Scott Shaobing Chen and P. S. Gopalakrishnan. Speaker, environment and channel change detection and clustering via the Bayesian information criterion. In *DARPA Broadcast News Transcription and Understanding Workshop*, pages 127–132, Lansdowne, Virginia, USA, February 1998.
11. R. Davidson and R. Hopely. Arabic and persian ocr training and test data sets. In *Symp. on Document Image Understanding Technology*, 30 April - 2 May 1997.
12. Trinh-Minh-Tri Do and Thierry Artières. Maximum margin training of gaussian hmms for handwritring recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 976–980, Barcelona, Spain, July 2009.
13. Philippe Dreuw. *Probabilistic Sequence Models for Image Sequence Processing and Recognition*. PhD thesis, RWTH Aachen University, Aachen, Germany, December 2011.
14. Philippe Dreuw, Patrick Doetsch, Christian Plahl, and Hermann 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, September 2011.
15. Philippe Dreuw, Georg Heigold, and Hermann Ney. Confidence-based discriminative training for model adaptation in offline Arabic handwriting recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 596–600, Barcelona, Spain, July 2009.
16. Philippe Dreuw, Georg Heigold, and Hermann Ney. Confidence and margin-based MMI/MPE discriminative training for offline handwriting recognition. *International Journal on Document Analysis and Recognition (IJDAR)*, 1:submitted for review, 2010.
17. Philippe Dreuw, Stephan Jonas, and Hermann Ney. White-space models for offline Arabic handwriting recognition. In *International Conference on Pattern Recognition (ICPR)*, Tampa, Florida, USA, December 2008.
18. Philippe Dreuw, David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. Speech recognition techniques for a sign language recognition system. In *Interspeech*, pages 2513–2516, Antwerp, Belgium, August 2007.
19. Philippe Dreuw, David Rybach, Christian Gollan, and Hermann Ney. Writer adaptive training and writing variant model refinement for offline Arabic handwriting recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, Barcelona, Spain, July 2009.
20. Haikal El Abed and Volker Märgner. Improvement of Arabic handwriting recognition systems: combination and/or reject? In *Document Recognition and Retrieval XVI*, volume 7247 of *SPIE*, San Jose, CA, USA, January 2009.
21. S. Espana-Boquera, M. Castro-Bleda, J. Gorbe-Moya, and F. Zamora-Martinez. Improving offline handwritten text recognition with hybrid HMM/ANN models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):pre–print, 2010.
22. Gernot A. Fink and Thomas Plötz. Unsupervised estimation of writing style models for improved unconstrained off-line handwriting recognition. In *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, La Baule, France, October 2006.
23. M. J. F. Gales. Maximum likelihood linear transformations for HMM-based speech recognition. *Computer Speech and Language*, 12(2):75–98, April 1998.
24. Adrià Giménez Pastor, Ihab Khoury, and Alfons Juan. Windowed bernoulli mixture hmms for arabic handwritten word recognition. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Kalkota, India, November 2010.
25. Christian Gollan and Michiel Bacchiani. Confidence scores for acoustic model adaptation. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 4289–4292, Las Vegas, NV, USA, April 2008.
26. Christian Gollan and Hermann Ney. Towards automatic learning in LVCSR: Rapid development of a Persian broadcast transcription system. In *Interspeech*, pages 1441–1444, Brisbane, Australia, September 2008.
27. A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):855–868, May 2009.

28. G. Heigold, S. Wiesler, M. Nussbaum, P. Lehnen, R. Schlüter, and H. Ney. Discriminative HMMs, log-linear models, and CRFs: What is the difference? In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5546–5549, Dallas, Texas, USA, March 2010.
29. Georg Heigold. *A Log-Linear Discriminative Modeling Framework for Speech Recognition*. PhD thesis, RWTH Aachen University, Aachen, Germany, June 2010.
30. Georg Heigold, Thomas Deselaers, Ralf Schlüter, and Hermann Ney. Modified MMI/MPE: A direct evaluation of the margin in speech recognition. In *International Conference on Machine Learning (ICML)*, pages 384–391, Helsinki, Finland, July 2008.
31. Georg Heigold, Philippe Dreuw, Stefan Hahn, Ralf Schlüter, and Hermann Ney. Margin-based discriminative training for string recognition. *IEEE Journal of Selected Topics in Signal Processsing - Statistical Learning Methods for Speech and Language Processing*, 4(6):to appear, December 2010.
32. Georg Heigold, Ralf Schlüter, and Hermann Ney. Modified MPE/MMI in a transducer-based framework. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3749–3752, Taipei, Taiwan, April 2009.
33. H. Hermansky and S. Sharma. Traps - classifiers of temporal patterns. In *International Conference on Spoken Language Processing (ICSLP)*, 1998.
34. Charles Jacobs, Patrice Y. Simard, Paul Viola, and James Rinker. Text recognition of low-resolution document images. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 695–699, 2005.
35. T. Jebara. *Discriminative, generative, and imitative learning*. PhD thesis, Massachusetts Institute of Technology, 2002.
36. A. Juan, A. H. Toselli, J. Domnech, J. Gonzlez, I. Salvador, E. Vidal, and F. Casacuberta. Integrated handwriting recognition and interpretation via finite-state models. *International Journal of Pattern Recognition and Artifial Intelligence (IJPRAI)*, 2004:519–539, 2001.
37. Andrew Kae and Erik Learned-Miller. Learning on the fly: Font-free approaches to difficult OCR problems. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 571–575, Barcelona, Spain, July 2009.
38. S. Kanthak, K. Schütz, and H. Ney. Using SIMD instructions for fast likelihood calculation in LVCSR. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1531–1534, Istanbul, Turkey, June 2000.
39. T. Kemp and T. Schaaf. Estimating confidence using word lattices. In *European Conference on Speech Communication and Technology*, Rhodes, Greece, 1997.
40. C. J. Leggetter and P. C. Woodland. Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models. *Computer Speech and Language*, 9(2):171 – 185, April 1995.
41. C.J. Leggetter and P.C. Woodland. Flexible speaker adaptation using maximum likelihood linear regression. In *ARPA Spoken Language Technology Workshop*, pages 104 – 109, Austin, TX, USA, January 1995.
42. J. Lööf, R. Schlüter, and H. Ney. Efficient estimation of speaker-specific projecting feature transforms. In *International Conference on Spoken Language Processing (ICSLP)*, pages 1557 – 1560, Antwerp, Belgium, August 2007.
43. Jonas Lööf, Christian Gollan, Stefan Hahn, Georg Heigold, Björn Hoffmeister, Christian Plahl, David Rybach, Ralf Schlüter, and Hermann Ney. The RWTH 2007 TC-STAR evaluation system for european English and Spanish. In *Interspeech*, pages 2145–2148, Antwerp, Belgium, August 2007.
44. Liana M. Lorigo and Venu Govindaraju. Offline Arabic handwriting recognition: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(85):712–724, May 2006.
45. Yi Lu. Machine printed character segmentation –; an overview. *Pattern Recognition*, 28(1):67–80, 1995.
46. Zhidong A. Lu, Issam Bazzi, Andras Kornai, John Makhoul, Premkumar S. Natarajan, and Richard Schwartz. A robust language-independent OCR system. In *AIPR Workshop: Advances in Computer-Assisted Recognition*, volume 3584 of *SPIE*, pages 96–104, 1998.

47. V. Märgner and H. El Abed. ICDAR 2007 Arabic handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, volume 2, pages 1274–1278, September 2007.
48. V. Märgner and H. El Abed. ICDAR 2009 Arabic handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 1383–1387, Barcelona, Spain, July 2009.
49. V. Märgner, M. Pechwitz, and H.E. Abed. ICDAR 2005 Arabic handwriting recognition competition. In *International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 70–74, Seoul, Korea, August 2005.
50. Volker Märgner and Haikal El Abed. ICFHR 2010 Arabic handwriting recognition competition. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, Kalkota, India, November 2010.
51. P. Natarajan, S. Saleem, R. Prasad, E. MacRostie, and K. Subramanian. *Arabic and Chinese Handwriting Recognition*, volume 4768/2008 of *LNCS*, chapter Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach, pages 231–250. Springer Berlin / Heidelberg, 2008.
52. Prem Natarajan. Portable language-independent adaptive translation from OCR, final report (phase 1). Technical report, BBN Technologies, June 2009.
53. Hermann Ney and Stefan Ortmanns. Progress in dynamic programming search for LVCSR. *Proceedings of the IEEE*, 88(8):1224–1240, August 2000.
54. R. Nopsuwanchai and D. Povey. Discriminative training for HMM-based offline handwritten character recognition. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 114–118, 2003.
55. Roongroj Nopsuwanchai, Alain Biem, and William F. Clocksin. Maximization of mutual information for offline Thai handwriting recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(8):1347–1351, 2006.
56. Markus Nußbaum-Thom, Simon Wiesler, Martin Sundermeyer, Christian Plahl, Stefan Hahn, Ralf Schlüter, and Hermann Ney. The RWTH 2009 Quaero ASR evaluation system for English and German. In *Interspeech*, Makuhari, Japan, September 2010.
57. J. Olive. Multilingual automatic document classification analysis and translation (MADCAT). Proposer Information Pamplet SOL BAA 07-38, DARPA/IPTO, 2007.
58. S. Ortmanns, H. Ney, and X. Aubert. A word graph algorithm for large vocabulary continuous speech recognition. *Computer Speech and Language*, 11(1):43–72, January 1997.
59. Stefan Ortmanns and Hermann Ney. Look-ahead techniques for fast beam search. *Computer Speech and Language*, 14(1):15–32, January 2000.
60. M. Padmanabhan, G. Saon, and G. Zweig. Lattice-based unsupervised MLLR for speaker adaptation. In *ISCA ITRW Automatic Speech Recognition: Challenges for the Millenium*, Paris, France, 2000.
61. M. Pechwitz, S. Snoussi Maddouri, V. Mägner, N. Ellouze, and H. Amiri. IFN/ENIT-database of handwritten Arabic words. In *Colloque International Francophone sur l'Ecrit et le Document (CIFED)*, Hammamet, Tunis, October 2002.
62. M. Pitz, F. Wessel, and H. Ney. Improved MLLR speaker adaptation using confidence measures for conversational speech recognition. In *International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, 2000.
63. Christian Plahl, Björn Hoffmeister, Mei-Yuh Hwang, Danju Lu, Georg Heigold, Jonas Lööf, Ralf Schlüter, and Hermann Ney. Recent improvements of the RWTH GALE Mandarin LVCSR system. In *Interspeech*, pages 2426–2429, Brisbane, Australia, September 2008.
64. D. Povey. *Discriminative Training for Large Vocabulary Speech Recognition*. PhD thesis, Cambridge, England, 2004.
65. D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah. Boosted MMI for model and feature-space discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Las Vegas, NV, USA, April 2008.

66. D. Povey and P. C. Woodland. Minimum phone error and I-smoothing for improved discriminative training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 1, Orlando, FL, USA, 2002.
67. R. Prasad, S. Saleem, M. Kamali, R. Meermeier, and P. Natarajan. Improvements in hidden markov model based Arabic OCR. In *International Conference on Pattern Recognition (ICPR)*, Tampa, FL, USA, December 2008.
68. David Rybach, Christian Gollan, Georg Heigold, Björn Hoffmeister, Jonas Lööf, Ralf Schlüter, and Hermann Ney. The RWTH Aachen university open source speech recognition system. In *Interspeech*, pages 2111–2114, Brighton, U.K., September 2009.
69. David Rybach, Stefan Hahn, Christian Gollan, Ralf Schlüter, and Hermann Ney. Advances in Arabic broadcast news transcription at RWTH. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 449–454, Kyoto, Japan, December 2007.
70. M.-P. Schambach, J. Rottland, and T. Alary. How to convert a latin handwriting recognition system to arabic. In *International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2008.
71. Joachim Schenk and Gerhard Rigoll. Novel Hybrid NN/HMM Modelling Techniques for On-line Handwriting Recognition. In *International Workshop on Frontiers in Handwriting Recognition (IWFHR)*, La Baule, France, October 2006.
72. R. Schlüter, B. Müller, F. Wessel, and H. Ney. Interdependence of language models and discriminative training. In *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, volume 1, pages 119–122, Keystone, CO, December 1999.
73. Ralf Schlüter. *Investigations on Discriminative Training Criteria*. PhD thesis, RWTH Aachen University, Aachen, Germany, September 2000.
74. F. Slimane, R. Ingold, S. Kanoun, M. A. Alimi, and J. Hennebert. A new Arabic printed text image database and evaluation protocols. In *International Conference on Document Analysis and Recognition (ICDAR)*, pages 946–950, Barcelona, Spain, July 2009.
75. Andreas Stolcke. SRILM - an extensible language modeling toolkit. In *International Conference on Spoken Language Processing (ICSLP)*, Denver, CA, USA, September 2002.
76. Fabio Valente, Jithendra Vepa, Christian Plahl, Christian Gollan, Hynek Hermansky, and Ralf Schlüter. Hierarchical neural networks feature extraction for LVCSR system. In *Interspeech*, pages 42–45, Antwerp, Belgium, August 2007.
77. Luis von Ahn, Benjamin Maurer, Colin McMillen, David Abraham, and Manuel Blum. recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468, September 2008.
78. Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying A. Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, Valtcho Valtchev, and Phil Woodland. *The HTK Book (for HTK Version 3.4)*. Cambridge University Engineering Department, 2006.
79. J. Zhang, R. Jin, Y. Yang, and A.G. Hauptmann. Modified logistic regression: An approximation to SVM and its applications in large-scale text categorization. In *International Conference on Machine Learning (ICML)*, August 2003.