

ICFHR–2010 Tutorial:
Multimodal Computer Assisted Transcription of
Handwriting Images
Ip – Off-line HTR in practice

Alejandro H. Toselli, Moisés Pastor and Verónica Romero

{ahector, moises, vromero}@iti.upv.es



Pattern Recognition and Human Language Technology Group

Instituto Tecnológico de Informática – Universidad Politécnica de Valencia



Spain

November 2010

ICFHR 2010: Multimodal Computer Assisted Transcription of Handwriting Images

A B L A N K P A G E

Tutorial Contents and Schedule

- I Introduction
 - Multimodal Interaction in Pattern Recognition
 - Quick Survey of Handwritten Text Recognition (HTR) concepts and techniques
 - Interactive-Predictive Pattern Recognition and Document Image Analysis
- I-p **Off-line HTR in practice**
 - HTR Preprocessing
 - Training HMMs using the "Hidden Markov Model Toolkit" (HTK)
 - Training Language Models and Dictionaries for HTR
 - HTR Experiments
- II Computer-Assisted Transcription of Text Images (CATTI)
 - Human interaction in HTR
 - A CATTI formal framework
 - Feedback-derived dynamic language modelling and search
 - Performance measures and results achieved in typical applications
- II-p CATTI in practice
 - Adapting Language Models and Search for CATTI
 - CATTI Experiments
 - Analyzing quantitatively the CATTI performance
- III Multimodality in CATTI (MM-CATTI)
 - Touchscreen based multimodal user correction
 - A MM-CATTI formal framework
 - Multimodal language modelling and search
 - Performance measures and results achieved in typical applications
- III-p Demonstration of a complete MM-CATTI System in a real HTR task

CATTI-Tutorial Website Download

URL address to download all the tutorial stuffs:

<http://prhlt.iti.es/tutorial/icfhr2010>

Files to download:

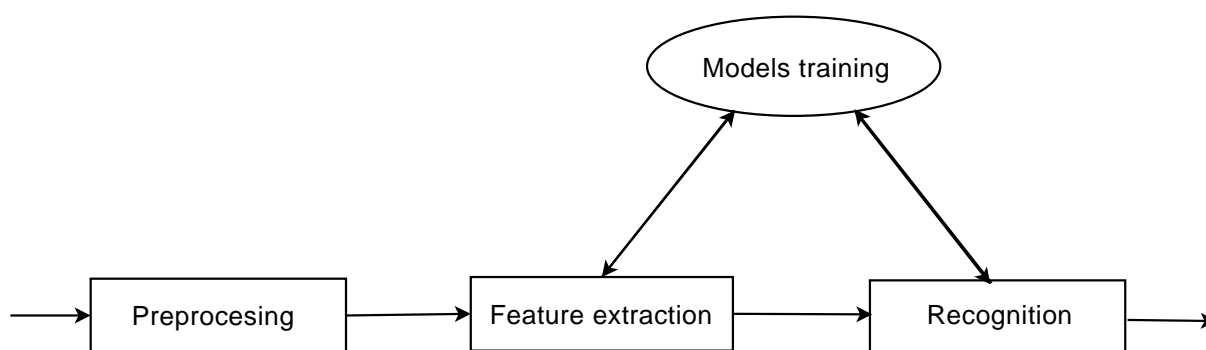
- Tutorial Slides
- [Experiment Guide](#)
- [Spanish-Number Corpus](#)
- [HTR processing tools](#)
- [HTK ToolKit 3.4](#)
- [SRI Language Modeling Toolkit \(SRILM\)](#)

Index

- 1 Handwritten Text Recognition Architecture ▷ 4
- 2 HTR: Corpus ▷ 5
- 3 HTR: Preprocessing ▷ 9
- 4 HTR: Feature Extraction ▷ 17
- 5 HTR: Training, Modelling and Decoding ▷ 22
- 6 Bibliography ▷ 28

HTR Segmentation-Free Architecture

Classical ASR-like architecture, composed of some main modules:



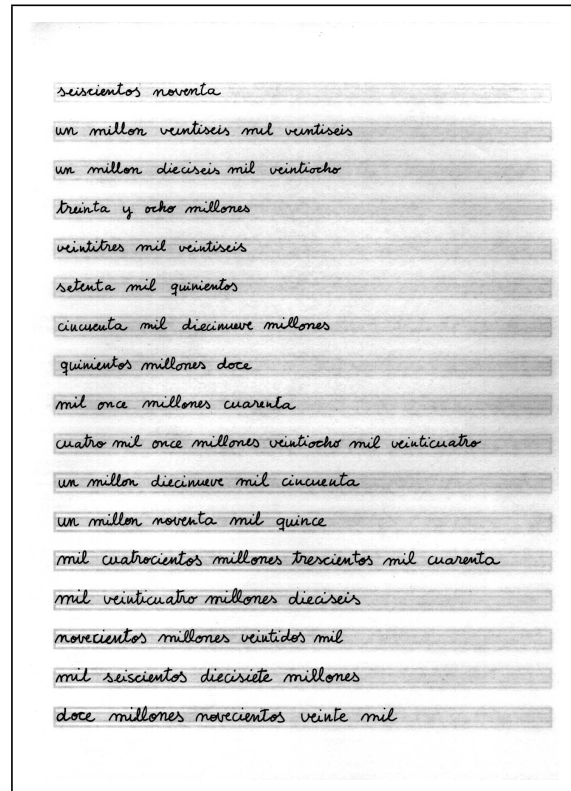
- *Preprocessing*: noise removal, line detection and geometric normalizations
- *Feature Extraction*: grey-levels or point coordinates and their derivatives
- *Modeling*: morphological Hidden Markov Models + N-gram Language model
- *Recognition*: Viterbi search

“Spanish Number” Task: Acquisition

Writers : 19 Phrases : 307 - 323

=====

307 seiscientos noventa
 308 un millon veintiseis mil veintiseis
 309 un millon dieciseis mil veintiocho
 310 treinta y ocho millones
 311 veintitres mil veintiseis
 312 setenta mil quinientos
 313 cincuenta mil diecinueve millones
 314 quinientos millones doce
 315 mil once millones cuarenta
 316 cuatro mil once millones veintiocho mil veinticuatro
 317 un millon diecinueve mil cincuenta
 318 un millon noventa mil quince
 319 mil cuatrocientos millones trescientos mil cuarenta
 320 mil veinticuatro millones dieciseis
 321 novecientos millones veintidos mil
 322 mil seiscientos diecisiete millones
 323 doce millones novecientos veinte mil



“Spanish Number” Task: Corpus Partition

Text lines examples from the “Spanish-Numbers” corpus:

trescientos sesenta y ocho mil veintiocho
 sesenta millones ochenta y siete mil veinticuatro
 un millon seiscientos doce

	Train	Test	Total	Lexicon
# writers	18	11	29	–
# sentences	297	187	484	–
# words	1 300	827	2 127	52

“Spanish Number” Task

- Multi-writer: 29.
- 19 character HMMs (including “@” symbol to represent the inter-word space):

@ a c d e h i l m n o q r s t u v y z

- Vocabulary: 52 words.
- Bi-grams language model:
 - Trained from 20 920 samples of “*Spanish legal amount numbers*” (91 760 **running words**).
 - 54 unigrams and 688 bi-grams
 - Back-off smoothing (Good-Turing discounting).

“Spanish Number” Task: Modeling

- 19 character classes modeled by left-to-right continuous density HMMs: 6 states and 10 transitions (2 per states).
- Each HMM state emission is given by mixtures of Gaussians densities with diagonal covariances: $NG \in \{1, 2, 4, 8\}$.
- 52 word models (SFSA) which take in account all possible ways to write each of the words.
- **LM**: bi-grams with back-off smoothing.
- *Baum-Welch* “embedded training” algorithm was employed to train the 19 HMMs character class models.
- The recognition process was performed by Viterbi “*beam-search*” on the global integrated model.
- Training and recognition were carried out using HTK ToolKit.

Off-Line HTR Preprocessing

Two preprocessing schemes at different levels are considered:

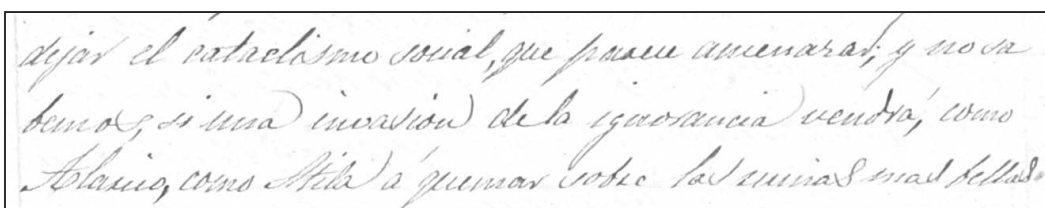
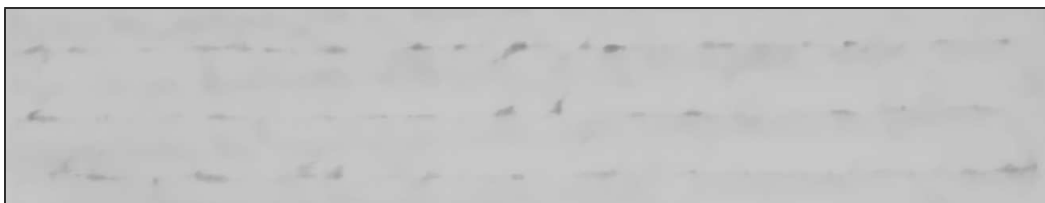
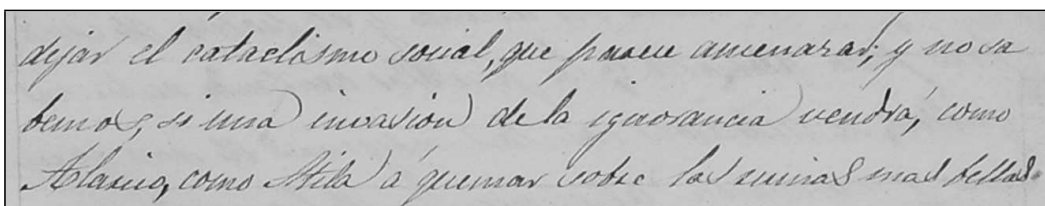
- Page or text-block preprocessing: *background removal, noise reduction, skew correction and text line extraction.*
- Line preprocessing: *Slope/slant corrections and non-linear size normalization.*



They intend to sit outside the Ministry of Defence .
 opposed the Government Bill which brought
 His Majesty's United Federal Party in opposition
 He said these concerned Mr. Weaver's
 association with organizations controlled by the
 Government - Immediately Mr. Cassidy asked a letter
 Sir Pierson Dixon, Britain's Ambassador to
 (Treasury), and Mr. G. H. Andrew (Board of Trade).

Background removal and noise reduction

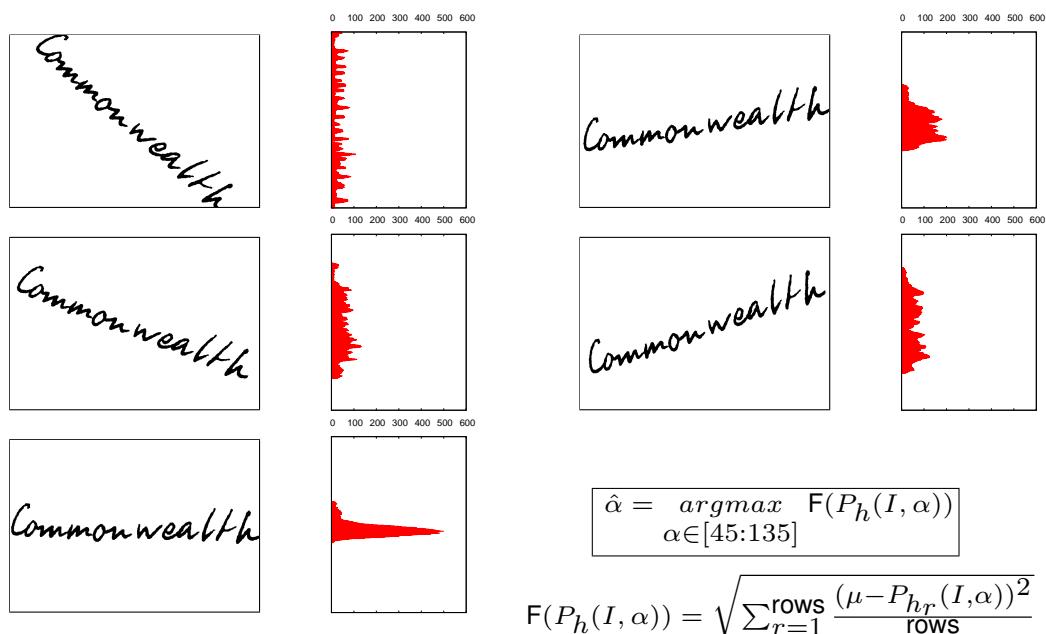
2-D median filtering, image subtraction and grey-level normalization



Preprocessing at Text Line Image Level

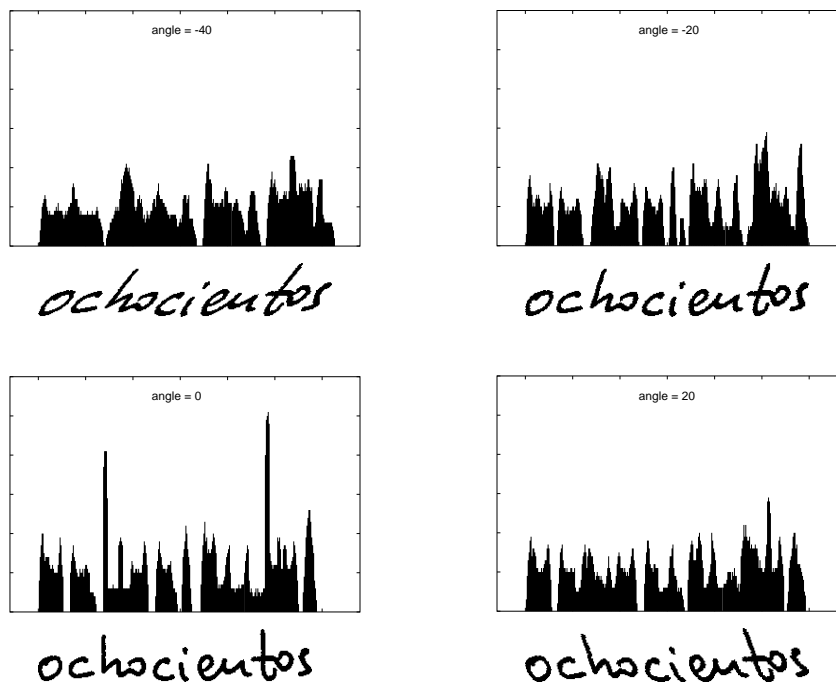
- *Slope angle*: angle of the handwritten text line with respect to the horizontal direction.
- *Slant angle*: angle of the handwritten text strokes with respect to the vertical direction.
- *Handwritten text height*: which can vary according to the task and writers. Of particular interest is the relationship among the sizes of ascender letters (e.g.: b, l, t), descender letter (e.g.: p, q, j) and normal letters (e.g.: a, c, u).
- *Character width*: like the text height, width of characters can vary according to the task and writers.
- *Stroke thickness*: the use of different writing elements can lead to a variable strokes types and thickness.

HTR preprocessing: Slope detection and correction



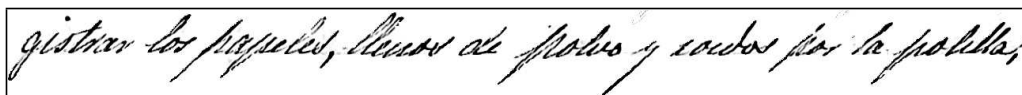
HTR preprocessing: Slant detection and correction

By maximizing the variance of the de-slanted vertical projection profile

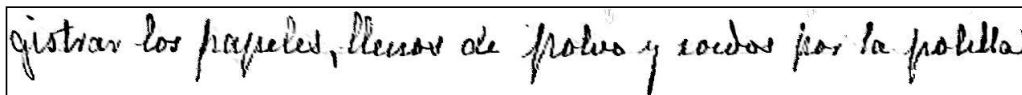


HTR preprocessing: Slant and Size normalization

Original line image:



Slant correction:

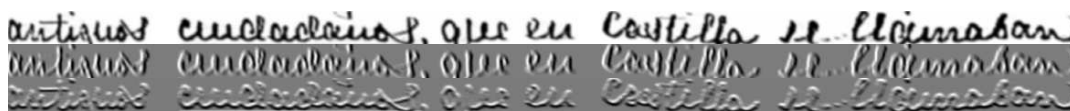


Non-linear vertical size normalization:

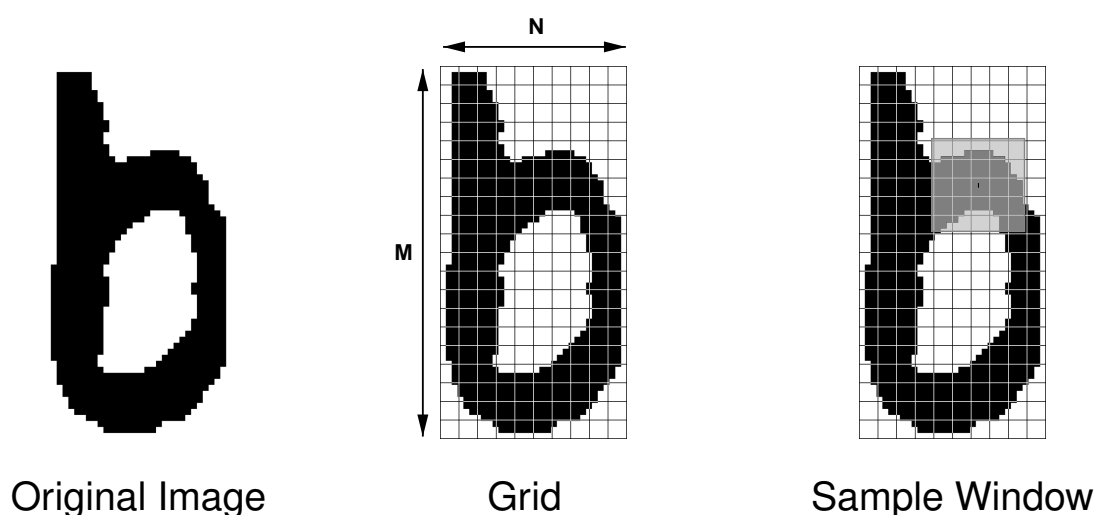


Feature Extraction for Off-Line HTR

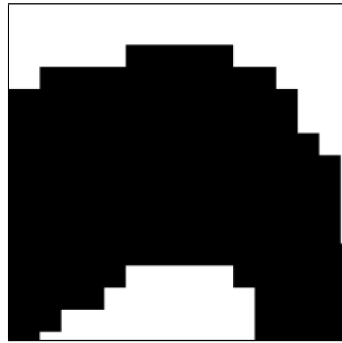
- A grid is applied to divide the image into $N \times M$ squared cells M satisfies the condition $M/N = \text{image aspect ratio}$
- Three features are calculated for each cell:
 - Normalized gray level
 - Horizontal gray level derivative
 - Vertical gray level derivative
- Columns of cells (*frames*) are processed from left to right and a feature vector is constructed for each *frame* by stacking the three features computed in its constituent cells
- Each text line image is represented as a sequence of $(3 \times M)$ -dimensional feature vectors



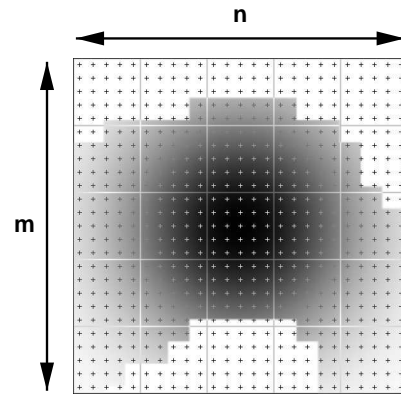
Sample Window



Feature Extraction: Grey Level Feature



Stroke segment enclosed by the sampling window

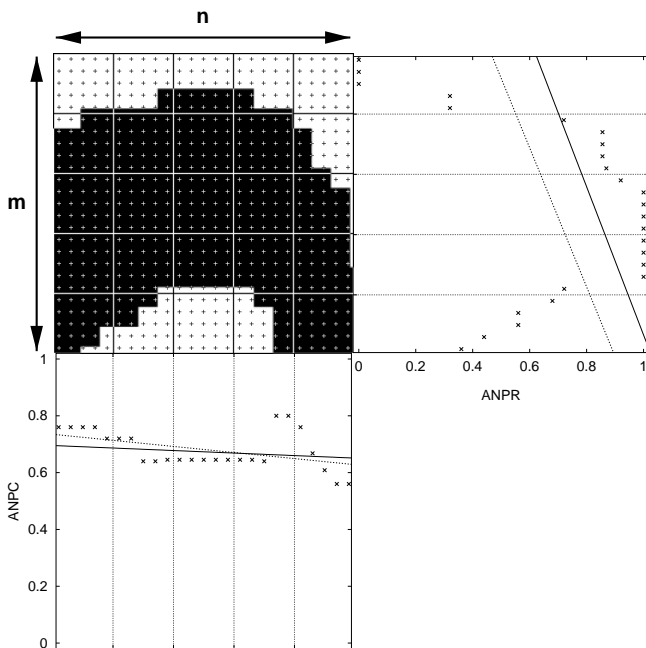


Sub-sampling points defined inside the sampling window and spread out over the stroke segment smoothed by the Gaussian filter

The image intensity $\hat{I}(i, j)$, smoothed by a Gaussian filter, is:

$$\hat{I}(i, j) = I(i, j) \exp \left[-\frac{1}{2} \left(\frac{j - (n/2)^2}{(n/4)^2} + \frac{i - (m/2)^2}{(m/4)^2} \right) \right]$$

Feature Extaction: Derivative Features



Average grey level per column (AGLPC):

$$g_j = \frac{\sum_{i=1}^m I(i, j)}{m}$$

Linear approximation: $\mathbf{mse} \rightarrow 0$

$$\mathbf{mse}(a, b) = \sum_{j=1}^n w_j (g_j - (a \cdot j + b))^2$$

Gaussian filter:

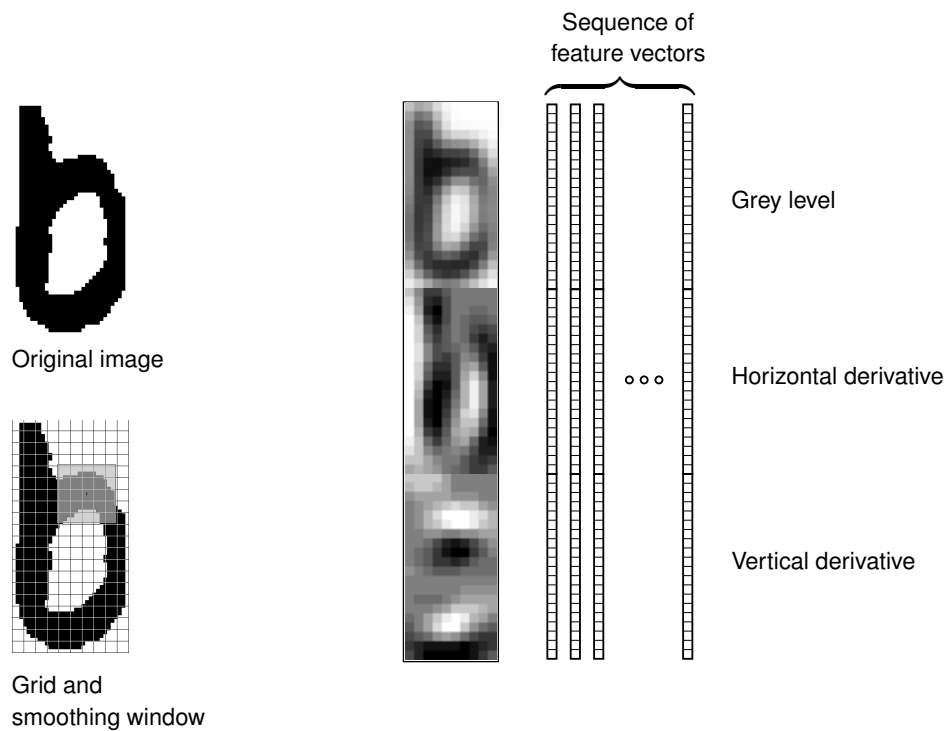
$$w_j = \exp \left(-\frac{1}{2} \frac{(j - n/2)^2}{(n/4)^2} \right)$$

Restriction:

$$\frac{\partial \mathbf{mse}(a, b)}{\partial a} = 0 \quad \text{and} \quad \frac{\partial \mathbf{mse}(a, b)}{\partial b} = 0$$

$$\text{Derivative: } a = \frac{\left(\sum_{j=1}^n w_j g_j \right) \left(\sum_{j=1}^n w_j j \right) - \left(\sum_{j=1}^n w_j \right) \left(\sum_{j=1}^n w_j g_j j \right)}{\left(\sum_{j=1}^n w_j j \right)^2 - \left(\sum_{j=1}^n w_j \right) \left(\sum_{j=1}^n w_j j^2 \right)}$$

Feature extraction: Sequence of real-values vectors



Statistical framework for HTR

Handwritten Text Recognition: Given a stream of feature vectors representing text (line) image, x , and a set of morphological character, lexicon and language models, \mathcal{M} , obtain a sequence of words (transcription) from which x can be produced with maximum likelihood; that is:

$$\hat{w} = \underset{w}{\operatorname{argmax}} P_{\mathcal{M}}(w | x)$$

Using the Bayes theorem (and dropping \mathcal{M} to simplify notation):

$$\hat{w} = \underset{w}{\operatorname{argmax}} P(x | w) \cdot P(w)$$

Popular models:

- $P(x | w)$: *morphological HMMs*
- $P(w)$: *N-Gram Language Model*

Balancing models impact in practice: *Grammar Scale Factor*

$$\hat{w} = \underset{w}{\operatorname{argmax}} P(x | w)^{(1-\alpha)} \cdot P(w)^{\alpha} \equiv \underset{w}{\operatorname{argmax}} P(x | w) \cdot P(w)^{\alpha'}$$

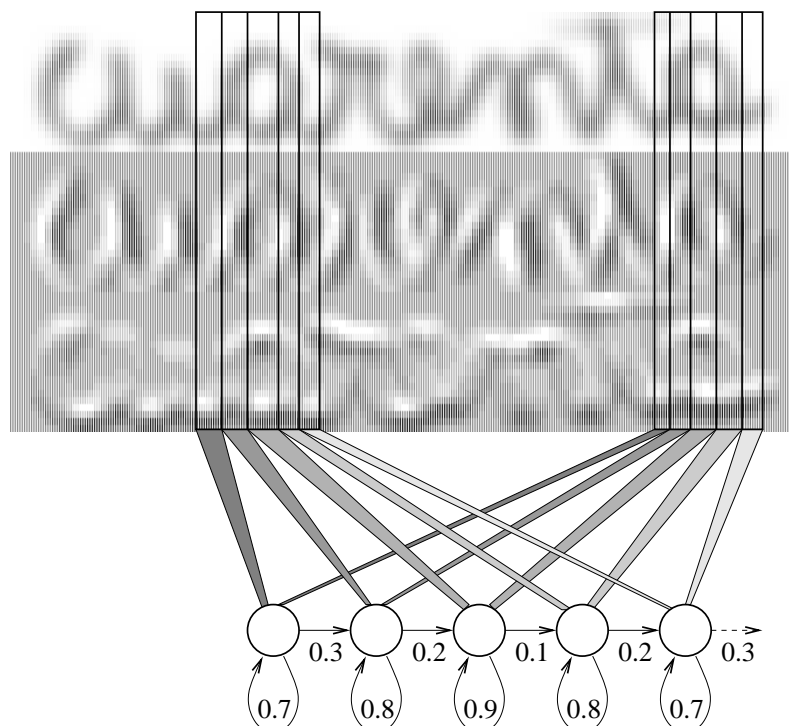
HMM Topology

- *Left-to-right* topology: state transitions to itself and to the next state.
- The required number of states to model a certain (or a set of) character depends on the underlying “horizontal variability”.
- The required number of densities in the mixture depends, along with many other factors, on the “vertical variability” associated with each state.
- Usually, Gaussians covariance matrices are used to be diagonal to reduce the number of training parameters.

For practical purposes:

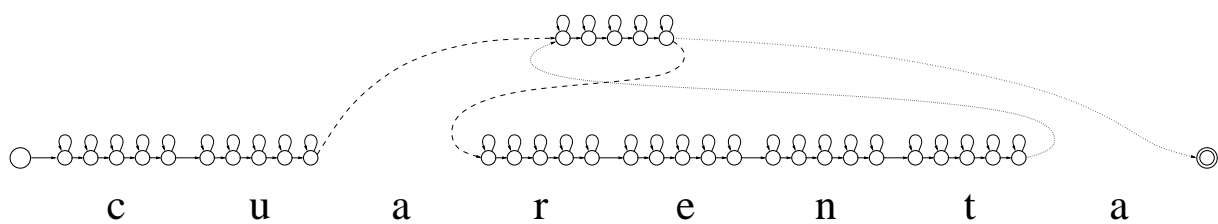
- in most applications, the number of states per each character HMM is set up to be the same.
- the number of Gaussians densities in the mixtures is usually the same for all HMMs states.

HMM morphological character modeling: Example

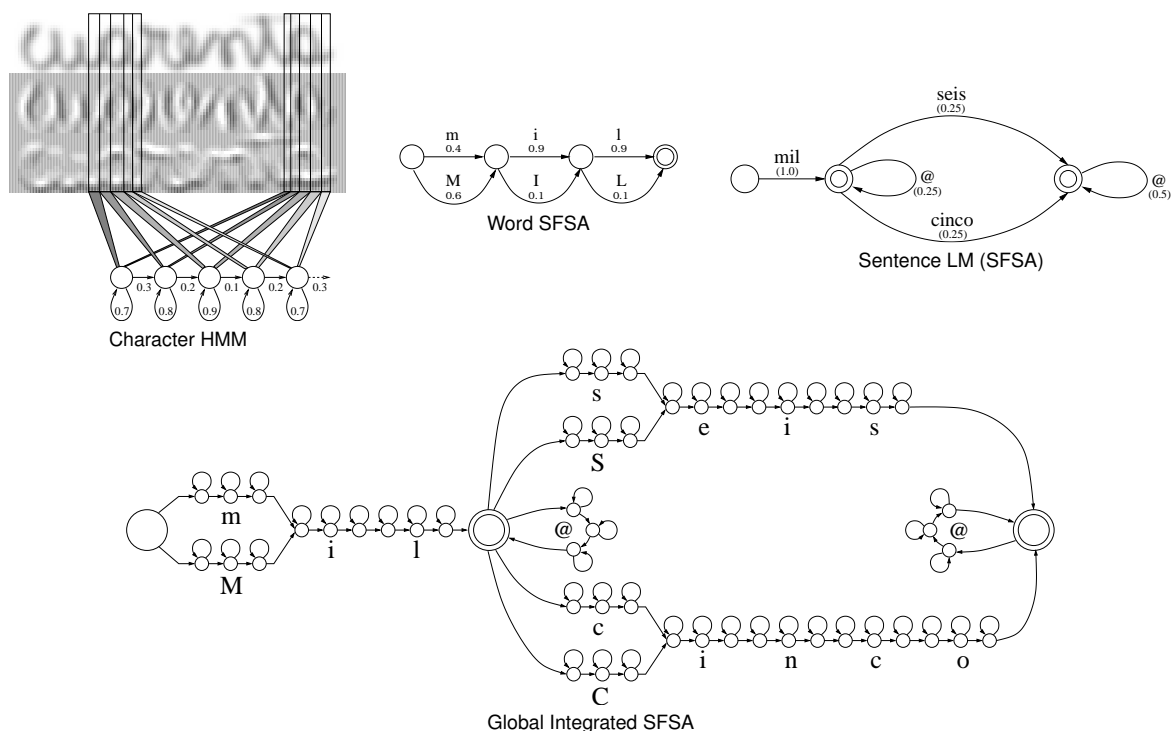


HTR: segmentation-free HMM training

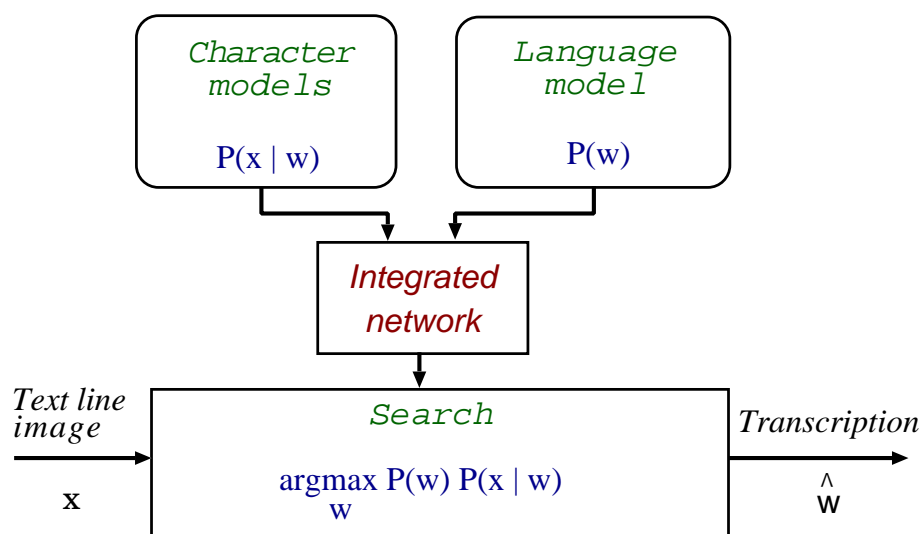
- *Integrated training* known as “embedded Baum-Welch training”
- HMM parameters trained from sentence or line image feature vector sequences and their associated transcriptions
- No prior segmentation into words or characters needed
- For each training sentence or line image, a long linear HMM is (dynamically) built by concatenating the HMMs of the successive characters of the image transcription:



Modeling using Stochastic Finite State Automaton (SFSA)



Integrated architecture for text image decoding



Search engine:

THE VITERBI ALGORITHM (+ beam search + ...)

Bibliography

- F. DRIRA. "Towards restoring historic documents degraded over time". In Proceedings of the Second International Conference on Document Image Analysis for Libraries (DIAL'06). Washington, DC, USA, IEEE Computer Society, pp.350-357, (2006).
- I. Bazzi, R. Schwartz, J. Makhoul. "An Omnifont Open-Vocabulary OCR System for English and Arabic". IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI) Vol.21 pp.495-504, 1999.
- A. Vinciarelli, S. Bengio, H. Bunke. "Offline Recognition of Unconstrained Handwritten Texts Using HMMs and Statistical Language Models". IEEE Trans. on PAMI, Vol.26, pp.709-720, 2004.
- A. H. Toselli, A. Juan, D. Keyzers, J. Gonzalez, I. Salvador, H. Ney, E. Vidal and F. Casacuberta. "Integrated Handwriting Recognition and Interpretation using Finite-State Models". Int. Journal of Pattern Recognition and Artificial Intell., 18(4):519-539, June 2004.
- M. Zimmermann, J.C. Chappelier and H. Bunke. "Off-line Grammar-Based Recognition of handwritten sentences". IEEE Trans. on Pattern Analysis and Machine Intelligence, 28(5):818-821, May 2006.
- L. Rabiner. "A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition". Proc. IEEE, 77:257-286, 1989.