ICFHR–2010 Tutorial:

Multimodal Computer Assisted Transcription of Handwriting Images

II – Computer-Assisted Transcription of Text Images (CATTI)

Alejandro H. Toselli & Moisés Pastor & 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

A BLANK PAGE

Tutorial Contents and Schedule

Index

Computer Assisted Transcription of Text Images (CATTI)

- Current cursive (off-line) HTR systems are far from being perfect and generally need human post-editing to check and correct the results of such systems
- In a computer-assisted, interactive framework, rather than full automation, the system aims to facilitate and speed up the human transcription task
- This framework, called “CATTI”, combines the efficiency of automatic handwriting recognition systems with the accuracy of the experts, leading to a cost-effective perfect transcription results
- CATTI can be properly formulated within the general *Interactive-Predictive Pattern Recognition* paradigm
- Performance measures for CATTI should aim at estimating human-effort, rather than error rate

ICFHR 2010: Multimodal Computer Assisted Transcription of Handwriting Images CATTI

A BLANK PAGE

A.H. Toselli, M. Pastor and V. Romero – ITI-UPV Page II.3

How does CATTI work?

- The HTR system proposes a full transcription of the input handwritten text line image.
- The human transcriptor (user) validates a prefix of the transcription which is error-free.
- The human enters a word (or words) to correct the erroneous text, producing a new prefix.
- The HTR suggests a suitable continuation to this prefix.
- This previous four steps are iterated until a final, perfect transcription is produced.

Every correction made by the user helps the system automatically avoid further errors in the suggested text.

CATTI operation example

	x							
STEP-0	p							
STEP-1	$\hat{s} \equiv \hat{w}$	**antiguas**	**cuidadelas**	**que**	**en**	**el Castillo**	**sus**	**llamadas**
	p'	**antigu**						
	κ	**os**						
	p	**antiguos**						
STEP-2	$\hat{s}$		**ciudadanos**	**que**	**en**	**el Castillo**	**sus**	**llamadas**
	p'	**antiguos**	**ciudadanos**	**que**	**en**			
	κ					**Castilla**		
	p	**antiguos**	**ciudadanos**	**que**	**en**	**Castilla**		
FINAL	$\hat{s}$						**se**	**llamaban**
	p'	**antiguos**	**ciudadanos**	**que**	**en**	**Castilla**	**se**	**llamaban**
	κ							**#**
	$p \equiv T$	**antiguos**	**ciudadanos**	**que**	**en**	**Castilla**	**se**	**llamaban**

Post-editing WER: 6/7 (86%)
Interactive WSR: 2/7 (29%, assuming a whole-word correction in step-1)
Estimated effort reduction: $1 - 29/86$ (66%).

Statistical framework for CATTI

Given a feature vector stream, x, a set of morphological, lexicon and language models, $\mathcal{M}$ and a *transcription prefix*, p, validated by the user in the previous step, obtain a proper completion (*suffix*) of p from which x can be produced with maximum likelihood; that is:

$$\hat{s} = \underset{s}{argmax}\, P_{\mathcal{M}}(s \mid x, p)$$

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

$$\hat{s} = \underset{s}{argmax}\, P(x \mid p, s) \cdot P(s \mid p)$$

In practice, a *Grammar Scale Factor* is generally used:

$$\hat{s} = \underset{s}{argmax}\, P(x \mid p, s)^{(1-\alpha)} \cdot P(s \mid p)^{\alpha}$$

Statistical framework for CATTI (cont.)

HTR main equation:

$$\hat{w} = \underset{w}{argmax}\, P(x \mid w) \cdot P(w)$$

CATTI main equation:

$$\hat{s} = \underset{s}{argmax}\, P(x \mid p, s) \cdot P(s \mid p)$$

The concatenation of p and s is the whole sentence, w. Therefore CATTI is very similar to the basic HTR. Two main differences:

- The maximization in CATTI must be carried out only over suffixes of p, rather than over whole sentences
- $P(s \mid p)$ in CATTI is interpreted as a kind of "dynamic" language model, conditioned by increasingly long prefixes, rather than the "static" HTR language model $P(w)$.

Statistical framework for CATTI (cont.)

Following the prefix-suffix assumption, x can be considered split into two fragments, x_1^b and x_{b+1}^m, where m is the length of x.

This allow us to marginalize $P(x \mid p, s)$ on the boundary point, b, leading to:

$$\hat{s} = \underset{s}{argmax} \sum_{1 \leq b \leq m} P(x, b \,|\, p, s) \cdot P(s \,|\, p)$$

Now (realistically) assuming that $P(x_1^b \mid p, s)$ does not depend on s and $P(x_{b+1}^m \mid p, s)$ does not depend on p:

$$\hat{s} \approx \underset{s}{argmax} \sum_{1 \leq b \leq m} P(x_1^b \,|\, p) \cdot P(x_{b+1}^m \,|\, s) \cdot P(s \,|\, p)$$

And approximating the sum by the dominating term:

$$\hat{s} \approx \underset{s}{argmax} \underset{1 \leq b \leq m}{max} P(x_1^b \,|\, p) \cdot P(x_{b+1}^m \,|\, s) \cdot P(s \,|\, p)$$

- $P(x_1^b \mid p), P(x_{b+1}^m \mid s)$: *conventional morphological word HMMs*
- $P(s \mid p)$: *prefix-conditioned Language Model*

CATTI Models

N-Gram Language Modeling:

Let $w = w_1^l$ be a full sentence hypothesis and $p = w_1^k$, $s = w_{k+1}^l$.

$$\begin{aligned} P(s \mid p) &= \frac{P(s, p)}{P(p)} = \frac{P(w)}{P(p)} \approx \frac{\prod_{i=1}^{l} P(w_i \mid w_{i-N+1}^{i-1})}{\prod_{j=1}^{k} P(w_j \mid w_{j-N+1}^{j-1})} \\ &= \prod_{i=k+1}^{l} P(w_i \mid w_{i-N+1}^{i-1}) \end{aligned}$$

The terms from $k+1$ to $k+N-1$ include dependences from the already known words w_{k-N+2}^k. The remaining terms are usual N-Grams; that is:

$$P(s \mid p) \approx \prod_{i=k+1}^{k+N-1} P(w_i \mid w_{i-N+1}^{i-1}) \cdot \prod_{i=k+N}^{l} P(w_i \mid w_{i-N+1}^{i-1})$$

CATTI Search

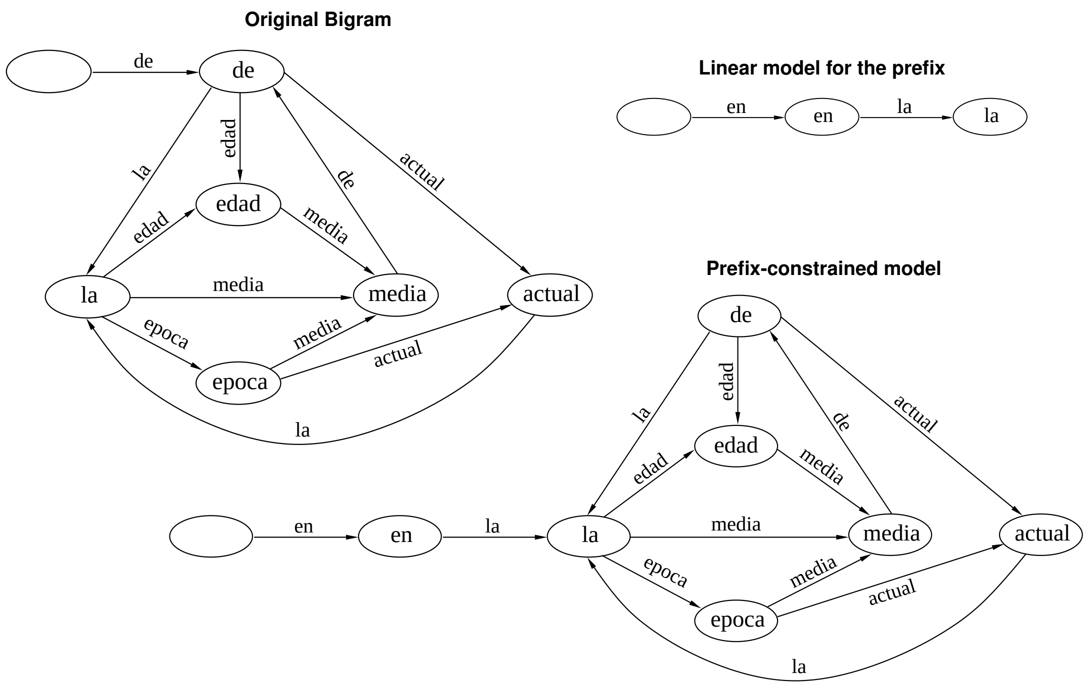
- In the first iteration the decoder generates a full transcription of x.
- Then, the user-validated prefix is used to generate a suitable continuation.
- The decoder matches the prefix p and then continue searching for the best suffix $\hat{s}$.
- Two possible implementations of the CATTI decoder are studied:
  - "Dynamic" Language Modeling (baseline):
    - ∗ On each step a linear LM that accounts for the fixed prefix is concatenated with a LM that accounts for $P(s \mid p)$.
    - ∗ Easily solved by the Viterbi algorithm
    - ∗ Computational cost that grows quadratically with the number of words
  - Faster approach based on WG derived from the initial Viterbi decoding:
    - ∗ Efficient, linear computational cost

CATTI "Dynamic" Language Modeling

Training samples

de la edad media
de edad media
de la epoca media
de la epoca actual
de la media
de la actual
de actual

Prefix = en la

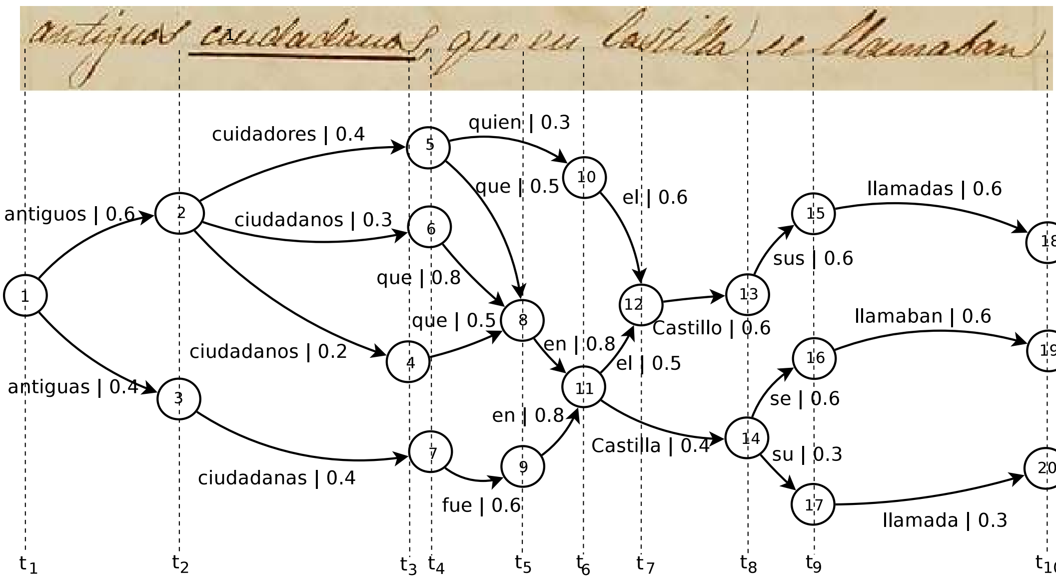


CATTI "dynamic language model" building. A *prefix-constrained model* is obtained by concatenating a *bigram* trained from the training samples to a *linear model* which accounts for the prefix "*en la*".

Word-graph based approach

- Efficient search techniques based on word-graphs are used
- A word graph represents the transcriptions with higher $Pr(w|x)$
- During the decoding process the system makes use of this word-graph in order to complete the prefixes accepted by the human transcriptor
- The system is able to interact with the human transcriptor in a time efficient way
- Drawback: some accuracy can be lost

Word-graph based approach:Example



Word-graph based approach

Let ϕ_w be a sequence of edges $e_1, e_2, ..., e_l$ such that $w = \omega(e_1), \omega(e_2), ..., \omega(e_l)$ and $d(w)$ the set of all the paths associated with w, the probability of the word sequence w given a word-graph and the word sequence with greatest probability are computed as:

$$P(w) = \sum_{\phi_w \in d(w)} \prod_{i=1}^{l} p(e_i) \qquad \hat{w} = \underset{w}{argmax} \sum_{\phi_w \in d(w)} \prod_{i=1}^{l} p(e_i)$$

We approximate it by means of the efficient Viterbi search algorithm:

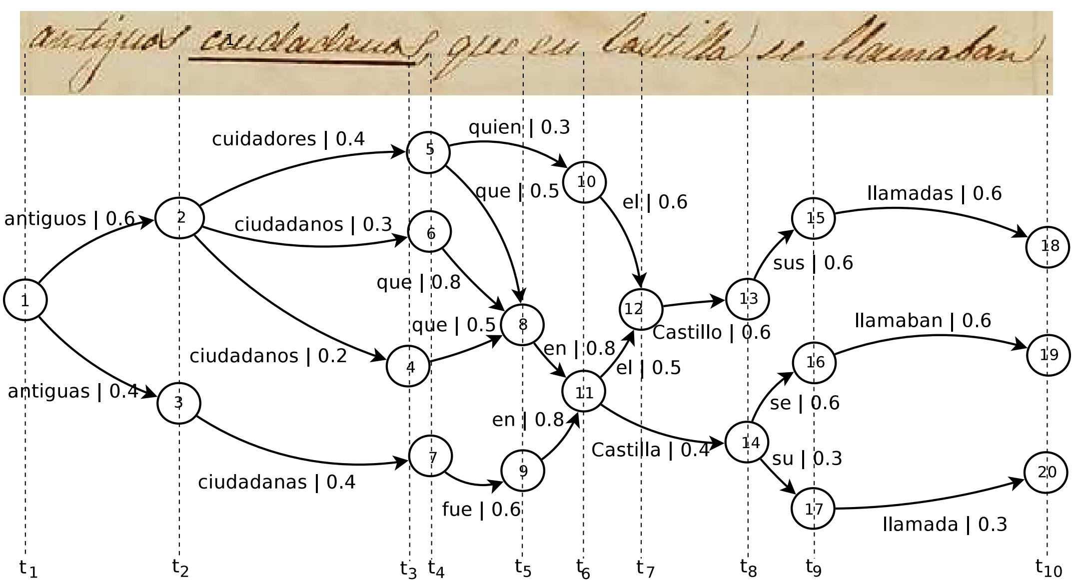
$$P(w) \approx \max_{\phi_w \in d(w)} \prod_{i=1}^{l} p(e_i) \qquad \hat{w} \approx \underset{w}{argmax} \max_{\phi_w \in d(w)} \prod_{i=1}^{l} p(e_i)$$

where the probability of an edge, $p(e)$, being $e = (i, j)$, is:

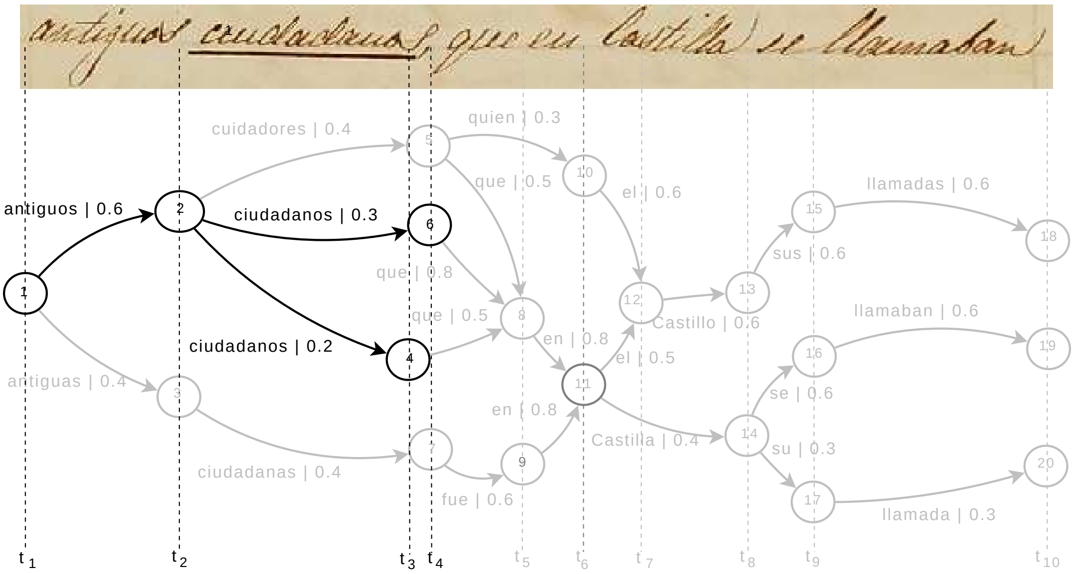
$$p(e) = P(x_{t(i)}^{t(j)} | \omega(e)) \cdot P(\omega(e))$$

$$\varphi(e) = \log P(x_{t(i)}^{t(j)} | \omega(e)) + \alpha \log P(\omega(e)) + \beta$$

Word-graph based approach:Example

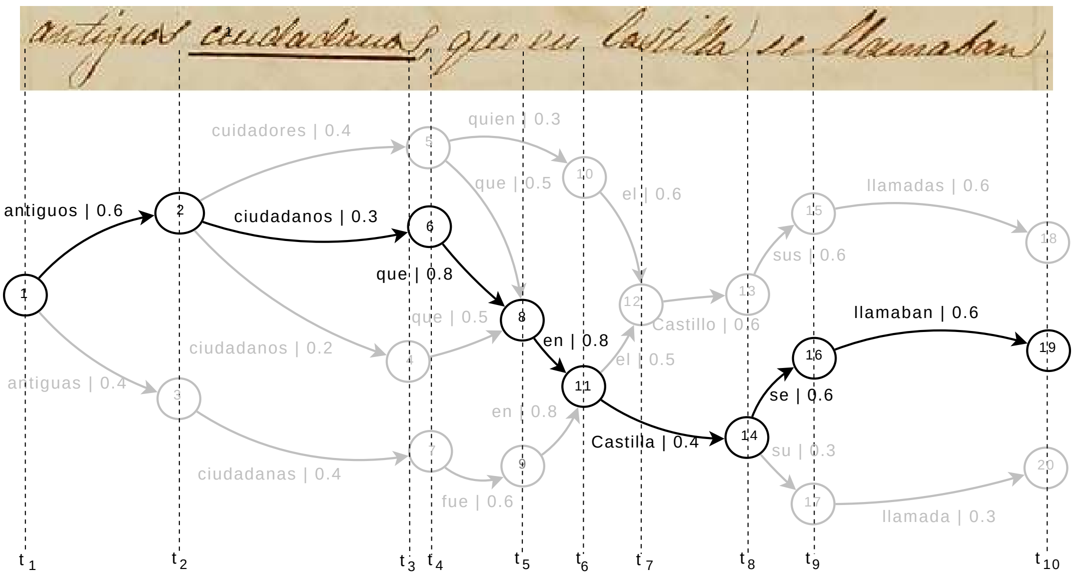


Word-graph based approach:Example



$p =$ antiguos ciudadanos

Word-graph based approach:Example



$p =$ antiguos ciudadanos

Word-graph based approach

- The decoder parses the validated prefix p over the WG defining a set of nodes Q_p corresponding to paths from the initial node whose associated word sequence is p
- The decoder continues searching for the suffix s from any of the nodes in Q_p that maximizes the posterior probability.

$$\hat{s} = \underset{s}{argmax} \max_{q \in Q_p} P(x_1^{t(q)} \mid p) \cdot P(x_{t(q)+1}^{M} \mid s) \cdot P(s \mid p)$$

- Problem: Some prefixes given by the user can not be exactly found in the word graph

Word-graph based approach: Error-correction parsing

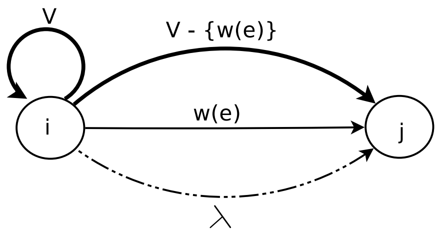
- Solution: not to use p but looking for the prefix p_e, from all the possibles prefix on the word graph, that best match the given prefix.
- Making the naive assumption that x and s do not depend of p and only depend of p_e:

$$(\hat{p}_e, \hat{s}) \approx \underset{p_e, s}{argmax} \max_{q \in Q} P(x_1^{t(q)} \mid p_e) \cdot P(x_{t(q)+1}^{M} \mid s) \cdot P(s \mid p_e) \cdot P(p_e \mid p)$$

where $P(p_e \mid p)$ measures the similarity between p_e and p and is modelled as an stochastic edit distance based on the lexical differences between the words

Word-graph based approach: Error-correction parsing

- $P(p_e \mid p)$ can be modelled in terms of probabistic error correcting parsing
- The WG is expanded with a set of edges representing the different edit operations.



  - The edge labelled with $\omega(e)$ corresponds to replacing the word $\omega(e)$ with itself.
  - Edges labelled with $V - \{\omega(e)\}$ models the substitution of $\omega(e)$ for another word.
  - The edge labelled with λ (empty symbol) models a deletion.
  - The group of edges labelled with V is for insertions.

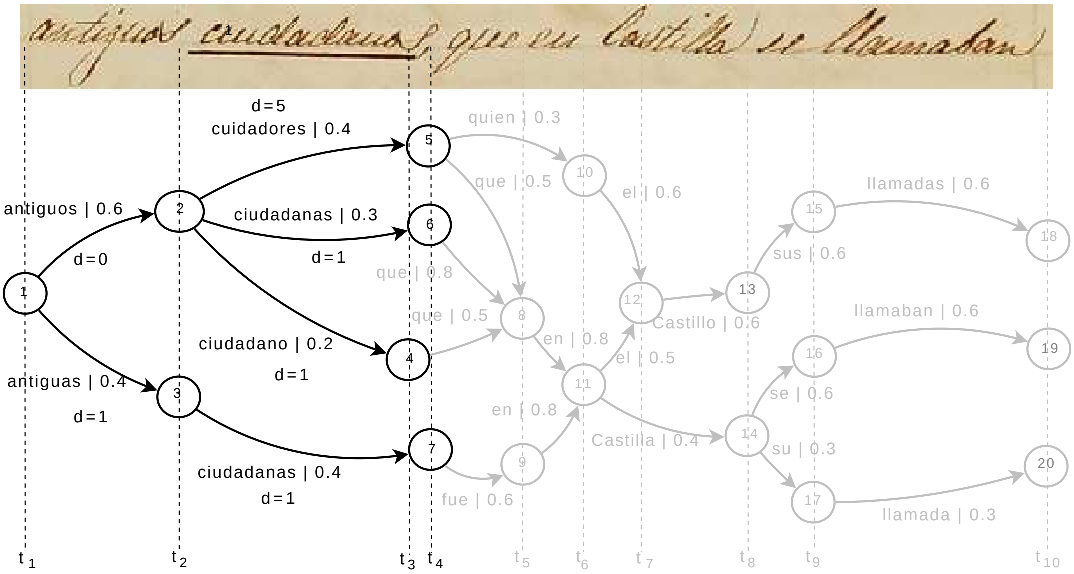
Word-graph based approach: Error-correction parsing

- The probabilities of the expanded edges are proportional to $\exp^{-d(v_1,v_2)}$, where $v_1, v_2 \in V \cup \lambda$ and $d(\cdot,\cdot)$ is the Levenstein distance between v_1 and v_2.
- Each edge must be represented using its start and end node and the word related with this edge, $e' = (i, j, v)$:

$$\varphi(i,j,v) = \begin{cases} \log P(x_{t(i)}^{t(j)} \mid \omega(e)) + \alpha \log P(\omega(e)) + \beta - \gamma d(\omega(e), v) & i \neq j, \\ & v \in V \cup \{\lambda\} \\ \beta - \gamma d(\lambda, v) & i = j \end{cases}$$

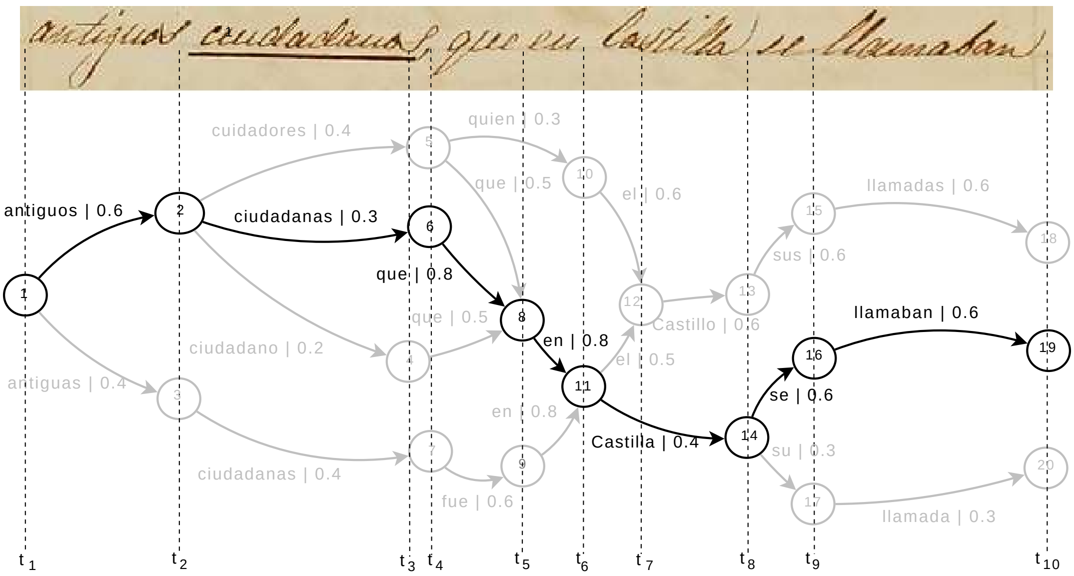
where e is the original edge between the nodes i and j and the parameter γ weights the penalization due to the number of different characters.

Word-graph based approach: Example



$p =$ antiguos ciudadanos

Word-graph based approach: Example



$p =$ antiguos ciudadanos

Increasing interaction ergonomy

- Making the interaction process easy is crucial.
- CATTI interaction: mouse-action to position the cursor + typing the correct word.
- Alternative interaction: the mouse-action directly triggers the system to propose a new suitable suffix.
- Mouse clicks do not involve extra human effort.
- The system changes the suffix by the next most probable suffix whose first word is different.
- Many explicit user corrections are avoided.

Increasing interaction ergonomy: Example

	x							
INTER-0	p							
INTER-1	$\hat{s}$	**antiguos**	**cuidadores**	**que**	**en**	**el Castillo**	**sus**	**llamadas**
	m		↑					
	p'	**antiguos**						
	- - - -	- - - -	- - - -	- - - -	- - - -	- - - -	- - - -	- - - -
	$\hat{s}$		**cortesanos**	**que**	**en**	**el Castillo**	**sus**	**llamadas**
	v		ciudadanos					
	p	**antiguos**	**ciudadanos**					
INTER-2	$\hat{s}$			**que**	**en**	**el Castillo**	**sus**	**llamadas**
	m					↑		
	p'	**antiguos**	**ciudadanos**	**que**	**en**			
FINAL	$\hat{s}$					**Castilla**	**se**	**llamaban**
	v							**#**
	$p \equiv T$	**antiguos**	***ciudadanos***	**que**	**en**	**Castilla**	**se**	**llamaban**

N. errors 1st hipotesis: 5 $WER = 71$

N. user interactions in CATTI: 2 $WSR = 29$

Number of user interactions in the new CATTI system: 1 $WSR = 14$

Increasing interaction ergonomy

- Scenarios:
  - Single-MA: the user only makes a MA when it is necessary to displace the cursor.
  - Performing one or several MA systematically before writing, even if the cursor is in the correct position.

- The decoder has to cope with the input image x, the validated prefix p' and the erroneous word e that follows the validated prefix:

$$\hat{s} = \underset{s}{argmax} \Pr(s \mid x, p', e) \approx \underset{s}{argmax}\, P(x \mid p', s, e) \cdot P(s \mid p', e)$$

  - $P(x \mid p', s, e)$: is modeled by standard HMMs following similar assumptions and developments carried out previously.
  - $P(s \mid p', e)$: can be provided by a language model constrained by the validated prefix p' and the erroneous word e that follows it.

Increasing interaction ergonomy: Adapting the language model

Let $p' = w_1^k$ be a validated prefix and $s = w_{k+1}^l$ be a possible suffix and considering that the wrong-recognized word e only affects to the first word of s, w_{k+1}:

$$P(s \mid p', e) \simeq P(w_{k+1} \mid w_{k+2-n}^k, e) \cdot \prod_{i=k+2}^{k+n-1} P(w_i \mid w_{i-n+1}^{i-1}) \cdot \prod_{i=k+n}^{l} P(w_i \mid w_{i-n+1}^{i-1})$$

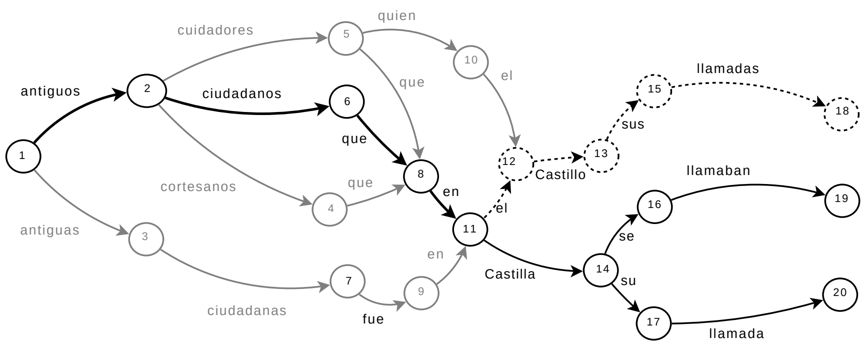
Taking into account that w_{k+1} has to be different to e:

$$P(w_{k+1} \mid w_{k+2-n}^k, e) = \frac{\bar{\delta}(w_{k+1}, e) \ \cdot P(w_{k+1} \mid w_{k+2-n}^k)}{\sum_{v'} \bar{\delta}(v', e) \ \cdot P(v' \mid w_{k+2-n}^k)}$$

where $\bar{\delta}(i, j)$ is 0 when $i = j$ and 1 otherwise

Increasing interaction ergonomy: Searching

- Using the Viterbi algorithm: A special language model can be build modifying the “Suffix Language Model“ in accordance with previous equations.
- An easier implementation can be carried out using word-graphs: The edge labelled with the word e after the prefix has been matches is deleted



CATTI at character level

- Only whole-word interactions has been considered until now on the CATTI system
- Keystroke interactions can allow for a more ergonomic and friendly interfaces
- As soon as the user introduces a new keystroke, the system proposes a new suitable continuation
- A timed system reaction scheme is also possible, where new predictions are computed only upon detecting a short period of user inactivity
- Autocompleting: The system looks for the most probable word that begins with the incomplete word that the user has validated

CATTI at character level

	x	
INTER-0	p	
INTER-1	$\hat{s} \equiv \hat{w}$	**antiguos cuidadores que en el Castillo sus llamadas**
	p'	**antiguos c**
	c	i
	p	**antiguos ci**
INTER-2	$\hat{s}$	**udadanos que en el Castillo sus llamadas**
	p'	**antiguos ciudadanos que en**
	c	C
	p	**antiguos ciudadanos que en C**
FINAL	$\hat{s}$	**astilla se llamaban**
	c	#
	$p \equiv T$	**antiguos ciudadanos que en Castilla se llamaban**

CATTI at character level

	x	
INTER-0	p	
INTER-1	$\hat{s} \equiv \hat{w}$	**antiguos cuidadores que en el Castillo sus llamadas**
	p'	**antiguos c**
	c	i
	p	**antiguos ci**
INTER-2	$\hat{s}$	**udadanos que en el Castillo sus llamadas**
	p'	**antiguos ciudadanos que en**
	c	C
	p	**antiguos ciudadanos que en C**
FINAL	$\hat{s}$	**astilla se llamaban**
	c	#
	$p \equiv T$	**antiguos ciudadanos que en Castilla se llamaban**

- Session 9: Off-line Recognition-III. November 18
  Character-level interaction in Computer-assisted Transcription of Text Images
  Verónica Romero, Alejandro H. Toselli and Enrique Vidal

Performance Measures for CATTI

- WORD ERROR RATE (WER):
  Minimum number of *non-interactive word* corrections (insertions, deletions and substitutions) needed to edit the system output into a (single) target reference

- WORD STROKE RATIO (WSR):
  Minimum number of word corrections that a (hypothetical) user would have to interactively make to achieve a given reference transcription, divided by the overall number of reference words.

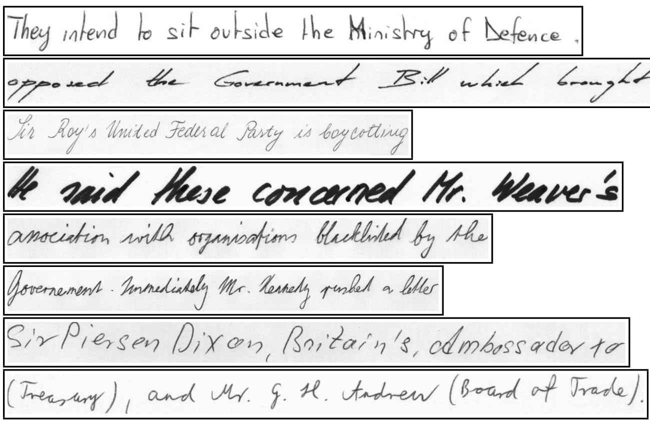
- WORD CLICK RATE (WCR):
  Number of additional mouse-clicks by word, divided by the total number of reference words.

The relative difference between WSR and WER estimates the human effort that CATTI would save, with respect to that of classical HTR followed by post-editing (EFR).

Corpora for HTR experiments: IAMDB

Handwritten texts from the Lancaster-Oslo/Bergen Corpus (LOB)

Publicly available: www.iam.unibe.ch/fki/databases

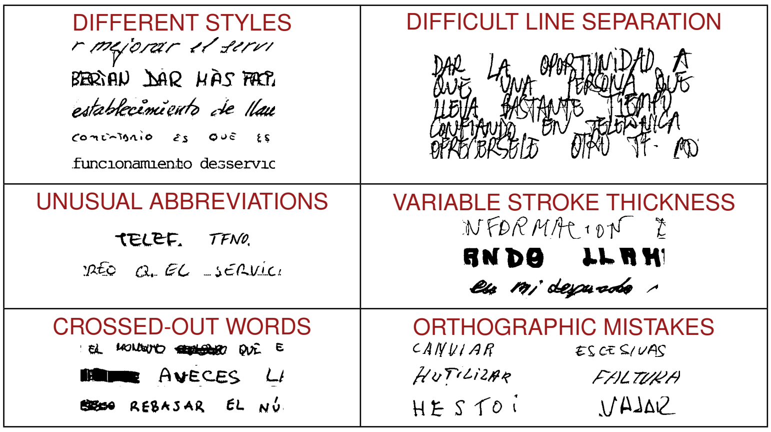


Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
writers	448	100	548	–	–	–
sentences	2 124	200	2 324	–	–	–
words	42 832	3 570	46 789	**8 017**	921	128
characters	216 774	20 726	237 500	78	0	2 779

LM traning data: approx. 10^6 **running words** from the LOB corpus

Corpora for HTR experiments: ODEC

Answers extracted from survey forms made for a telecommunication company

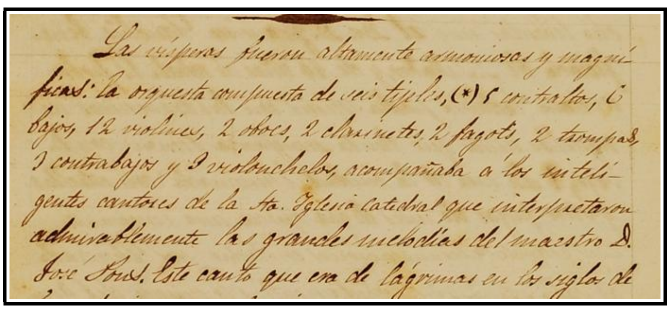


Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
writers/sentences	676	237	913	–	–	–
words	12 287	4 001	16 371	2 790	518	4.4
characters	64 666	21 533	86 199	80	0	808

ICFHR 2010: Multimodal Computer Assisted Transcription of Handwriting Images CATTI

Corpora for HTR experiments: “Cristo Salvador”(CS)

Single writer manuscript from the XIX century


Las vísperas fueron altamente armoniosas y magní-
ficas: la orquesta compuesta de seis tiples, (*) 5 contraltos, 6
bajos, 12 violines, 2 oboes, 2 clarinetes, 2 fagots, 2 trompas,
3 contrabajos y 3 violonchelos, acompañaba a los inteli-
gentes cantores de la Sta. Iglesia catedral que interpretaron
admirablemente las grandes melodías del maestro D.
José Pons. Este canto que era de lágrimas en los siglos de

Corpora for HTR experiments: CS partitions

CS *Page* partition

Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
pages	53	53	53	–	–	–
text lines	681	491	1 172	–	–	–
words	**6 432**	4 440	10 911	**2 623**	1 313	2.5
characters	36 699	25 460	62 159	78	0	470

CS *Book* partition

Number of:	Training	Test	Total	Lexicon	OOV	Tr. Ratio
pages	33	20	53	–	–	–
text lines	675	497	1 172	–	–	–
words	**6 222**	4 689	10 911	**2 536**	1 400	2.5
characters	35 845	26 314	62 159	78	0	460

Non-interactive HTR baseline results

WER obtained with *closed vocabulary* for different corpora: IAMDB, ODEC and CS (*book* and *page* partitions).

- No case distinction or diacritics; no punctuation marks.
- Character HMMs: 6 states, 64 Gaussian densities per state
- Language models: Bi-grams

Corpus		IAMDB	ODEC	CS-page	CS-book
Writters		many	many	1	1
HMMs	Characters	78	80	78	78
	Tr. Ratio	2 779	808	470	460
Lang. Model	Lexicon	8 017	2 790	2 623	2 536
	OOV	921	518	1 313	1 400
	Tr. Ratio	128	4.4	2.5	2.5
WER (%)		**25.3**	**22.9**	**28.5**	**33.5**

CATTI Results

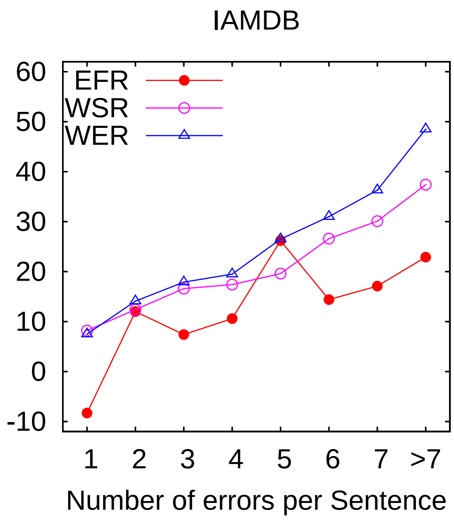
In all the experiments, only interaction at the *word level* is assumed; that is, each interaction step involves the correction of a *single, whole word* from the system-predicted suffix.

This allows proper comparisons of the *estimated user effort* needed for non-interactive post-editing (WER) versus interactive processing (WSR).

Performance of the baseline off-line HTR system (WER) and the CATTI system (WSR) for different tasks using the Viterbi-based approach. 2-gram language models in all the cases:

	IAMDB	ODEC	CS *page*	CS *book*
WER (%)	25.3	22.9	28.5	33.5
WSR (%)	21.1	18.9	26.9	32.1
EFR (%)	16.6	17.5	5.7	4.2

CATTI Results: impact of number of errors per sentence



Estimated effort reduction using the sentences with zero and one error (EFR*) compared with the EFR in all the sentences

Corpus	EFR	EFR*
ODEC	17.5	17.9
IAMDB	16.6	18.4
CS-page	5.7	6.9
CS-book	4.2	4.8

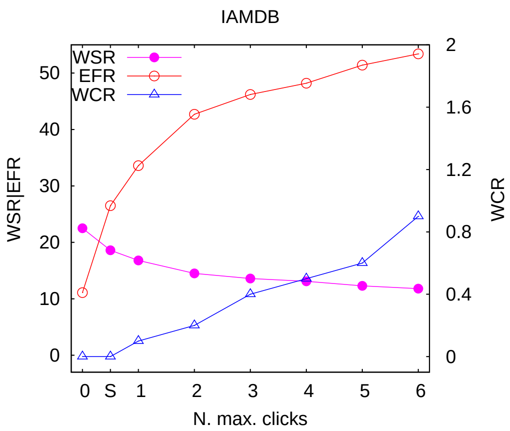
CATTI Results: Word-graph based approach

Performance of non-interactive off-line HTR and CATTI along with the relative difference between them, using all the test sentences (Word-graph EFR) and excluding the sentences with zero and one errors (EFR*).

Corpus	WER	WSR	EFR	EFR*
ODEC	22.9	21.5	6.1	6.8
IAMDB	25.3	22.5	11.1	12.9
CS-page	28.5	27.7	2.8	3.8
CS-book	33.5	32.3	3.6	4.1

- Given the efficient linear cost of WGs they are preferred altought the EFR is smaller than using the Viterbi-based approach.

CATTI Results: Using MA in the CATTI interaction process



- WCR: Number of additional mouse-clicks by word, divided by the total number of reference words
- Scenarios:
  - Single-MA: the user only makes a MA when it is necessary to displace the cursor.
  - Performing one or several MA systematically before writing, even if the cursor is in the correct position.

CATTI Results: Word-graph based approach

	Viterbi	Word-graph	
Corpus	EFR	EFR	EFR-S
ODEC	17.5	6.1	20.5
IAMDB	16.6	11.1	26.5
CS-book	4.2	3.6	15.2

- Given the efficient linear cost of WGs they are preferred altought the EFR is smaller than using the Viterbi-based approach.

Bibliography

- E. Vidal, F. Casacuberta, L. Rodríguez, J. Civera and C. Martínez. “Computer-assisted translation using speech recognition”. IEEE Trans. on Audio, Speech and Language Processing, 14(3):941-951, 2006.

- E. Vidal, L. Rodriguez, F. Casacuberta and I. García-Varea: “Interactive Pattern Recognition”. 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI-07), Volume 4892 of LNCS, pp.60-71. Brno, Czech Republic, June 2007.

- L. Rodriguez, F. Casacuberta, and E. Vidal. “Computer Assisted Transcription of Speech” Proc. of the third Iberian Conference on Pattern Recognition and Image Analysis, Volume 4477 of LNCS, pp.241-248, Girona (Spain), June 2007.

- A.H. Toselli, V. Romero, L. Rodríguez and E. Vidal. “Computer Assisted Transcription of Handwritten Text”. 9th Int. Conference on Document Analysis and Recognition (ICDAR 2007), pp.944-948. IEEE Computer Society, Curitiba, ParanÃ¡ (Brazil), September 2007.

- V. Romero, A.H. Toselli, L. Rodríguez and E. Vidal. “Computer Assisted Transcription for Ancient Text Images”. Int. Conference on Image Analysis and Recognition (ICIAR 2007), volume 4633 of LNCS, pp.1182-1193. Montreal (Canada), August 2007.