CACI

EVER VIGILANT™

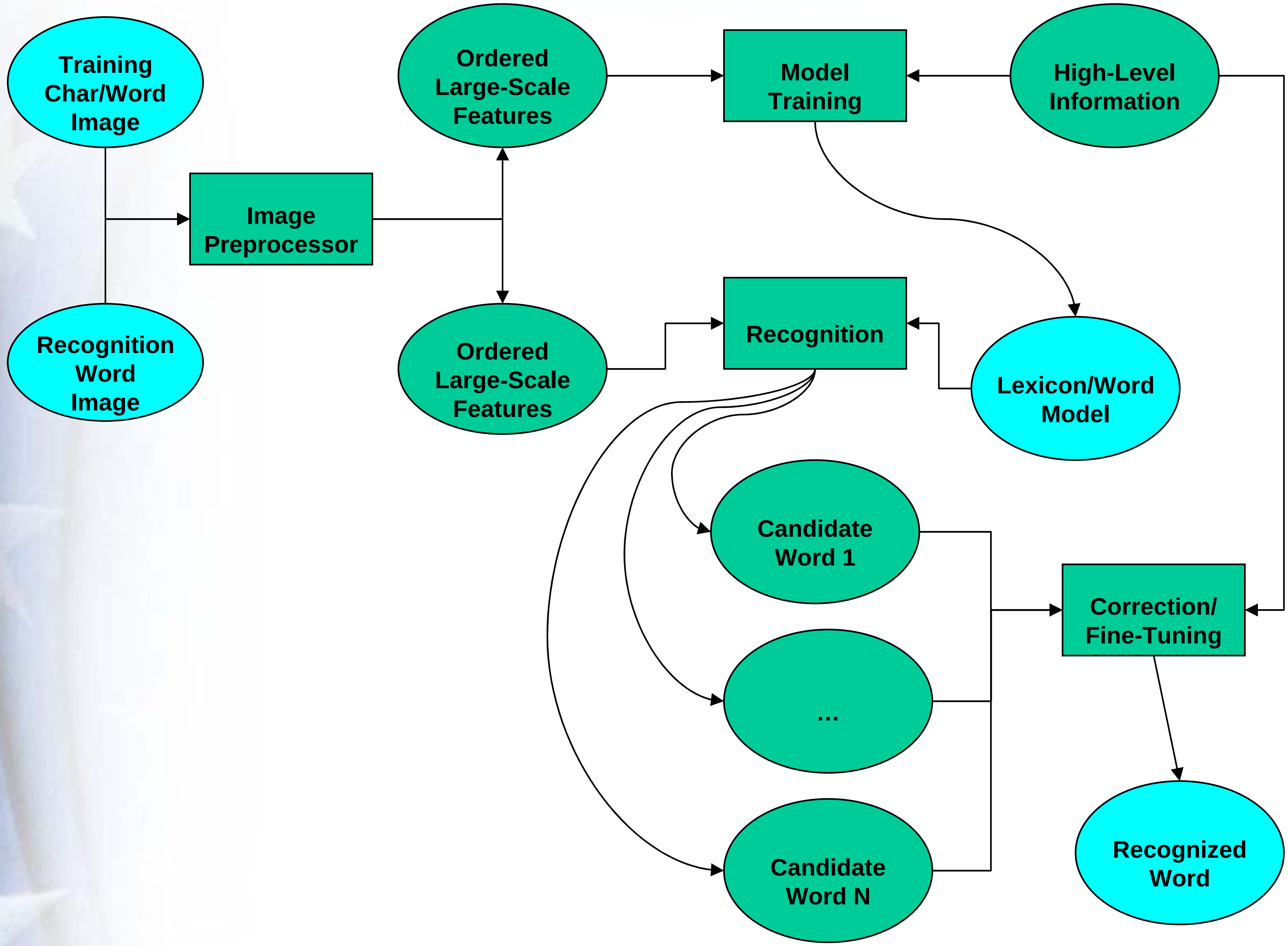
Initial Results in Offline Arabic Handwriting Recognition Using Large-Scale Geometric Features

Ilya Zavorin, Eugene Borovikov, Mark Turner

System Overview

- **Based on large-scale features:** robust to handwriting variations, both inter-writer and intra-writer
- **Simulates writing process:** robust to overlaps
- **Trainable:** uses discrete Hidden Markov Models (HMM)
- **Uses high-level information:**
  - Permits "filling in" gaps in recognition unavoidable in degraded handwritten text
  - Resolves ambiguity
- **Based on work by Khorsheed, with various modifications**

Processing Flowchart



Large-Scale Features

- **Represent connected components of image of letter or word as** *sequence* **of simple** *geometric objects*, **e.g. loops, turning and intersection points**
- **Example: "Mem"**  **= LOOP, TURN, END**
- **Robust to:**
  - Handwriting variations: size, orientation, local distribution
  - Letter variants: positional forms, casheedas
  - Individual style differences
  - Poor image quality

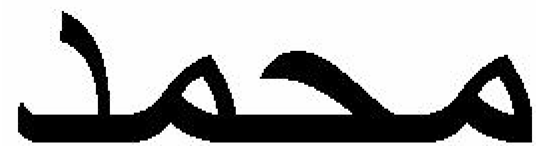
Simulation of Writing

- **Individually process different connected components (CCs) of image skeleton**

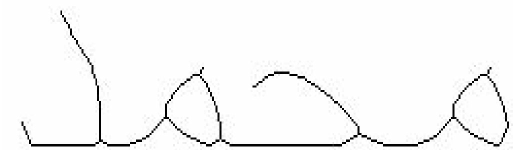
- **Use set of consistent tracing rules to traverse each CC**
- **Rules resemble those of human handwriting, but *any consistent set of rules can be used***

Preprocessing Stages

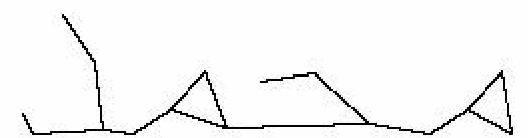
- **Image:**



- **Skeleton:**



- **Stick Approximation:**



- **Labeled Sequence: LOOP,TURN,X,END,...**

Discrete HMMs

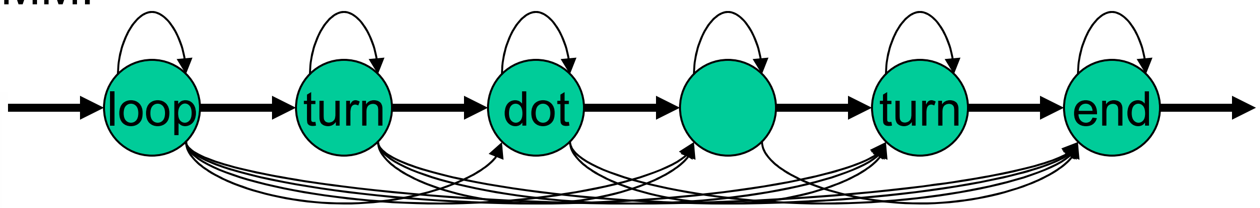
CACI
EVER VIGILANT

- **"Universal" model of lexicon or collection of word models assembled from individual letter HMMs**
- **Letter model size proportional to letter complexity**
- **Universal combination reflects letter combination probabilities (bigram or other)**
- **Very general framework:** the only requirement is ordered set of discrete features/events to be used for training and recognition

Discrete HMMs

single letter HMM:

ق



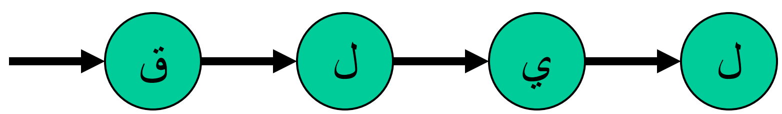
Topology permits skipped and repeated components

universal (composite) HMM based on individual letter HMMs and bigram statistics:

	ا	ب	…	ي
ا	p_{11}	p_{12}	…	p_{1n}
ب	p_{21}	p_{22}	…	p_{2n}
…	…	…	…	…
ي	p_{n1}	p_{n2}	…	p_{nn}

قليل

character block (word) HMM:



Training Modes

- **Original algorithm: training individual letter models by manually pre-segmented data**
  - Straight-forward, accurate training
  - Not practical for high volume systems
- **Improvement: training by unsegmented words using word models**
  - Lose some accuracy, but
  - Preprocessing becomes fully automatic

Training with manual segmentation

1. **For each training word**
   I. Manually pre-segment into character sub-images
   II. Convert character sub-images into observation sequences
2. **For each character**
   I. Compute initial model
   II. Collect all training sequences from data in Step **1.**
   III. Train character model
3. **Connect all trained character models into universal model using bigram statistics**

Training with no manual segmentation

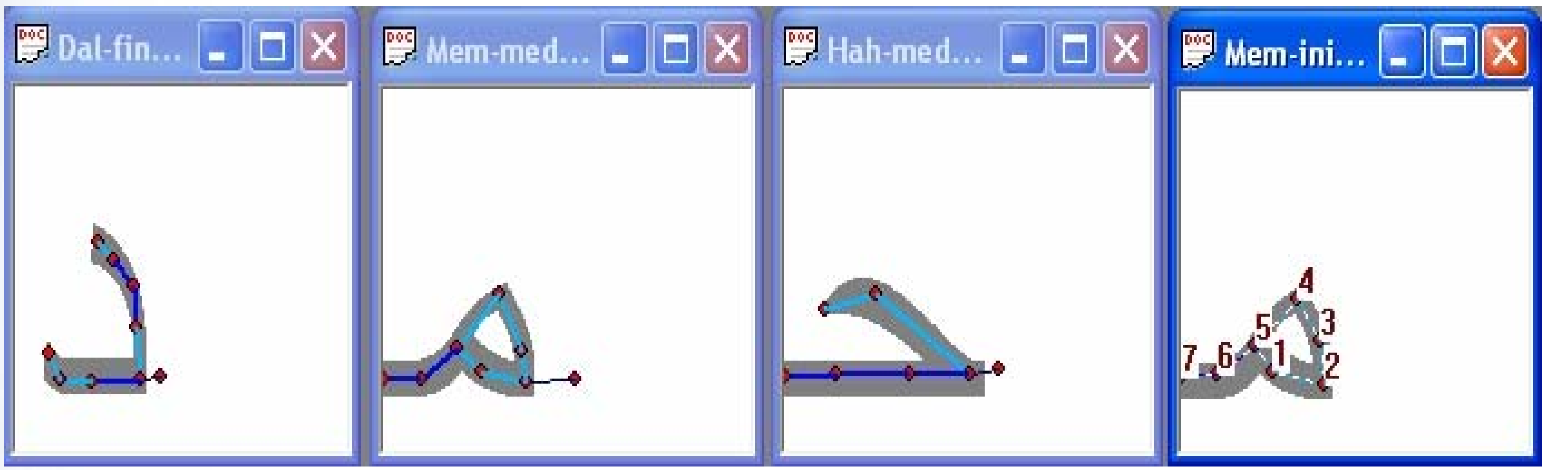
1. **For each character, compute initial model**
2. **For each training word**
   1. Assemble initial character models into word model
   2. Convert entire training word image into observation sequence
   3. Use sequence to train model
   4. Disassemble word model into instances of trained character models
3. **For each character, combine all instances into single trained character model**
4. **Connect all trained character models into universal model using bigram statistics**

Experiments

- **Proof of Concept**
  - Manual preprocessing of segmented letters
  - Emulate handwriting differences by random perturbation ("shaking")
  - Training and test words created by concatenation

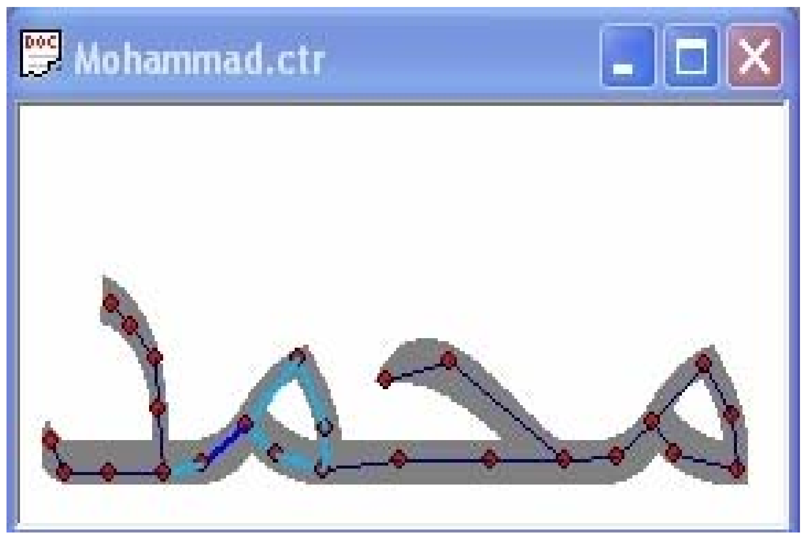
Manual Processing

- **Using specially designed graphical tool**
- **Stick traces superimposed onto "template" images**
- **Segments labeled**



Training/Test Word Concatenation

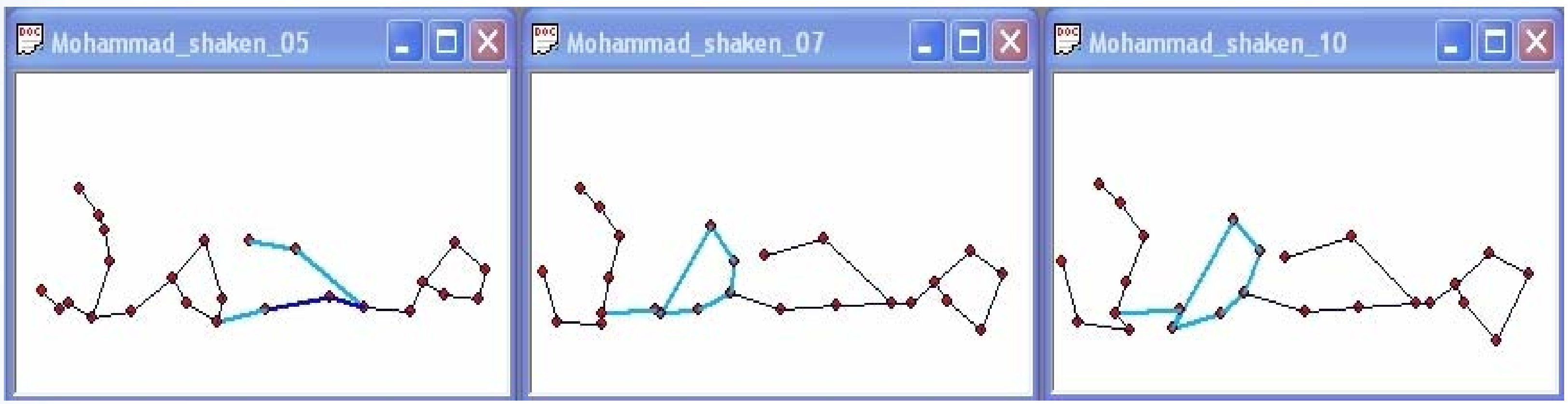
- **Traces merged automatically**



Handwriting Style Emulation

- **Traces randomly perturbed ("shaken")**
- **Letter models trained by shaken skeletons**
- **Perturbation measured by standard deviation, in pixels**

Image size ≈ 100 x 200 pixels



$\sigma = 5$ $\sigma = 7$ $\sigma = 10$

Test data

- **Tested on *synthetic* data**
- **29 main Arabic characters included in model**
- **Characters models with 3 to 20 hidden states**
- **Lexicon of 2000+ words**
- **Various degrees of perturbation, from σ=0 to σ=15**
- **Three sets of tests**
  1. Machine-printed images as templates; train *with* pre-segmentation
  2. Machine-printed images as templates; train *without* pre-segmentation
  3. Machine-printed and handwritten images as templates; train *without* pre-segmentation

Test Sets 1 and 2: Results

		With Segmentation	Without Segmentation
Character Accuracy	Precision	99.5%	93.1%
	Recall	99.6%	92.1%
Word Accuracy	Number of word sequences	107950	10800
	Number of matches	105691	8059
	Recognition rate	97.9%	74.6%

Test Set 3: Handwritten and machine-printed data

CACI
EVER VIGILANT

- Purpose: to show that one can model one handwriting style with perturbed versions of other styles
- Two words, 3 handwritten and 1 machine-printed samples of each
- Skeletons created manually
- Trained using perturbed handwritten skeletons, no pre-segmentation (as in Test 2)
- Tested on machine-printed skeleton

Test Set 3: Training and Test Data

CACI
EVER VIGILANT

Test Set 3: Results

- **Tested universal model**
- **Both words correctly recognized**
- **Perturbation during training was critical: no path at all through universal model without it**

Future Work

- **Automatic preprocessing**
  - Rectification
  - Feature extraction
- **Study role of high-level info**
  - Simple human handwriting recognition tests using native and non-native Arabic speakers to establish baseline of human performance
- **Tests on more realistic data**
  - IFN/ENIT
  - ISG
  - Other datasets