# The RWTH-OCR Handwriting Recognition System for Arabic Handwriting

Philippe Dreuw, Georg Heigold, David Rybach,
Christian Gollan, and Hermann Ney
dreuw@cs.rwth-aachen.de

DAAD Workshop, Sousse, Tunisia – March 2010

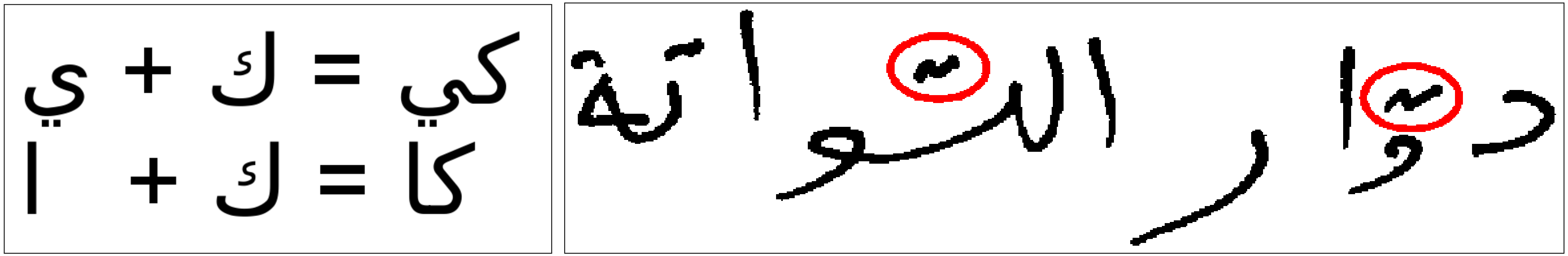
Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6
Computer Science Department
RWTH Aachen University, Germany

# Outline

1. Introduction
2. Adaptation of the RWTH-ASR framework for Handwriting Recognition
    - ► System Overview
    - ► Discriminative training using modified MMI criterion
    - ► Unsupervised confidence-based discriminative training during decoding
    - ► Writer Adaptive Training
3. Experimental Results
4. Summary

# Introduction

- **Arabic handwriting system**
  - **right-to-left, 28 characters, position-dependent character writing variants**
  - **ligatures and diacritics**
  - **Pieces of Arabic Word (PAWs) as subwords**



(a) Ligatures

(b) Diacritics

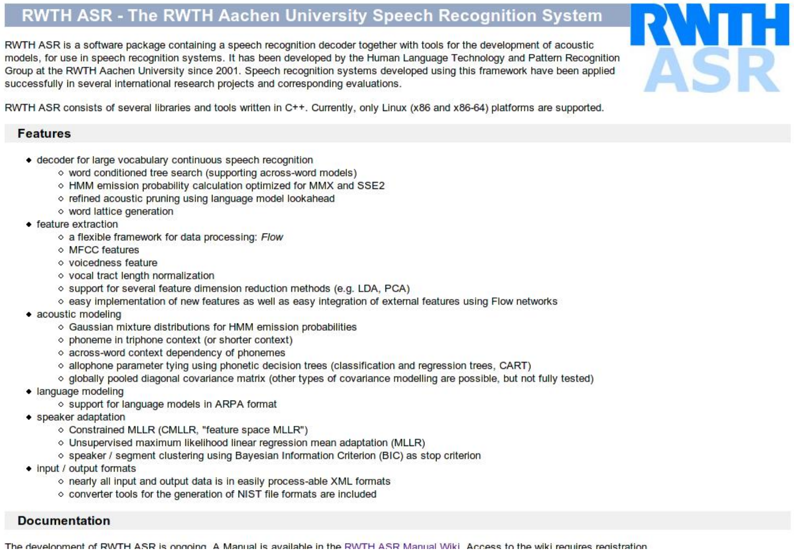
- **state-of-the-art**
  - **preprocessing (normalization, baseline estimation, etc.) + HMMs**
- **our approach:**
  - **adaptation of RWTH-ASR framework for handwriting recognition**
  - **preprocessing-free feature extraction, focus on modeling**

# RWTH ASR System: Overview

**The RWTH Aachen University Open Source Speech Recognition System [Rybach & Gollan$^{+}$ 09]**

http://www-i6.informatik.rwth-aachen.de/rwth-asr/

- **speech recognition framework supporting:**
  - ▷ **acoustic training including speaker adaptive training**
  - ▷ **speaker normalization / adaptation: VTLN, CMLLR, MLLR**
  - ▷ **multi-pass decoding**
- **framework also used for machine translation, video / image processing**



## RWTH ASR - The RWTH Aachen University Speech Recognition System

RWTH ASR is a software package containing a speech recognition decoder together with tools for the development of acoustic models, for use in speech recognition systems. It has been developed by the Human Language Technology and Pattern Recognition Group at the RWTH Aachen University since 2001. Speech recognition systems developed using this framework have been applied successfully in several international research projects and corresponding evaluations.

RWTH ASR consists of several libraries and tools written in C++. Currently, only Linux (x86 and x86-64) platforms are supported.

### Features

- decoder for large vocabulary continuous speech recognition
  - word conditioned tree search (supporting across-word models)
  - HMM emission probability calculation optimized for MMX and SSE2
  - refined acoustic pruning using language model lookahead
  - word lattice generation
- feature extraction
  - a flexible framework for data processing: *Flow*
  - MFCC features
  - voicedness feature
  - vocal tract length normalization
  - support for several feature dimension reduction methods (e.g. LDA, PCA)
  - easy implementation of new features as well as easy integration of external features using Flow networks
- acoustic modeling
  - Gaussian mixture distributions for HMM emission probabilities
  - phoneme in triphone context (or shorter context)
  - across-word context dependency of phonemes
  - allophone parameter tying using phonetic decision trees (classification and regression trees, CART)
  - globally pooled diagonal covariance matrix (other types of covariance modelling are possible, but not fully tested)
- language modeling
  - support for language models in ARPA format
- speaker adaptation
  - Constrained MLLR (CMLLR, "feature space MLLR")
  - Unsupervised maximum likelihood linear regression mean adaptation (MLLR)
  - speaker / segment clustering using Bayesian Information Criterion (BIC) as stop criterion
- input / output formats
  - nearly all input and output data is in easily process-able XML formats
  - converter tools for the generation of NIST file formats are included

### Documentation

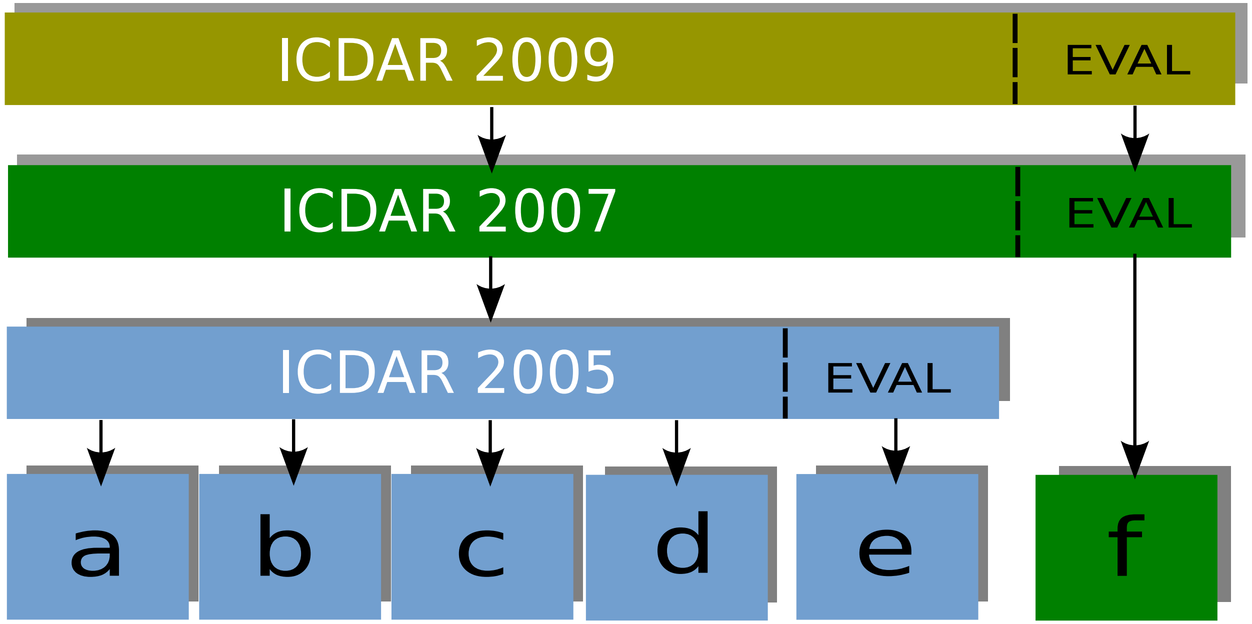
The development of RWTH ASR is ongoing. A Manual is available in the RWTH ASR Manual Wiki. Access to the wiki requires registration

- **published under an open source licence (RWTH ASR Licence)**
- **commercial licences available on request**
- **more than 100 registrations until today**

# Arabic Handwriting - IFN/ENIT Database

## Corpus development

- ICDAR 2005 Competition: a, b, c, d sets for training, evaluation on set e
- ICDAR 2007 Competition: ICDAR05 + e sets for training, evaluation on set f
- ICDAR 2009 Competition: ICDAR 2007 for training, evaluation on set f

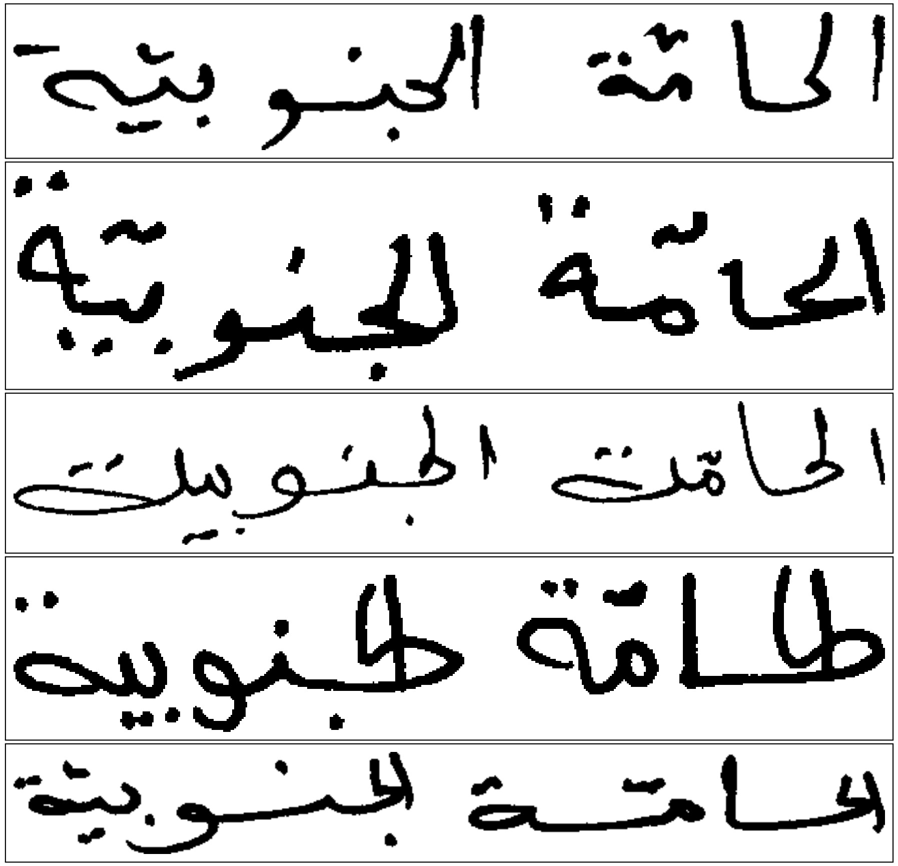


# Arabic Handwriting - IFN/ENIT Database

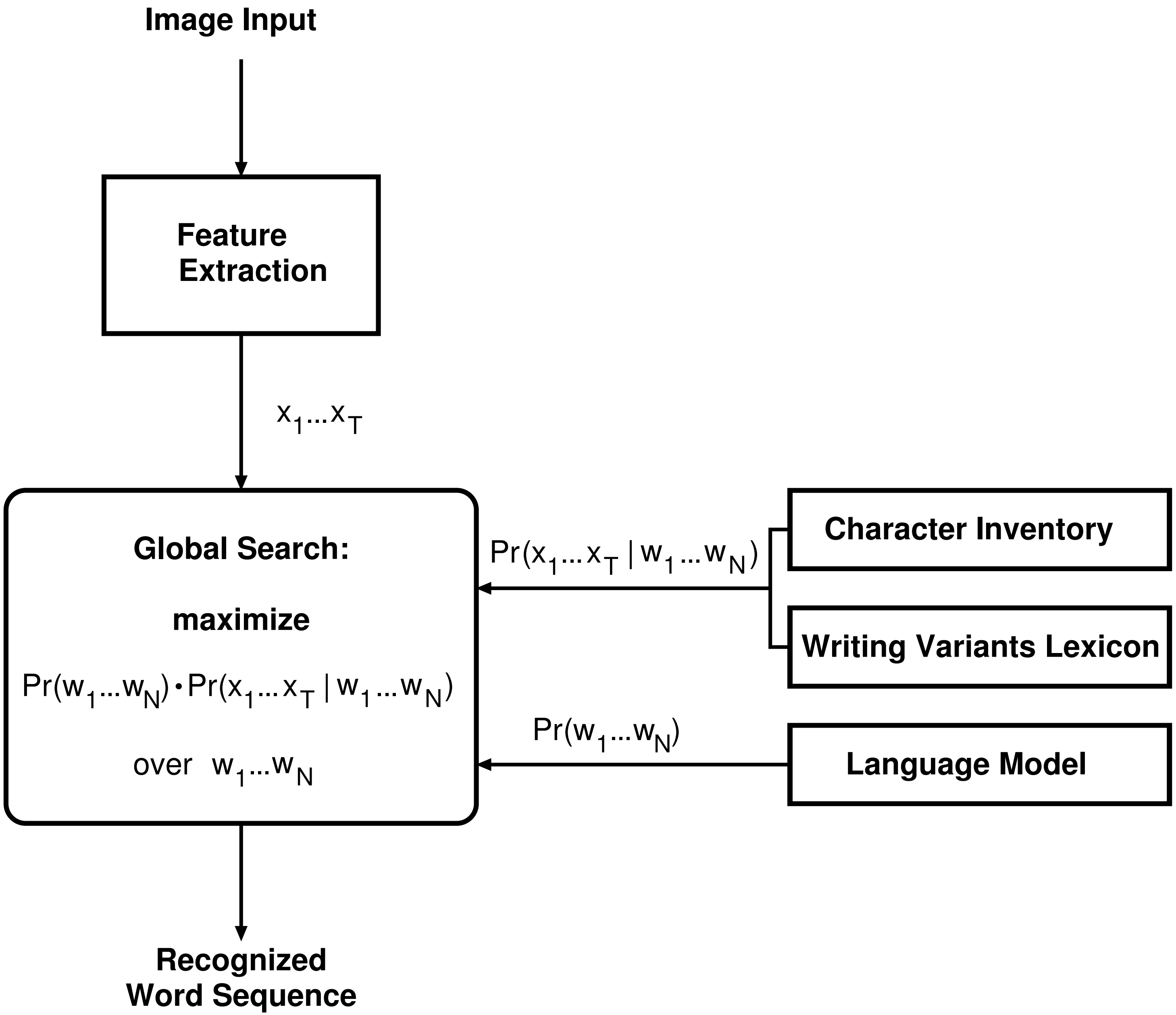
- 937 classes
- 32492 handwritten Arabic words (Tunisian city names)
- database is used by more than 60 groups all over the world
- writer statistics

| set | #writers | #samples |
|---|---|---|
| a | 102 | 6537 |
| b | 102 | 6710 |
| c | 103 | 6477 |
| d | 104 | 6735 |
| e | 505 | 6033 |
| Total | 916 | 32492 |

- examples (same word):



# System Overview



# Writing Variant Model Refinement

- HMM baseline system
  - searching for an unknown word sequence $w_1^N := w_1, \ldots, w_N$
  - unknown number of words $N$
  - maximize the posterior probability $p(w_1^N | x_1^T)$
  - described by Bayes' decision rule:

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

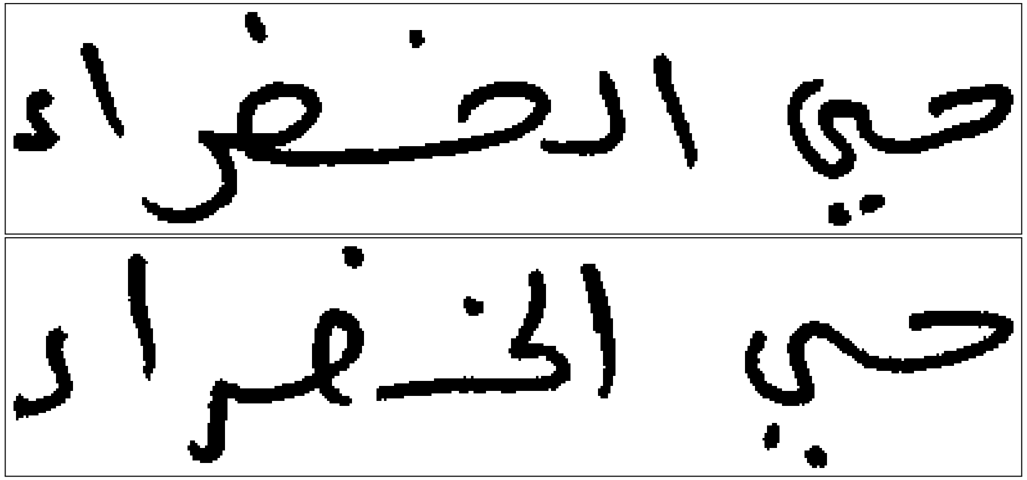
with $\gamma$ a scaling exponent of the language model.

# Writing Variant Model Refinement

- ligatures and diacritics in Arabic handwriting
  - ▷ same Arabic word can be written in several writing variants

  → depends on writer's handwriting style

- Example: *laB khM* vs. *khMlaB*



- lexicon with multiple writing variants [Details]
  - ▷ problem: many and rare writing variants

# Writing Variant Model Refinement

- probability $p(v|w)$ for a variant $v$ of a word $w$
  - ▷ usually considered as equally distributed
  - ▷ here: we use the count statistics as probability:

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

- writing variant model refinement:

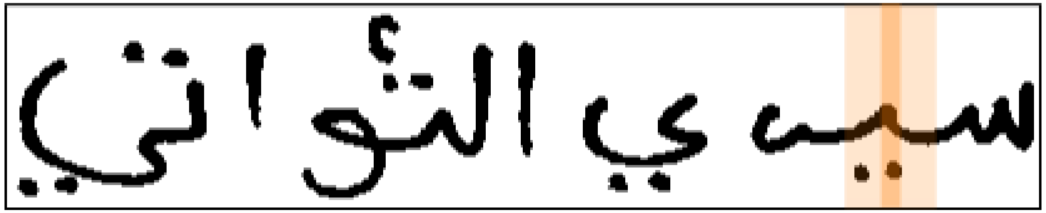
$$p(x_1^T|w_1^N) \approx \max_{v_1^N|w_1^N} \left\{ p^\alpha(v_1^N|w_1^N) p(x_1^T|v_1^N, w_1^N) \right\}$$

with $v_1^N$ a sequence of unknown writing variants
$\alpha$ a scaling exponent of the writing variant probability

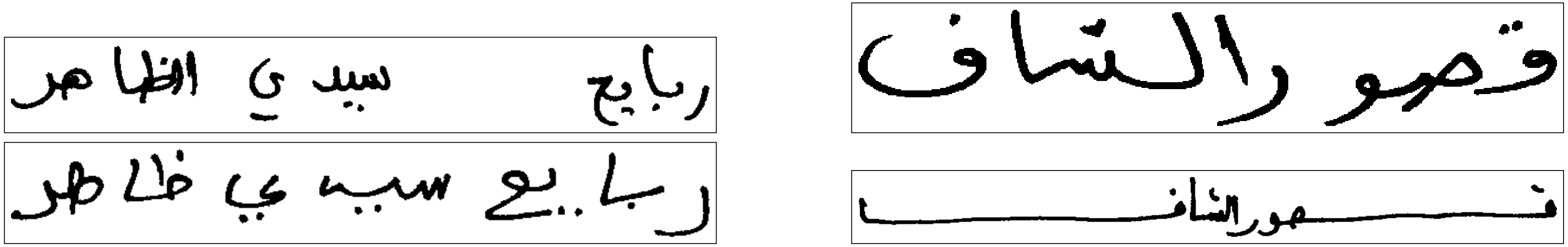
- training: corpus and lexicon with supervised writing variants possible!

# Visual Modeling: Feature Extraction and HMM Transitions

- recognition of characters within a context, temporal alignment necessary
- features: sliding window, no preprocessing, PCA reduction



- important: HMM whitespace models (a) and state-transition penalties (b)

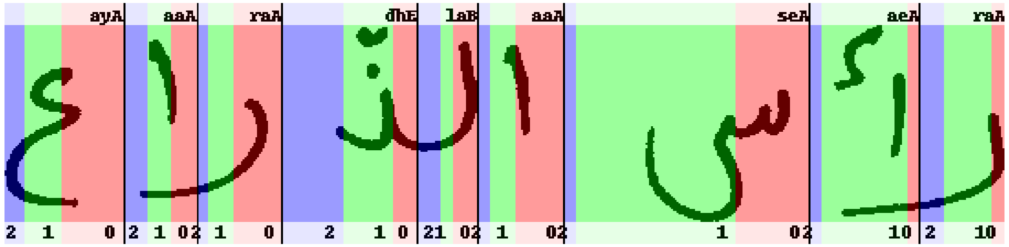


(a)

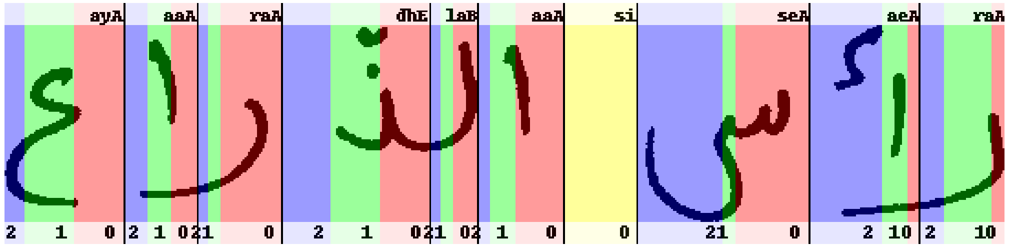
(b)

# Visual Modeling: Writing Variants Lexicon

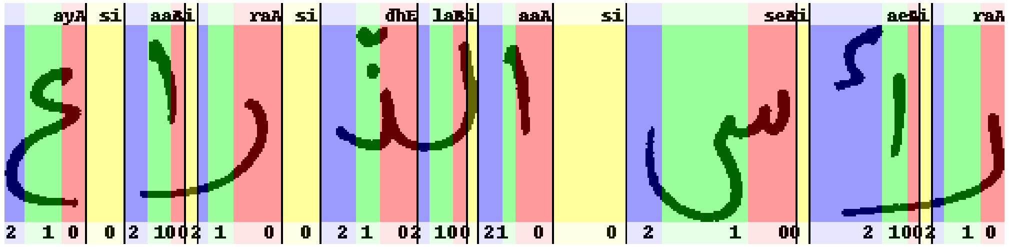
- ▶ most reported error rates are dependent on the number of PAWs
- ▶ without separate whitespace model



- ▶ always whitespaces between compound words



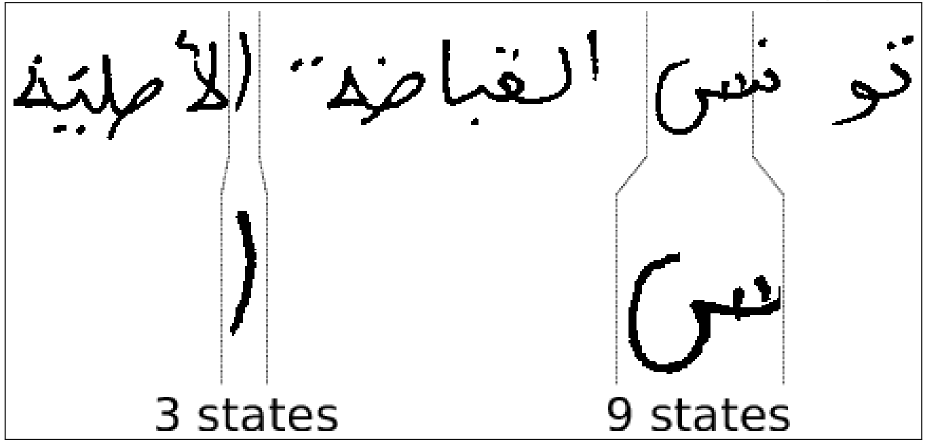
- ▶ whitespaces as writing variants between and within words



White-Space Models for Pieces of Arabic Words [Dreuw & Jonas$^{+}$ 08] in ICPR 2008

# Visual Modeling: Model Length Estimation

- more complex characters should be represented by more HMM states



- the number of states $S_c$ for each character $c$ is updated by

$$S_c = \frac{N_{x,c}}{N_c} \cdot \alpha$$

with

$S_c$ = estimated number states for character $c$

$N_{x,c}$ = number of observations aligned to character $c$

$N_c$ = character count of $c$ seen in training

$\alpha$ = character length scaling factor.

[Visualization]

# RWTH-OCR Training and Decoding Architectures

- Training
  - Maximum Likelihood (ML)
  - CMLLR-based Writer Adaptive Training (WAT)
  - discriminative training using modified-MMI criterion (M-MMI)

- Decoding
  - 1-pass
    - ML model
    - M-MMI model
  - 2-pass
    - segment clustering for CMLLR writer adaptation
    - unsupervised confidence-based M-MMI training for model adaptation

# Discriminative Training: Modified-MMI Criterion

- training: weighted accumulation of observations $x_t$:

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

1. ML: Maximum Likelihood

$$\omega_{r,s,t} := 1.0$$

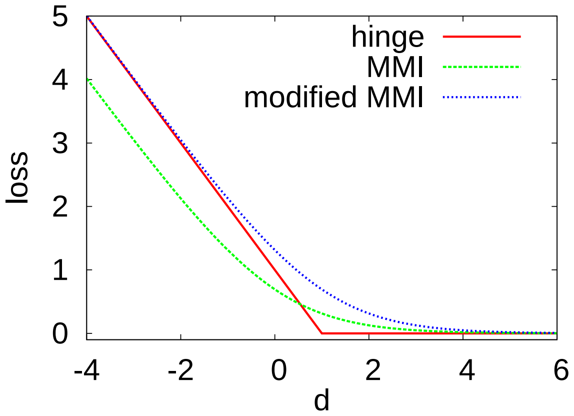
2. MMI: Maximum Mutual Information

$$\omega_{r,s,t} := \frac{\sum\limits_{s_1^{T_r}:s_t=s} p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(W_r)}{\sum\limits_{V} \sum\limits_{s_1^{T_r}:s_t=s} p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(V)}$$

- $\omega_{r,s,t}$ is the "(true) posterior" weight
- iteratively optimized with Rprop

# Discriminative Training: Modified-MMI Criterion

- margin-based training for HMMs
  - similar to SVM training, but simpler/faster within RWTH-OCR framework?
  - M-MMI = differentiable approximation to SVM optimization



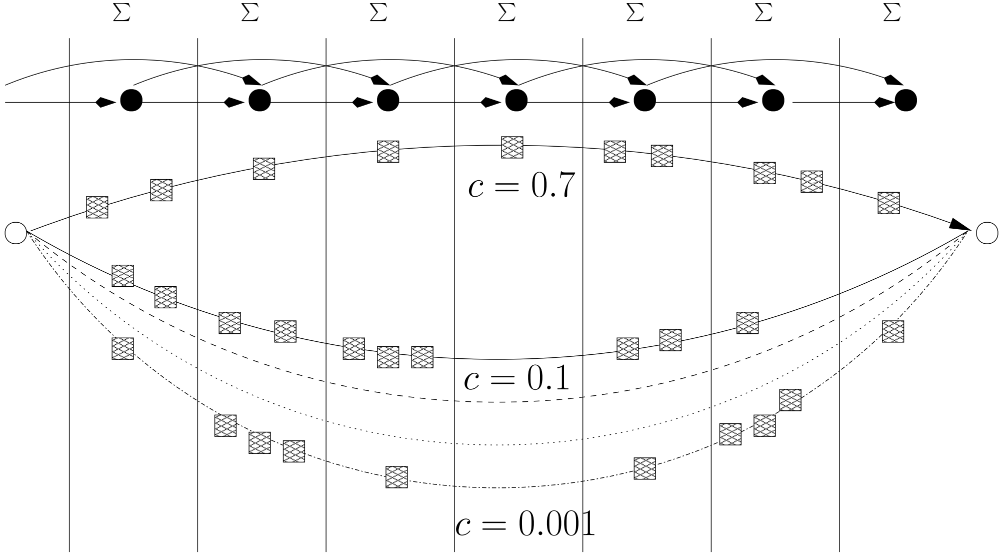
3. M-MMI:

$$\omega_{r,s,t}(\rho \neq 0) := \frac{\sum_{s_1^{T_r}:s_t=s} [p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(W_r) \cdot e^{-\rho\delta(W_r,W_r)}]^\gamma}{\sum_V \sum_{s_1^{T_r}:s_t=s} [p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(V) \cdot e^{-\rho\delta(W_r,V)}]^\gamma}$$

- $\omega_{r,s,t}$ is the “margin posterior” weight
- $e^{-\rho\delta(W_r,W_r)}$ corresponds to the margin offset
- with $\gamma \rightarrow \infty$ equals to the SVM hinge loss function
- iteratively optimized with Rprop

# Decoding: Unsupervised Confidence-Based Discriminative Training

- example for a word-graph and the corresponding 1-best state alignment



- necessary steps for **margin-based model adaptation during decoding**:
  - 1-pass recognition (unsupervised transcriptions and word-graph)
  - calculation of corresponding confidences (sentence, word, or state-level)
  - unsupervised **M-MMI-conf** training on test data to adapt models (w/ regularization)
- can be done iteratively with unsupervised corpus update!

# Decoding: Modified-MMI Criterion And Confidences

## 4. M-MMI-conf:

$$\omega_{r,s,t}(\rho \neq 0) := \frac{\sum_{s_1^{T_r}:s_t=s} p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(W_r) \cdot e^{-\rho\delta(W_r,W_r)}}{\underbrace{\sum_V \sum_{s_1^{T_r}:s_t=s} p(x_1^{T_r}|s_1^{T_r})p(s_1^{T_r})p(V)}_{\text{posterior}} \cdot \underbrace{e^{-\rho\delta(W_r,V)}}_{\text{margin}}} \cdot \underbrace{\delta(c_{r,s,t} > c_{\text{threshold}})}_{\text{confidence}}$$

- weighted accumulation becomes:

$$\text{acc}_s = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \underbrace{\omega_{r,s,t}(\rho)}_{\text{margin posterior}_{\rho\neq 0}} \cdot \underbrace{c_{r,s,t}}_{\text{confidence}} \cdot x_t$$

- confidences at:
  - sentence-, word-, or state-level

# Training Criterions

- **ML training: accumulation of observations $x_t$:**

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

- **M-MMI training: weighted accumulation of observations $x_t$:**

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

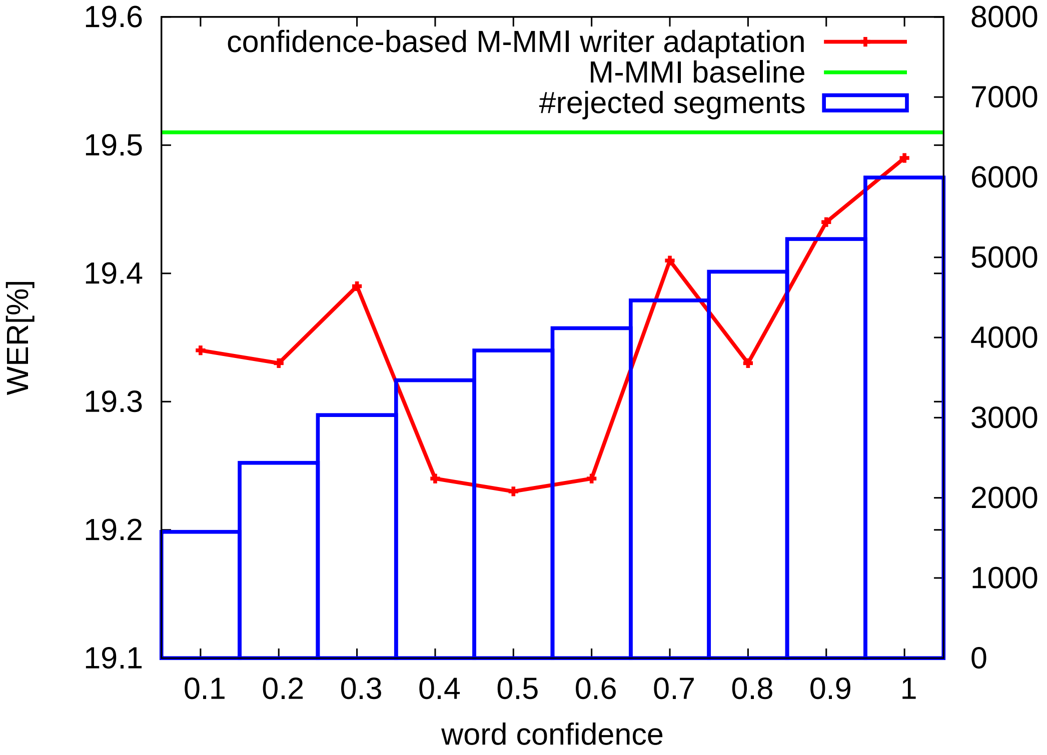
- **M-MMI-conf training: confidence-weighted accumulation of observations $x_t$:**

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

  - **with confidence $c_{r,s,t}$ at sentence-, word, or state-level**

# Results - Unsupervised Model Adaptation: M-MMI-conf

- M-MMI criterion with posterior confidences (M-MMI-conf)
- unsupervised training for model adaptation during decoding
- word-confidence based M-MMI-conf training and rejections



- ▷ confidence threshold $c = 0.5 \rightarrow$ more than 60% segment rejection rate
- ▷ small amount of adaptation data only

# Results - Unsupervised Model Adaptation: M-MMI-conf

- **unsupervised training for model adaptation during decoding**
- **state-confidence based M-MMI-conf training and rejections**
  - ▷ **arc posteriors from the lattice output from the decoder**
  - ▷ **only word frames aligned with a high confidence in 1st pass**
    **→ unsupervised model adaptation**
  - ▷ **only 5% frame rejection rate (20,970 frames of 396,416)**

- **ICDAR 2005 Setup** [Comparison]

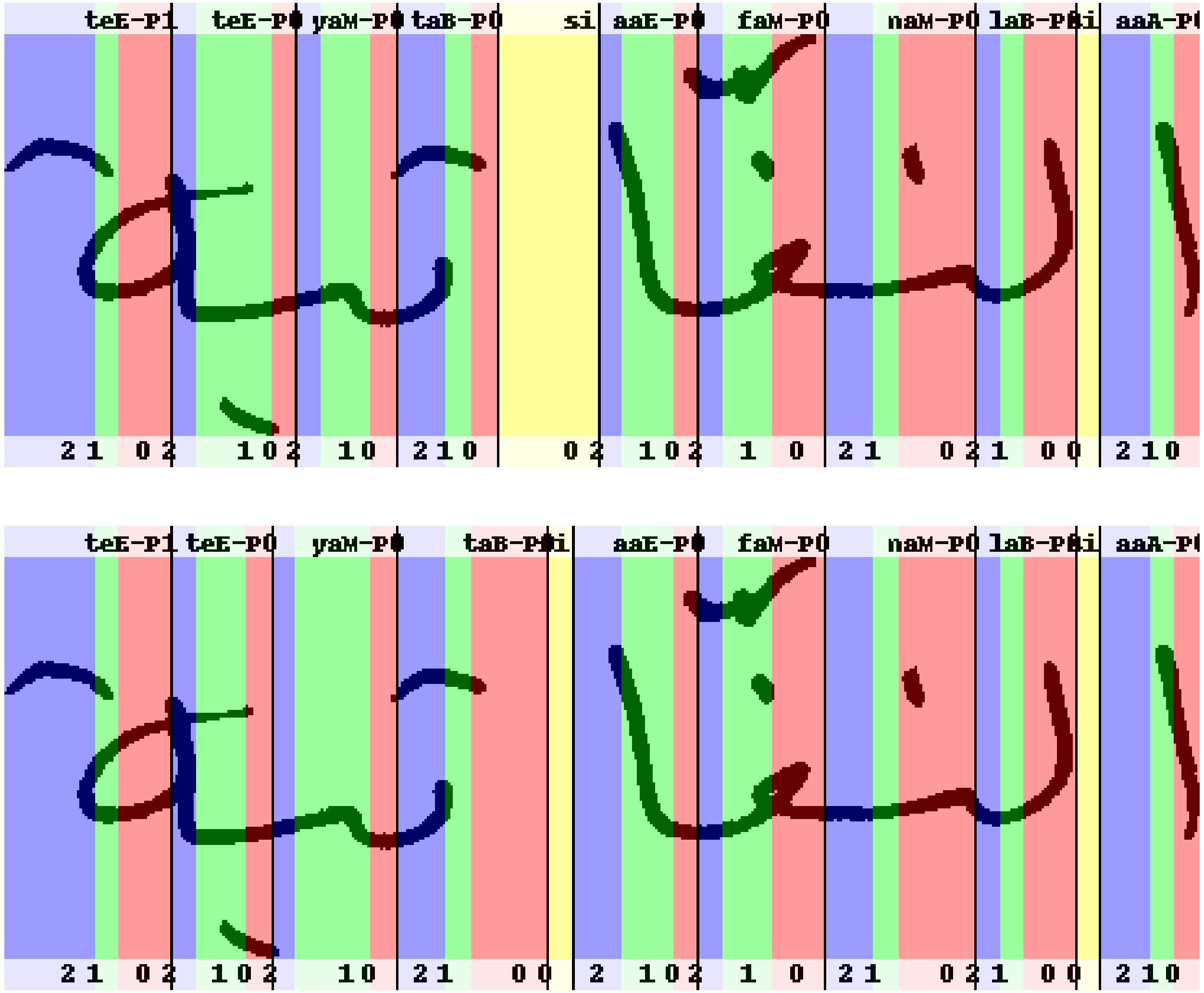
| Training/Adaptation | WER[%] | CER[%] |
|---|---:|---:|
| ML | 21.86 | 8.11 |
| M-MMI | 19.51 | 7.00 |
| + unsupervised adaptation | 20.11 | 7.34 |
| + word-confidences | 19.23 | 7.02 |
| + state-confidences | **17.75** | **6.49** |
| + supervised adaptation | 2.06 | 0.77 |

# Results - Training: ML vs. MMI vs. Modified-MMI Criterion

- **ML = Maximum Likelihood**
- **MLE = Model Length Estimation**
- **MMI vs. modified-MMI after 30 Rprop iterations**
- **ICDAR 2005 Setup** [Comparison]

|  |  | WER [%] |  |  |  |
|---|---|---|---|---|---|
| Train | Test | ML | +MLE | +MMI | +Modified MMI |
| abc | d | 10.88 | 7.80 | 7.44 | **6.12** |
| abd | c | 11.50 | 8.71 | 8.24 | **6.78** |
| acd | b | 10.97 | 7.84 | 7.56 | **6.08** |
| bcd | a | 12.19 | 8.66 | 8.43 | **7.02** |
| abcd | e | 21.86 | 16.82 | 16.44 | **15.35** |

# Visual Inspection of M-MMI Training



# Constrained Maximum Likelihood Linear Regression (CMLLR)

- writer adaptation
  - method for improving visual models in handwriting recognition
  - refine models by adaptation data of particular writers
  - widely used is affine transform based model adaptation

- CMLLR
  - Idea: normalize writing styles by adaptation of the features $x_t$
  - constrained MLLR feature adaptation technique
  - also known as feature space MLLR (fMLLR) [Details]
  - estimate affine feature transform:

$$x'_t = Ax_t + b$$

  - CMLLR is text dependent
    - requires an (automatic) transcription

# Training: CMLLR-based Writer Adaptive Training

- writer adaptation compensates for writer differences during recognition

  → do the same during visual model training

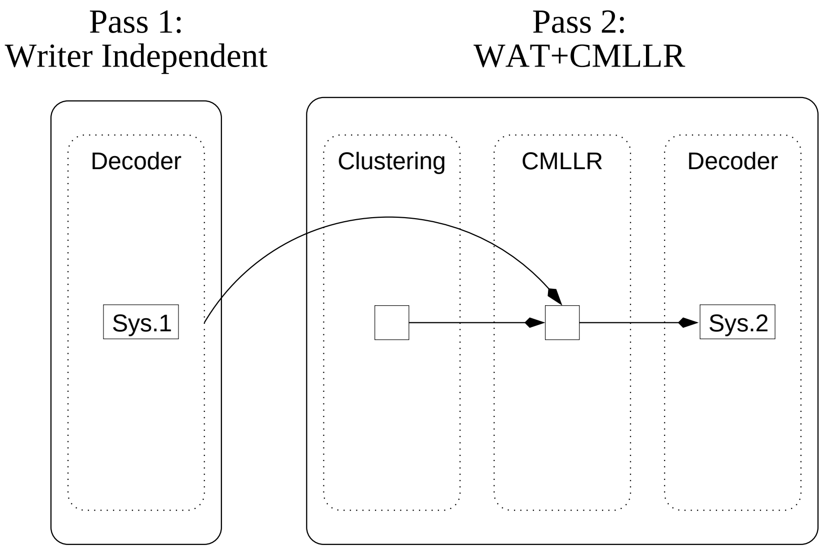
  → maximize the performance gains from writer adaptation

- writer variations are compensated by writer adaptive training (WAT)
- writer normalization using CMLLR
- necessary steps

  1. train writer independent GMMs model
  2. CMLLR transformations are estimated for each (estimated) writer
     - ▷ supervised if writers are known
  3. apply CMLLR transformations on features to train writer dependent GMMs

# Decoding: CMLLR-based Writer Adaptation

- writers and writing styles are unknown
- necessary steps
  1. estimate writing styles using clustering
     - Bayesian Information Criterion (BIC) based stopping condition
  2. estimate CMLLR feature transformations for every estimated writing style cluster
  3. second pass recognition
     - WAT models + CMLLR transformed features



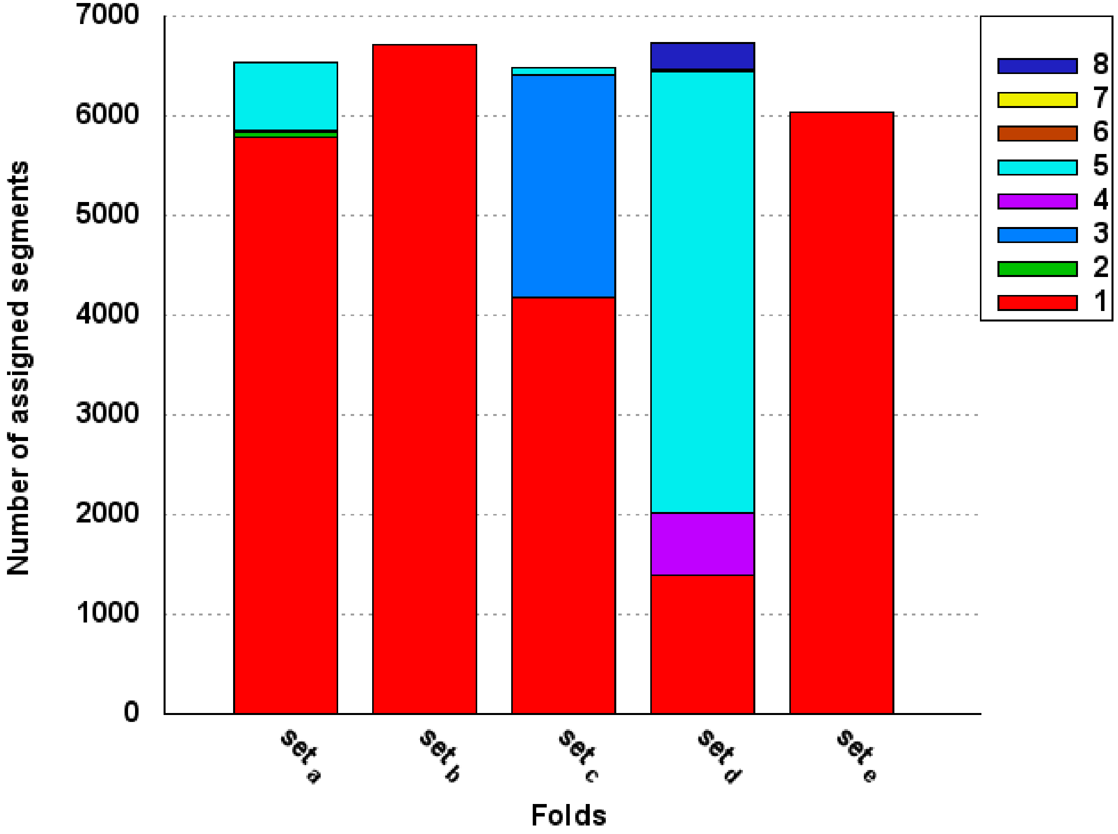
# Results - Decoding: Writer Adaptation

- comparison of MLE, WAT, and CMLLR based feature adaptation
- comparison of unsupervised and supervised writer clustering
  - decoding always unsupervised
  - supervised clustering → only the writer labels are used!

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

# Results - Decoding: Writer Adaptation

- unsupervised clustering: error analysis
  - histograms for segment assignments over the different test folds
  - **problem:** unbalanced segment assignments



# Arabic Handwriting - Experimental Results for IFN/ENIT

- **Writer Adaptive Training + CMLLR for Writer Adaptation**

  **see [Dreuw & Rybach$^{+}$ 09], ICDAR 2009 [Visualization]**

- **M-MMI Training + Unsupervised Confidence-Based Model Adaptation**

  **see [Dreuw & Heigold$^{+}$ 09], ICDAR 2009 [Details]**

- **ICDAR 2005 Setup [Comparison]**

| Train | Test | WER[%] | | | | | |
|---|---|---|---|---|---|---|---|
| | | 1st pass | | | 2nd pass | | |
| | | ML | +MLE | +M-MMI | WAT+CMLLR | | M-MMI-conf |
| | | | | | unsup. | sup. | |
| abc | d | 10.88 | 7.83 | **6.12** | 7.72 | 5.82 | **5.95** |
| abd | c | 11.50 | 8.83 | **6.78** | 9.05 | 5.96 | **6.38** |
| acd | b | 10.97 | 7.81 | **6.08** | 7.99 | 6.04 | **5.84** |
| bcd | a | 12.19 | 8.70 | **7.02** | 8.81 | 6.49 | **6.79** |
| abcd | e | 21.86 | 16.82 | **15.35** | 17.12 | 11.22 | **14.55** |

# Arabic Handwriting - Experimental Results for IFN/ENIT

- **evaluation of RWTH-OCR systems at *Arabic HWR Competition*, ICDAR 2009**
  - **external evaluation at TU Braunschweig, Germany**
  - **set $f$ and set $s$ are unknown (not available)**
  - **unsupervised M-MMI-conf model adaptation achieved similar improvements**
  - **3rd rank (group)**

| ID | WRR[%] | | | | |
|---|---|---|---|---|---|
| | set $f_a$ | set $f_f$ | set $f_g$ | set $f$ | set $s$ |
| RWTH-OCR, ID12 | 86.97 | 88.08 | 87.98 | 85.51 | 71.33 |
| RWTH-OCR, ID13 | 87.17 | 88.63 | 88.68 | 85.69 | 72.54 |
| RWTH-OCR, ID15 | 86.97 | 88.08 | 87.98 | 83.90 | 65.99 |
| A2iA, ID8 | 90.66 | 91.92 | 92.31 | 89.42 | 76.66 |
| MDLSTM, ID11 | 94.68 | 95.65 | 96.02 | 93.37 | 81.06 |

- **Note:**
  - **focus on modeling (ID12 and ID13) and speed (ID15) - no preprocessing**

# Summary

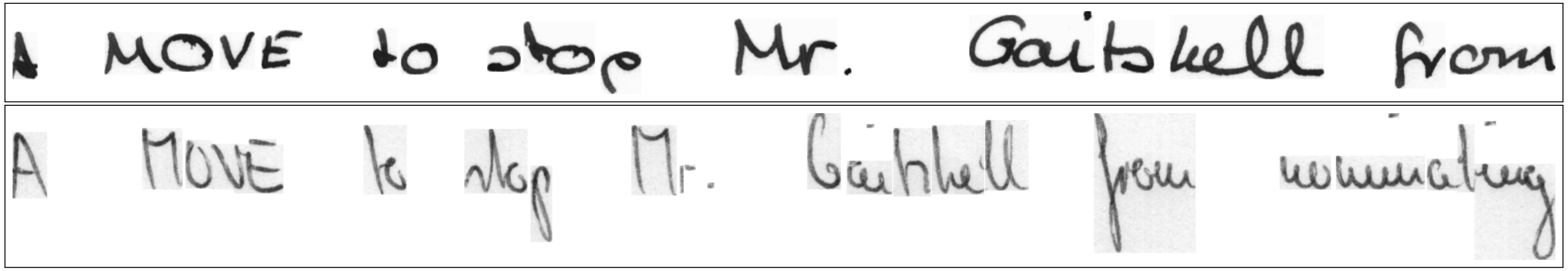
- RWTH-ASR → RWTH-OCR
  - simple feature extraction and preprocessing
  - Arabic: created a SOTA system, ranked 3rd at ICDAR 2009
- discriminative training
  - margin-based HMM training (ML vs. MMI vs. M-MMI)
  - unsupervised confidence-based MMI model adaptation (M-MMI-conf)
- writer adaptive training
  - supervised writer adaptation demonstrated the potential
- ongoing work
  - impact of preprocessing in feature extraction (Arabic vs. Latin)
  - more complex features (e.g. MLP)
  - character context modeling (e.g. CART)
  - Latin: created a SOTA system, best single system

# Outlook: Latin Handwriting - IAM Database

- English handwriting, continuous sentences

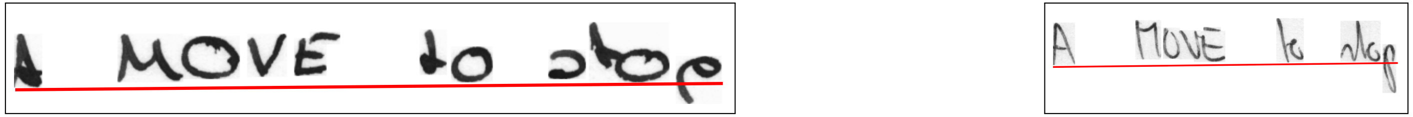
| | Train | Devel | Eval 1 | Eval 2 | Total |
|---|---|---|---|---|---|
| Lines | 6,161 | 1,861 | 900 | 940 | 9,862 |
| Running words | 53,884 | 17,720 | 7,901 | 8,568 | 88,073 |
| Vocabulary size | 7,754 | 3,604 | 2,290 | 2,290 | 11,368 |
| Characters | 281,744 | 83,641 | 41,672 | 42,990 | 450,047 |
| Writers | 283 | 128 | 46 | 43 | 500 |
| OOV Rate | | ≈15% | ≈17% | ≈15% | |

- Example lines:

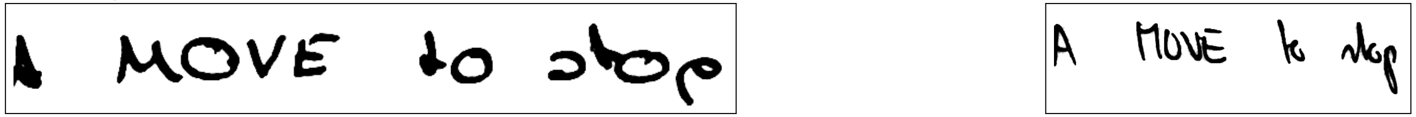

A MOVE to stop Mr. Gaitskell from
A MOVE to stop Mr. Gaitskell from nominating

# Outlook: Latin Handwriting - UPV Preprocessing

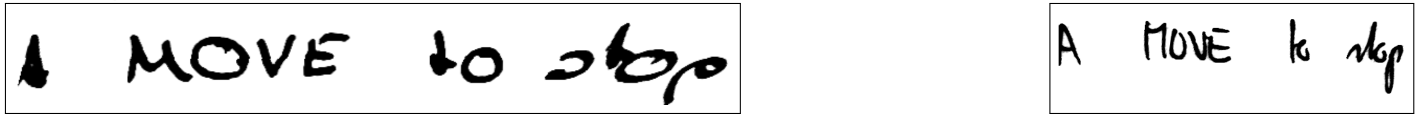
- Original images



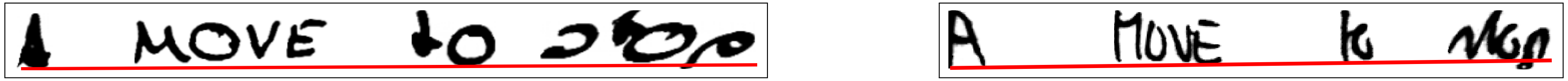
- Images after color normalisation



- Images after slant correction



- Images after height normalisation



**Note: preprocessing did not help for Arabic handwriting** [Visualization]

# Outlook: Latin Handwriting - Experimental Results on IAM Database

| Systems | Devel WER [%] | Eval WER [%] |
|---|---|---|
| **RWTH-OCR** | | |
| Baseline* | 81.07 | 83.60 |
| + UPV Preprocessing* | 57.59 | 65.26 |
| + LBW LM & 50k Lexicon* | 31.92 | 38.98 |
| + discriminative training (M-MMI) | 26.19 | 32.52 |
| + confidences (M-MMI-conf) | - | 31.87 |
| + discriminative training (M-MPE) | 24.31 | 30.07 |
| + confidences (M-MPE-conf) | **23.75** | **29.23** |
| **Other Single HMM Systems** | | |
| [Bertolami & Bunke 08] | 30.98 | 35.52 |
| [Natarajan & Saleem$^{+}$ 08] | - | *40.01*** |
| [Romero & Alabau$^{+}$ 07] | *30.6*** | - |
| **System Combination** | | |
| [Bertolami & Bunke 08] | *26.85* | *32.83* |

***see [Jonas 09] for details**

**** different data**

# Thank you for your attention

**Philippe Dreuw**

`dreuw@cs.rwth-aachen.de`

`http://www-i6.informatik.rwth-aachen.de/`

# References

[Bertolami & Bunke 08] R. Bertolami, H. Bunke: Hidden Markov model-based ensemble methods for offline handwritten text line recognition. *Pattern Recognition*, Vol. 41, No. 11, pp. 3452–3460, Nov 2008. 35

[Dreuw & Heigold$^{+}$ 09] P. Dreuw, G. Heigold, H. Ney: Confidence-Based Discriminative Training for Writer Adaptation in Offline Arabic Handwriting Recognition. In *International Conference on Document Analysis and Recognition*, Barcelona, Spain, July 2009. 30

[Dreuw & Jonas$^{+}$ 08] P. Dreuw, S. Jonas, H. Ney: White-Space Models for Offline Arabic Handwriting Recognition. In *International Congress on Pattern Recognition*, pp. 1–4, Tampa, Florida, USA, Dec 2008. 12, 47

[Dreuw & Rybach$^{+}$ 09] P. Dreuw, D. Rybach, C. Gollan, H. Ney: Writer Adaptive Training and Writing Variant Model Refinement for Offline Arabic Handwriting Recognition. In *International Conference on Document Analysis and Recognition*, Barcelona, Spain, July 2009. 30

[Jonas 09] S. Jonas: Improved Modeling in Handwriting Recognition. Master's thesis, Human Language Technology and Pattern Recognition Group, RWTH Aachen University, Aachen, Germany, Jun 2009. 35

[Natarajan & Saleem$^{+}$ 08] P. Natarajan, S. Saleem, R. Prasad, E. MacRostie, K. Subramanian: *Arabic and Chinese Handwriting Recognition*, Vol. 4768/2008 of *LNCS*, chapter Multi-lingual Offline Handwriting Recognition Using Hidden Markov Models: A Script-Independent Approach, pp. 231–250. Springer Berlin / Heidelberg, 2008. 35

[Romero & Alabau$^{+}$ 07] V. Romero, V. Alabau, J.M. Benedi: Combination of N-Grams and Stochastic Context-Free Grammars in an Offline Handwritten Recognition System. *Lecture Notes in Computer Science*, Vol. 4477, pp. 467–474, 2007. 35

[Rybach & Gollan$^{+}$ 09] D. Rybach, C. Gollan, G. Heigold, B. Hoffmeister, J. Lööf, R. Schlüter, H. Ney: The RWTH Aachen University Open Source Speech Recognition System. In *Interspeech*, Brighton, U.K., Sep 2009. 4

# Appendix: Comparisons for IFN/ENIT

- **ICDAR 2005 Evaluation**

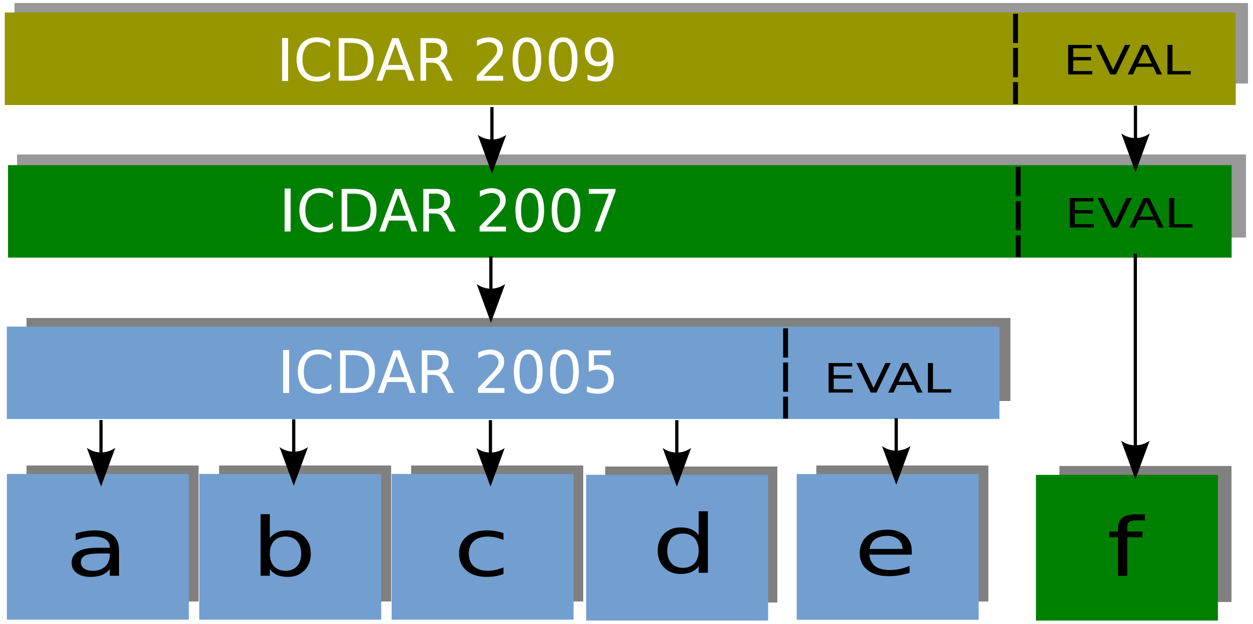
| Rank | Group | WRR [%] | |
|---|---|---|---|
| | | abc-d | abcd-e |
| 1. | UOB | 85.00 | 75.93 |
| 2. | ARAB-IFN | 87.94 | 74.69 |
| 3. | ICRA (Microsoft) | 88.95 | 65.74 |
| 4. | SHOCRAN | 100.00 | 35.70 |
| 5. | TH-OCR | 30.13 | 29.62 |
| | BBN | 89.49 | N.A. |
| 1* | RWTH | **94.05** | **85.45** |

***own evaluation result (no tuning on test data)**

# Appendix: Arabic Handwriting - IFN/ENIT Database

## Corpus development

- ICDAR 2005 Competition: a, b, c, d sets for training, evaluation on set e
- ICDAR 2007 Competition: ICDAR05 + e sets for training, evaluation on set f
- ICDAR 2009 Competition: ICDAR 2007 for training, evaluation on set f

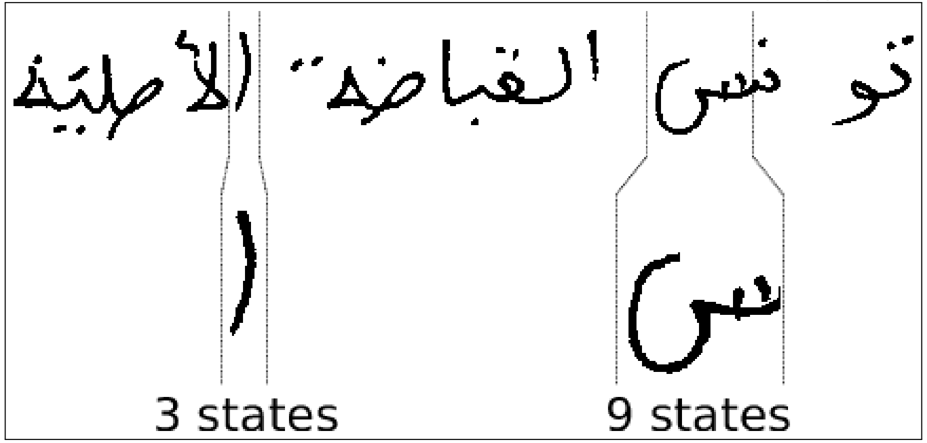


# Appendix: Participating Systems at ICDAR 2005 and 2007

- **MITRE: Mitre Cooperation, USA**
  **over-segmentation, adaptive lengths, character recognition with post-processing**
- **UOB-ENST: University of Balamand (UOB), Lebanon and Ecole Nationale Superieure des Telecommunications (ENST), Paris**
  **HMM-based (HTK), slant correction**
- **MIE: Mie University, Japan**
  **segmentation, adaptive lengths**
- **ICRA: Intelligent Character Recognition for Arabic, Microsoft**
  **partial word recognizer**
- **SHOCRAN: Egypt**
  **confidential**
- **TH-OCR: Tsinghua Universty, Beijing, China**
  **over-segmentation, character recognition with post-processing**
- **CACI: Knowledge and Information Management Division, Lanham, USA**
  **HMM-based, trajectory features**
- **CEDAR: Center of Excellence for Document Analysis and Recognition, Buffalo, USA**
  **over-segmentation, HMM-based**
- **PARIS V / A2iA: University of Paris 5, and A2iA SA, France**
  **hybrid HMM/NN-based, shape-alphabet**
- **Siemens: SIEMENS AG Industrial Solutions and Services, Germany**
  **HMM-based, adapative lenghths, writing variants**
- **ARAB-IFN: TU Braunschweig, Germany**
  **HMM-based**

# Appendix: Visual Modeling - Model Length Estimation

- more complex characters should be represented by more HMM states



- the number of states $S_c$ for each character $c$ is updated by

$$S_c = \frac{N_{x,c}}{N_c} \cdot \alpha$$

with

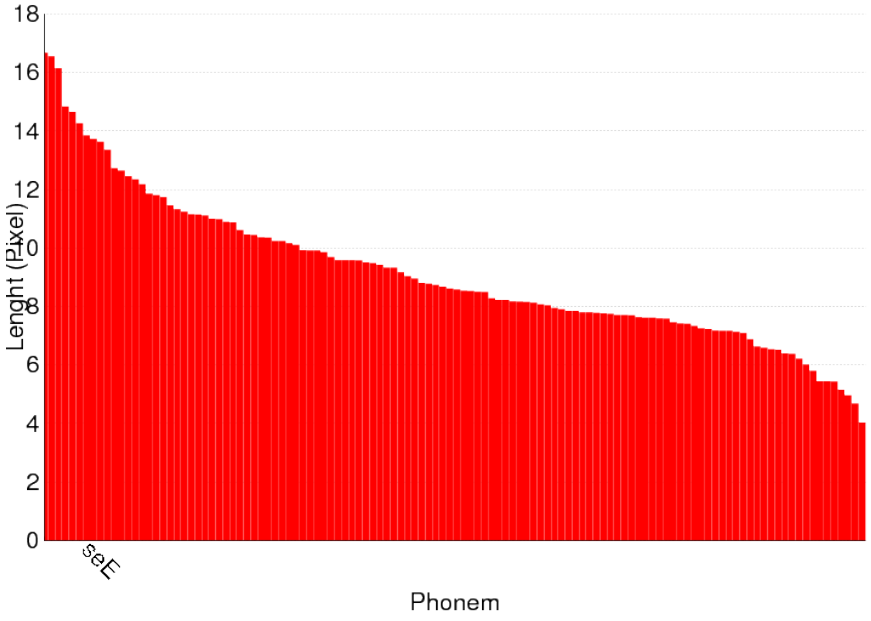
| | | |
|---|---|---|
| $S_c$ | = | estimated number states for character $c$ |
| $N_{x,c}$ | = | number of observations aligned to character $c$ |
| $N_c$ | = | character count of $c$ seen in training |
| $\alpha$ | = | character length scaling factor. |

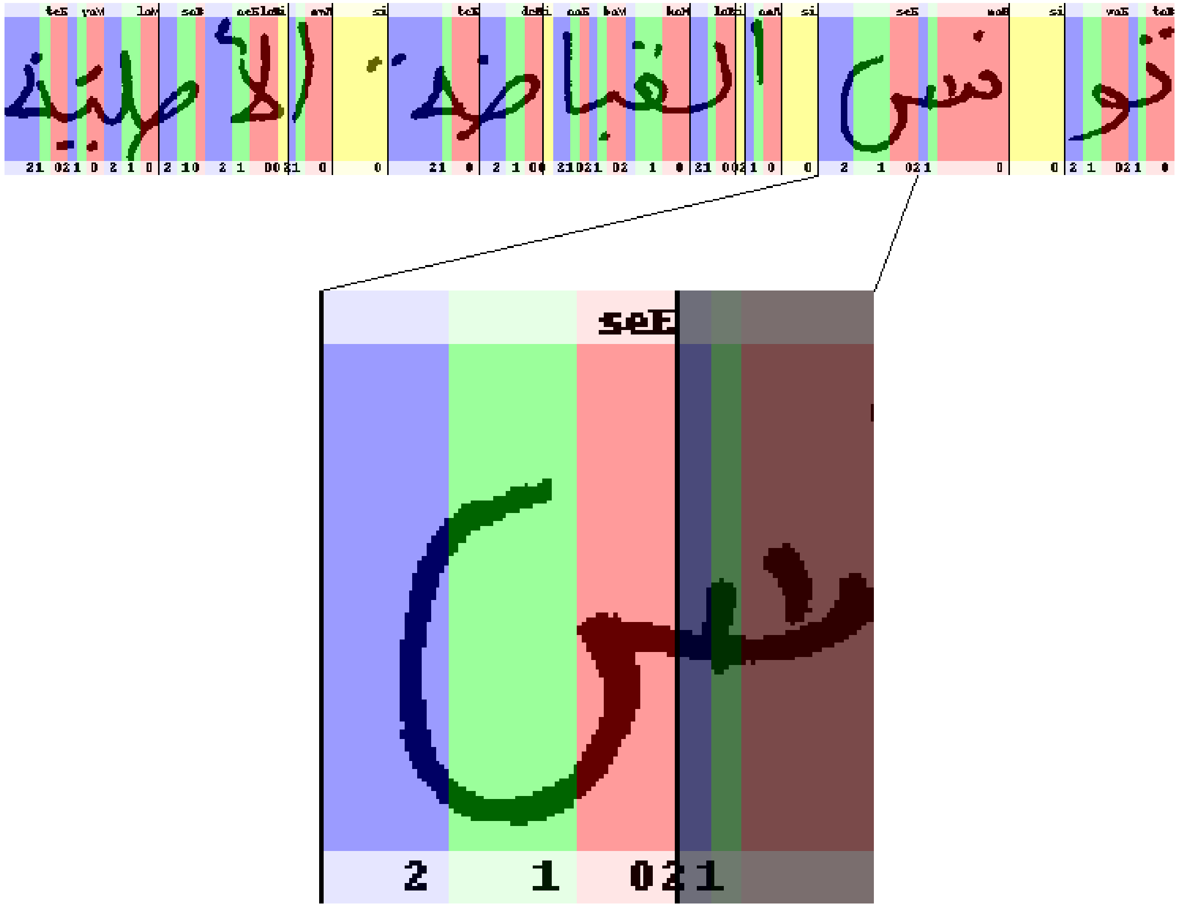
[Visualization]

# Appendix: Visual Modeling - Model Length Estimation

## Original Length

- overall mean of character length = 7.9 pixel (≈ 2.6 pixel/state)
- total #states = 357

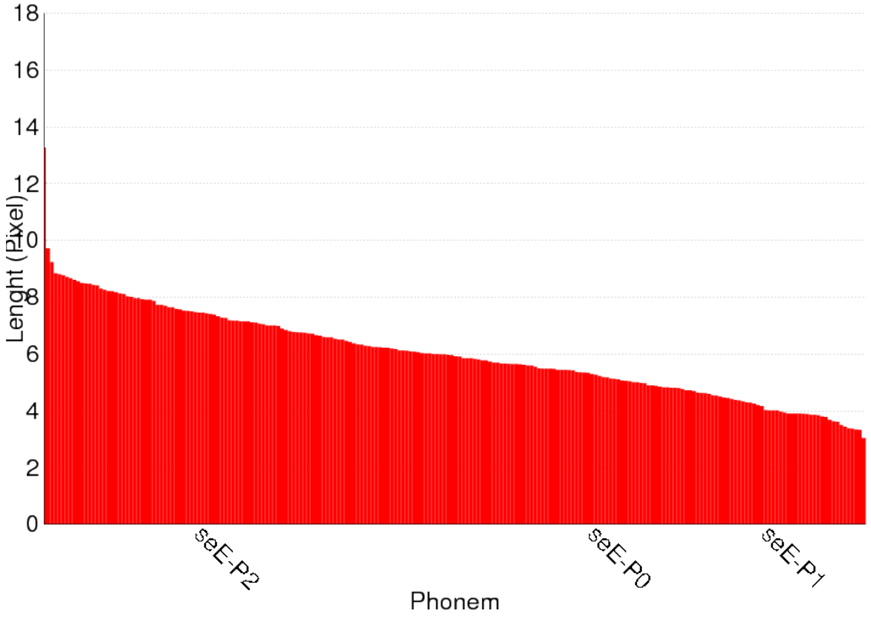


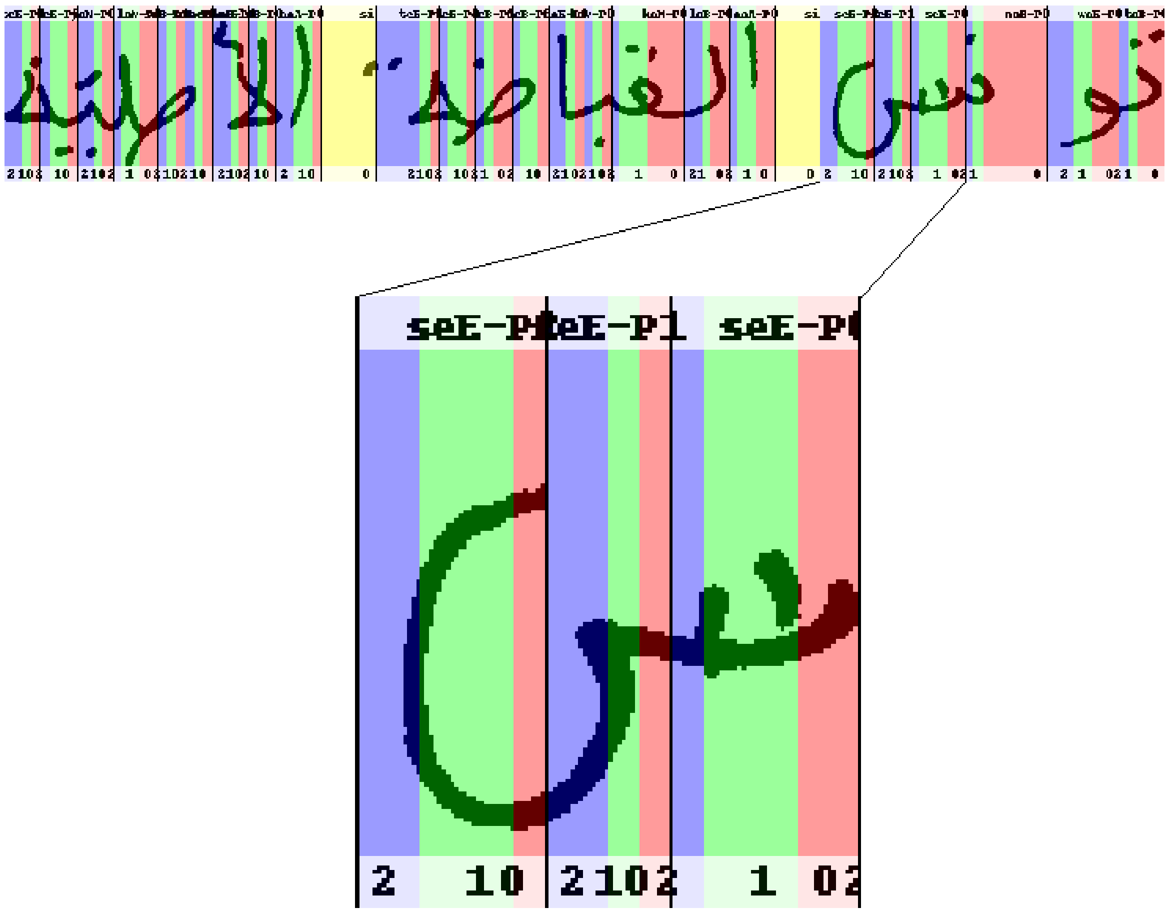


# Appendix: Visual Modeling - Model Length Estimation

## Estimated Length

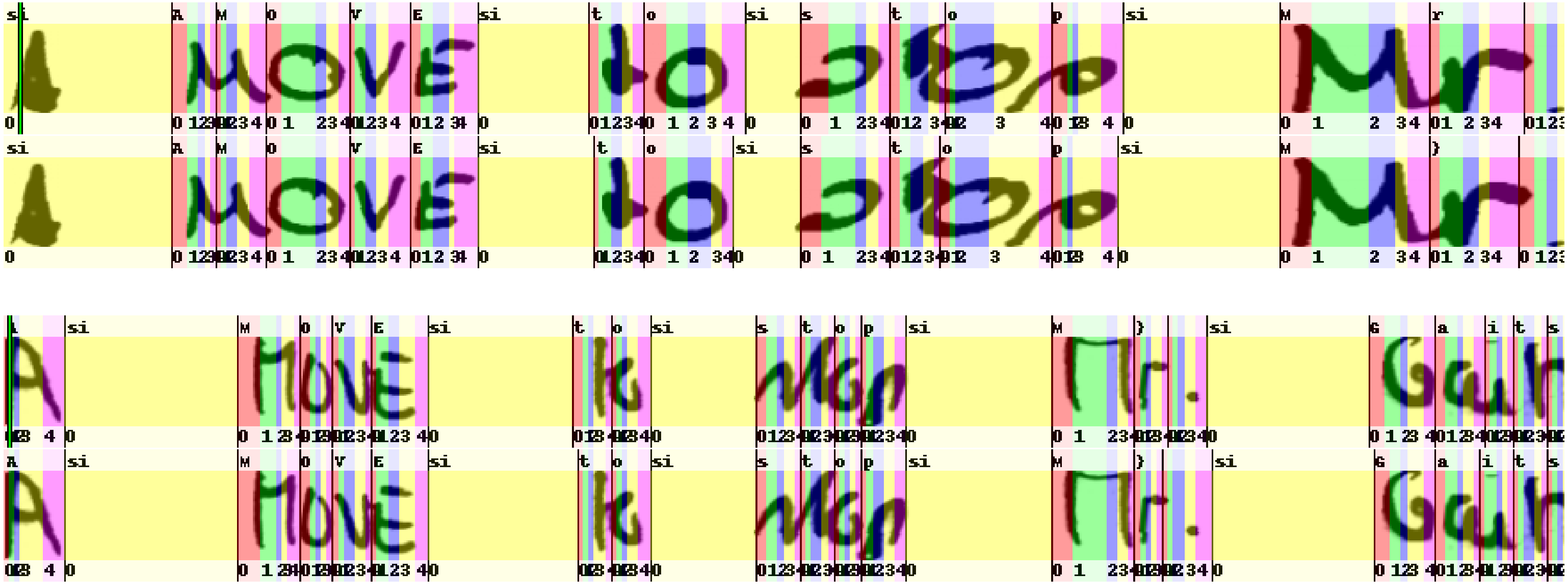
- overall mean of character length = 6.2 pixel (≈ 2.0 pixel/state)
- total #states = 558





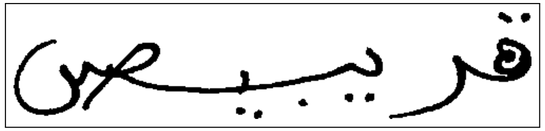
# Appendix: Alignment Visualization

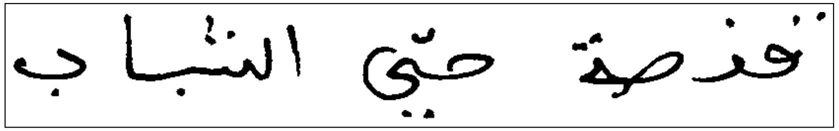
- **alignment visualization with and without discriminative training**
- **upper lines with 5-2 baseline setup, lower lines with additional discriminative training**


si A M o V E si t o si s t o p si M r
si A M o V E si t o si s t o p si M }
A si M o V E si t o si s t o p si M } si G a i t s
A si M o V E si t o si s t o p si M } si G a i t s

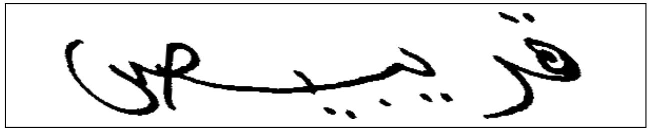
# Appendix: Arabic Handwriting - UPV Preprocessing

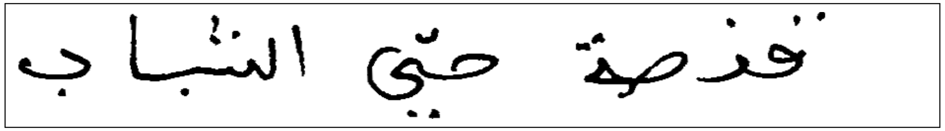
- Original images



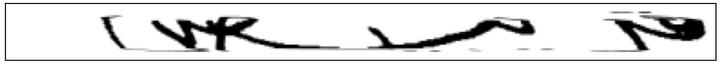


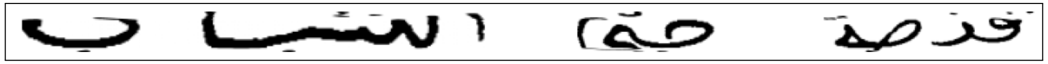
- Images after slant correction





- Images after size normalisation



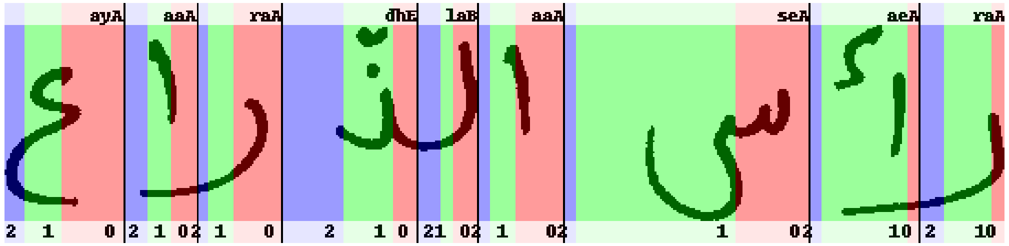


## Experimental Results:

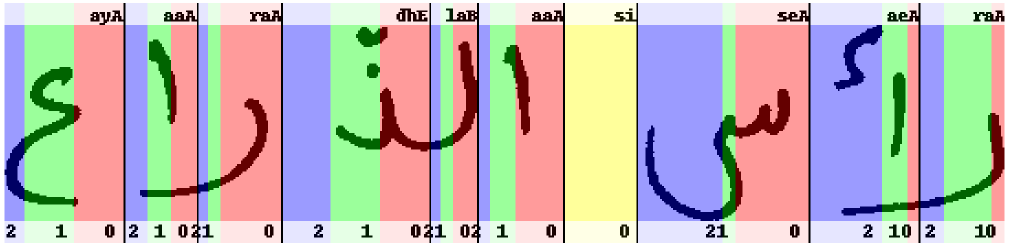
- important informations in ascender and descender areas are lost
- not yet suitable for Arabic HWR

# Appendix: Visual Modeling - Writing Variants Lexicon

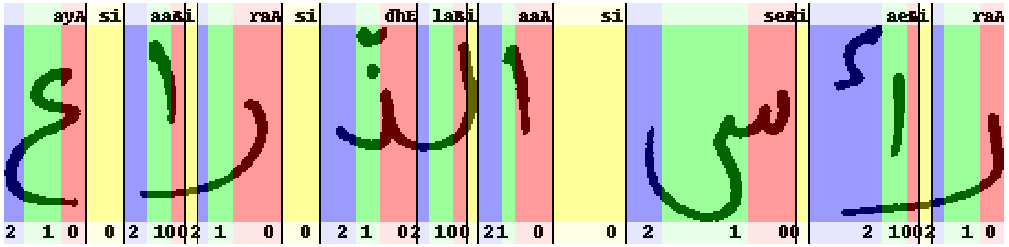
- most reported error rates are dependent on the number of PAWs
- without separate whitespace model



- always whitespaces between compound words



- whitespaces as writing variants between and within words



**White-Space Models for Pieces of Arabic Words [Dreuw & Jonas$^+$ 08] in ICPR 2008**

# Appendix: Constrained Maximum Likelihood Linear Regression

**Idea:** improve the hypotheses by adaptation of the features $x_t$

- effective algorithm for adaptation to a new speaker or environment (ASR)
- GMMs are used to estimate the CMLLR transform
- iterative optimization (ML criterion)
  - align each frame $x$ to one HMM state (i.e. GMM)
  - accumulate to estimate the adaptation transform $A$
  - likelihood function of the adaptation data given the model is to be maximized with respect to the transform parameters $A, b$
- one CMLLR transformation per (estimated) writer
- constrained refers to the use of the same matrix $A$ for the transformation of the mean $\mu$ and variance $\Sigma$:

$$x_t' = Ax_t + b \rightarrow N(x|\hat{\mu}, \hat{\Sigma}) \text{ with } \begin{aligned} \hat{\mu} &= A\mu + b \\ \hat{\Sigma} &= A\Sigma A^T \end{aligned}$$