Improvements in the Computer Assisted Transcription System of Handwritten Text Images

Verónica Romero, Alejandro H. Toselli, Jorge Civera and Enrique Vidal

Instituto Tecnológico de Informática, Universidad Politécnica de Valencia
Camí de Vera s/n, 46022 Valencia, Spain

Abstract. To date, automatic handwriting recognition systems are far from being perfect. Therefore, once the full recognition process of a handwritten text image has finished, heavy human intervention is required in order to correct the results of such systems. As an alternative, an interactive framework has been presented in previous works. The results obtained in these works show that significant amounts of human effort can be saved. Here a new way to interact with this interactive system is proposed. Now, as soon as the user points to the next system error, the system proposes a new suitable continuation. This way, many explicit user corrections are avoided. Empirical results suggest that the new interaction method can lead to further improvements in user productivity.

1 Introduction

Many documents used every day include handwritten text and, in many cases, it would be interesting to recognize these text images automatically. To date, automatic handwriting recognition systems (HTR) have proven to be suitable for restricted applications with very limited vocabulary (reading of postal addresses or bank checks) or constrained handwriting (forms) achieving in these kinds of tasks relatively high recognition rates. However, in the case of unconstrained transcription applications (such as old manuscripts or spontaneous sentences), the current HTR technology typically only achieves results which do not met the quality requeriments of practical applications.

In these cases, to obtain high quality transcriptions it is necessary a *post editing* process, where a human transcriptor intervention is required to check and correct the mistakes made by the HTR system. This post-editing solution is rather uncomfortable and inefficient for the human corrector.

In previous works [1, 2], an *interactive* scenario called "Computer Assisted Transcription of Text Images" (CATTI) has been presented. In this scenario, the system uses the text image and a previously validated part (prefix) of its transcription to propose a suitable continuation of the transcription. Then the user finds *and correct* the next system error, thereby providing a longer prefix which the system uses to suggest a new, hopefully better continuation. The results obtained show that this system can save significant amounts of human effort.

In this work, a change in the CATTI user interaction is studied. Now, as soon as the user points to the next system error, the system proposes a *new* suitable continuation.

Romero V., H. Toselli A., Civera J. and Vidal E. (2008).
Improvements in the Computer Assisted Transcription System of Handwritten Text Images.
In *Proceedings of the 8th International Workshop on Pattern Recognition in Information Systems*, pages 103-112

This way, many explicit user corrections are avoided. To allow for an efficient implementation of this interaction improvement, a search strategy based on *word-graphs* is adopted. This allows a simple modification of standard *n-best* lists decoding to take into account the information provided by each error pointed by the user.

2 Foundations of CATTI

This section reviews the approach to CATTI presented in [1, 2]. The process starts when the HTR system proposes a full transcription $\hat{s}$ of a feature vectors sequence x, extracted from a handwritten text line image. Then, the human transcriptor (named user from now on) reads this transcription until he or she finds a mistake; i.e, he or she validates a prefix p' of the transcription which is error-free. Now, the user can enter a word, c, to correct the erroneous text that follows the validated prefix. This action produces a new prefix p (the previously validated prefix, p', followed by c). Then, the HTR system takes into account the new prefix to suggest a suitable continuation to this prefix (i.e., a new $\hat{s}$), thereby starting a new cycle. This process is repeated until a correct, full transcription t of x is accepted by the user.

2.1 Formal Framework

The traditional handwritten text recognition problem can be formulated as the problem of finding a most likely word sequence, $\hat{w}$, for a given handwritten sentence image represented by a feature vector sequence x, that is:

$$\hat{w} = \arg\max_{w} Pr(w|x) = \arg\max_{w} Pr(x|w) \cdot Pr(w) \ . \tag{1}$$

$Pr(x|w)$ is typically approximated by concatenated character Hidden Markov Models [3, 4]) and $Pr(w)$ is usually approximated by a n-gram word language model [3].

In the CATTI framework, in addition to the given feature sequence, x, a prefix p of the transcription is available and the HTR should try to complete this prefix by searching for a most likely suffix $\hat{s}$ as:

$$\hat{s} = \arg\max_{s} Pr(s|x,p) = \arg\max_{s} Pr(x|p,s) \cdot Pr(s|p) \ . \tag{2}$$

Eq. (2) is very similar to (1), being w the concatenation of p and s. The main difference is that now p is given. Therefore, the search must be performed over all possible suffixes s of p and the language model probability $Pr(s|p)$ must account for the words that can be written after the prefix p. Following assumptions and developments carried out in [1, 2] we can write:

$$\hat{s} \approx \arg\max_{s} \max_{1 \leq b \leq m} Pr(x_1^b|p) \cdot Pr(x_{b+1}^m|s) \cdot Pr(s|p) \ . \tag{3}$$

This optimization problem entails finding an optimal boundary point, $\hat{b}$, associated with the optimal suffix decoding, $\hat{s}$. That is, the signal x is split into two segments, $x_p = x_1^{\hat{b}}$ and $x_s = x_{\hat{b}+1}^m$ and the search for the best transcription suffix that completes a prefix p

can be performed just over segments of the signal corresponding to the possible suffixes. On the other hand, we can take advantage of the information coming from the prefix to tune the language model constraints modelled by $Pr(s|p)$.

As discussed in [5], the simplest way to deal with $Pr(s|p)$ is to adapt an n-gram language model to cope with the consolidated prefix. Assuming an n-gram model is used for $Pr(w)$, leads to the following decomposition:

$$Pr(s|p) \simeq \prod_{i=k+1}^{k+n-1} Pr(w_i|w_{i-n+1}^{i-1}) \cdot \prod_{i=k+n}^{l} Pr(w_i|w_{i-n+1}^{i-1}) \; , \tag{4}$$

where the consolidated prefix is $w_1^k = p$ and $w_{k+1}^l = s$ is a possible suffix. The first term of Eq. (4) accounts for the probability of the $n-1$ words of the suffix, whose probability is conditioned by words from the validated prefix, and the second one is the usual n-gram probability for the rest of the words in the suffix.

3 Searching

In previous works, a Viterbi-based approach was used to solve the search problem corresponding to Eq. 3 and 4. In this section, a more efficient approach is proposed.

As discussed in [5], we can explicitly rely on Eq. (3) to implement a decoding process in one step, as in conventional HTR systems. The decoder is forced to *match* the previously validated prefix p and then continue searching for a suffix $\hat{s}$ according to the constraints of Eq. (4). In the present work, more efficient search techniques based on *word-graphs* are used. These techniques are similar to those described in [6, 7] for Computer Assisted Translation and for multimodal speech post-editing.

A word graph represents the transcriptions with higher $Pr(w|x)$ of the given image text sentence. In this case, the word graph is just (a pruned version of) the Viterbi search trellis obtained when transcripting the whole image sentence. Fig. 2 shows an example of a word graph. During the CATTI process the system makes use of this word graph in order to complete the prefixes accepted by the human transcriptor.

A word graph can be represented as a weighted directed acyclic graph, where each edge (e) is labeled with a word (w_e) and a score ($score(e)$), and each node (n) is labeled with a point (horizontal position) of the handwritten image (t_n). For each edge, we denote S_e, E_e as its start node and end node respectively. The graph has a single start node, that points to the start of the text image, and a single end node.

The word labels of any path from the start node to the end node form a transcription hypothesis, whose probability is as given in the Eq. (1). In practice, the simple multiplication of $Pr(x|w)$ and $Pr(w)$ is modified to balance the absolute values of both probabilities. The most common modification is to use the *language weight* α and the *insertion penalty* β as it is used in speech recognition [8]. So, we can write:

$$\hat{w} = \arg\max_{w} \; \log Pr(x|w) + \alpha \log Pr(w) + m\beta \; , \tag{5}$$

where m is the word length of w. The score of an edge is computed considering the image between its start and end node points ($x_{t_{S_e}}^{t_{E_e}}$) and the given word at the edge (w_e):

$$score(e) = \log Pr(x_{t_{S_e}}^{t_{E_e}}|w_e) + \alpha \log Pr(w_e) + \beta \; . \tag{6}$$

As the word graph is a representation of a *subset* of the possible transcriptions for a source handwritten text image, it may happen that some prefixes given by the user can not be exactly found in the word graph. To circumvent this problem some heuristics need to be implemented. In this work, we modified the score associated to each edge in order to cope with the differences between the words in the prefix and the words in the path that best match the given prefix. This heuristic can be implemented as an error-correcting parsing dynamic programming algorithm. Moreover, this algorithm takes advantage of the incremental way in which the user prefix is generated, parsing only the new suffix appended by the user in the last interaction. The modification of the score of each edge is carried out by adding a weighted component that penalizes the score taking into account the number of different characters (c_d) between the word associated to the edge (w_e) and the word of the prefix that is being analized (w_p):

$$score(e) = \log Pr(x_{t_{S_e}}^{t_{E_e}} | w_e) + \alpha \log Pr(w_e) + \beta + \gamma \; c_d \; . \tag{7}$$

Note that if $w_e = w_p$ the number of different characters will be 0, therefore the equations (7) and (6) will become identical. In other case c_d will be the minimum edit distance between w_e and w_p. Sometimes, it can be better delete the word associated to an edge. In this case, we consider that the word associated to the edge is being substituted for the empty word, so c_d is the number of characters of w_e. Finally, at times it can be better to insert the word w_p instead of substituting it for other one. In this case a new edge is generated whose begin and end nodes are the same and whose score is:

$$score(e) = \beta + \gamma \; c_d \; , \tag{8}$$

where c_d is the number of characters of w_p. The parameter γ weights the penalization due to the number of different characters. Its value has to be greater than 0 because, otherwise, we will be encouraging paths which are more different from the given prefix.

The computational cost of this approach is much lower than use the naïve Viterbi adaptation we had used in previous works, because in the Viterbi adaptation the computational cost grows quadratically with the number of words of each sentence. Therefore, using word-graph techniques the system is able to interact with the human transcriptor in a time efficient way. However, a drawback of this implementation is that some accuracy can be lost.

4 Improvements in the CATTI Interaction Process

In CATTI applications the user is repeatedly interacting with the system. Hence, making the interaction process easy is crucial for the success of the system. As it is shown in the section 2, the interaction in the conventional CATTI consists in a mouse-click (or equivalent pointer-positioning keystrokes) to validate the longest prefix which is error-free, followed by typing a word to correct the erroneous text that follows the validated prefix. In this section, a more effective way to interact with the system is presented. Now, the mouse-click which the user makes to mark a mistake directly triggers the system to propose a new suitable suffix.

	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*	
	c		**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*	
	c								#
	$p \equiv t$	antiguos	<u>**ciudadanos**</u>	que	en	Castilla	se	llamaban	

Fig. 1. Example of CATTI operation. Starting with an initial recognized hypothesis $\hat{s}$, the user validates its longest well-recognized prefix p', making a mouse-click (m), and the system emits a new recognized hypothesis $\hat{s}$. As the new hypothesis does not correct the mistake the user types the corrects word c, generating a new validated prefix p (c concatenated to p'). Taking into account the new prefix the system suggests a new hypothesis $\hat{s}$ starting a new cycle. Now, the user validates the longest prefix p' with is error-free. The system takes into account the new prefix p' to propose a new suffix $\hat{s}$ one more time. As the new hypothesis corrects the erroneous word a new cycle start. This process is repeated until the final error-free transcription t is obtained. Underlined boldface word in the final transcription is the only one which was corrected by user. Note that in the iteration 1 it is needed a mouse-click to validate the longest prefix that is error-free and then, to type the correct word. However, the iteration 2 only needs a mouse-click.

In fig. 1 we can see an example of the CATTI process with the new interaction mode. As in the conventional CATTI, the process starts when the HTR system proposes a full transcription $\hat{s}$ of the input image x. Then, the user reads this prediction until a transcription error is found (e) and makes a mouse-click (m) to position the cursor at this point. This way, the user validates an error-free transcription prefix p'. Now, before the user introduces a word to correct the erroneous one, the HTR system, taking into account the new prefix and the wrong word that follows the validated prefix, suggests a suitable continuation to this prefix (i.e., a new $\hat{s}$). If the new $\hat{s}$ corrects the erroneous word (e) a new cycle starts. However, if the new $\hat{s}$ has an error in the same position that the previous one, the user can enter a word, c, to correct the erroneous text e. This action produces a new prefix p (the previously validated prefix, p', followed by c). Then, the HTR system takes into account the new prefix to suggest a new suffix and a new cycle starts. This process is repeated until a correct transcription of x is accepted by the user.

It is worth noting that in the example shown in fig. 1, without interaction, a user should have to correct about *six* errors from the original recognized hypothesis. If the conventional CATTI is used the user only has to correct *two* words. However, with the new interaction only one user-correction is necessary to get the final error-free transcription. Note that the mouse-click m that the user makes to validate the prefix p' does not involve extra human effort, because it is the same action that the user should make in the conventional CATTI to position the cursor before typing the correct word.

This new kind of interaction needs not be restricted to a single pointer-positioning mouse-click. If the system reaction to this mouse-click is not satisfactory (i.e., it does

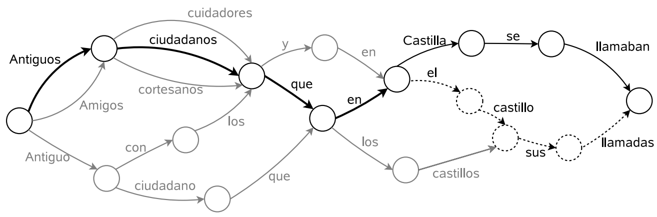

Fig. 2. Example of word-graph generated after the user validates the prefix "antiguos ciudadanos que en". The edge corresponding to the wrong-recognized word "el" was disabled.

not correct the error pointed to), the user may continue clicking and the system can react to each click by displaying the next suffix (ordered by posterior probability) which does not start with the already seen wrong words. It should be noted, however, that these additional multiple clicks do involve extra user (hypothesis pondering) effort.

Since we have already dealt, in the section 2, with the problem of finding a suitable suffix $\hat{s}$ when the user validate a prefix p' and introduce a correct word c, we focus now on the problem in which the user only makes a mouse-click. In this case the decoder has to cope with the input image x, the validated prefix p' and the erroneous word that follows the validated prefix e, in order to search for a transcription suffix $\hat{s}$:

$$\hat{s} = \arg\max_{s} Pr(s|x, p', e) = \arg\max_{s} Pr(x|p', s, e) \cdot Pr(s|p', e) \ . \tag{9}$$

Similar assumptions and developments followed in section 2 can be carried out to model $Pr(x|p', s, e)$. On the other hand, $Pr(s|p', e)$ can be provided by a language model constrained by the validated prefix p' and by the erroneous word that follows it.

4.1 Language Model and Search

$Pr(s|p', e)$ can be approached by adapting an n-gram language model so as to cope with the validated prefix p' and with the erroneous word that follows it e. The language model presented in section 2 would provide a model for the probability $Pr(s|p')$, but now the first word of s is conditioned by e. Therefore, some changes are needed.

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

$$Pr(s|p', e) \simeq Pr(w_{k+1}|w_{k+2-n}^k, e) \cdot \prod_{i=k+2}^{k+n-1} Pr(w_i|w_{i-n+1}^{i-1}) \cdot \prod_{i=k+n}^{l} Pr(w_i|w_{i-n+1}^{i-1}) \ . \tag{10}$$

Now, taking into account that the first word of the possible suffix w_{k+1} has to be different to the erroneous word e, $Pr(w_{k+1}|w_{k+2-n}^k, e)$ can be formulated as follows:

$$Pr(w_{k+1}|w_{k+2-n}^k, e) = \frac{\bar{\delta}(w_{k+1}, e) \ \cdot Pr(w_{k+1}|w_{k+2-n}^k)}{\sum_{w'} \bar{\delta}(w', e) \ \cdot Pr(w'|w_{k+2-n}^k)} \ , \tag{11}$$

where $\bar{\delta}(i, j)$ is 0 when $i = j$ and 1 otherwise.
As in the conventional CATTI, the decoder can be implemented using a word-graph.

The restrictions entailed by the modelling (11) can be easily implemented by deleting the edge labeled with the word e after the prefix has been matched. An example is shown in fig. 2. This example assumes the user has validated the prefix “antiguos ciudadanos que en” and the wrong-recognized word was “el”. Hence, the new word-graph has the edge labeled with the word “el” disabled.

5 HTR System Overview

The HTR system used here follows a classical architecture composed of three modules: preprocessing, feature extraction and recognition (see [9]).

The following steps take place in the preprocessing module: first, the skew of each page is corrected. Then, conventional noise reduction method is applied on the whole document image, whose output is then fed to the text line extraction process which divides it into separate text lines images. Finally, slant correction and size normalization are applied on each separate line. More detailed description can be found in [9, 10].

As our HTR system is based on Hidden Markov Models (HMMs), each preprocessed line image is represented as a sequence of feature vectors. To do this, the feature extraction module applies a grid to divide the text line image into $N \times M$ squared cells. In this work, N and M are chosen empirically. From each cell, three features are calculated: normalized gray level, horizontal gray level derivative and vertical gray level derivative. The way these three features are determined is described in [9]. Columns of cells or *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. Hence, at the end of this process, a sequence of M $3 \times N$-dimensional feature vectors is obtained.

The characters are modeled by continuous density left-to-right HMMs with 6 states and 64 Gaussian mixture components per state. Gaussians mixture serves as a probabilistic law to model the emission of feature vectors of each HMM state. The optimum number of HMM states and Gaussian densities per estate were tuned empirically.

Each lexical word is modelled by a stochastic finite-state automaton (SFS), which represents all possible concatenations of individual characters to compose the word. On the other hand, according to section 2, text line sentences are modelled using bi-grams, with Kneser-Ney back-off smoothing [11] and estimated directly from the training transcriptions of the text line images.

6 Experimental Results

In order to test the effectiveness of the new way to interact with the CATTI system different experiments were carried out. The corpora used, the different measures and the obtained experimental results are explained in the following subsections.

6.1 Corpora

Two different corpora have been used in our experiments. The first one, called ODEC, is a corpus based on a realistic application: transcriptions of handwritten answers extracted from survey forms made for a telecommunication company[1]. These answers were written by a heterogeneous group of people, without any explicit or formal restriction. So, paragraphs become very variable and noisy. More information about this corpus can be found in [12]. The relevant features of this corpus are shown in table 1.

Table 1. Basic statistics of the databases ODEC and CS.

	ODEC				CS			
Number of:	Training	Test	Total	Lexicon	Training	Test	Total	Lexicon
Phrases	676	237	913	–	681	491	1,172	–
Words	12,287	4,084	16,371	3,308	6,432	4,479	10,911	3,408
Characters	64,666	21,533	86,199	80	36,699	25,460	62,159	78

The second corpus was compiled from the legacy handwriting document from the nineteenth century identified as "*Cristo-Salvador*" (CS), which was kindly provided by the *Biblioteca Valenciana Digital* (BIVALDI)[2]. This corpus is composed of 53 text page images, written by only one writer. As it has been explained in section 5, the page images have been preprocessed and divided into lines, resulting in a data-set of 1,172 text line images. A summary of relevant features of this partitions is shown in table 1. The partition used here corresponds with the partition called "*soft*" in [2].

6.2 Assessment Measures

Different evaluation measures have been adopted. On the one hand, the quality of the transcription without any system-user interactivity is given by the well known *word error rate* (WER). On the other hand, *the word stroke ratio* (WSR) can be defined as the number of (word level) user interactions that are necessary to produce correct transcriptions using the CATTI system, divided by the total number of reference words. Finally, the *word click rate* (WCR) can be defined as the number of additional mouse-clicks by word that the user has to do using the new interaction with respect to using the conventional CATTI system, also relative to the total number of words in the correct transcription. In the experiments presented here real user interaction is simulated by using the given reference transcriptions of the text images. Therefore, results should be understood as estimates of expected real user effort.

The relative difference between WER and WSR (called Effort-Reduction) gives us an estimation of the reduction in human effort achieved by using CATTI with respect to using a conventional HTR system followed by human postediting.

6.3 Results

Table 2 shows the results obtained with the two corpora explained previously. In the first part of the table we can see an estimation of the reduction in human effort (E-R) achieved by using the conventional CATTI system with respect to the classic HTR post editing. In the second part, the results obtained with the new single-click interaction mode (explained in section 4) are shown.

[1] Data kindly provided by ODEC, S.A. (`www.odec.es`)
[2] http://bv2.gva.es

It is important to notice that some of the results in table 2 do not correspond with those reported in ([1, 2]). The differences are due to variations in the feature extraction process and on the implementation of this system. In previous works, a Viterbi-based approach was used while in this work word graphs search is used.

Table 2. Results obtained with the corpora ODEC and CS using the conventional CATTI (top) and the new kind of single-click interaction (bottom).

	ODEC	Cristo-Salvador
WER (%)	25.3	33.9
WSR (%)	22.7	32.5
Estimated E-R (%)	10.3	4.1
WSR (%)	19.8	27.8
Estimated E-R (%)	21.7	18.0

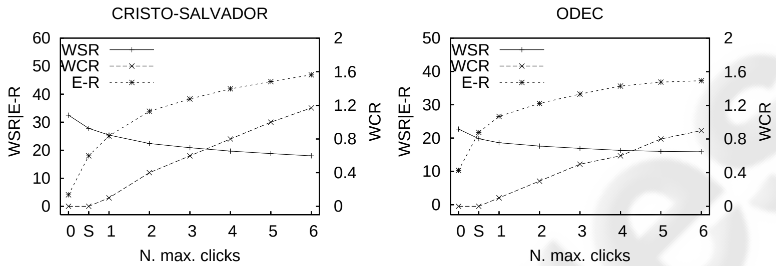

Fig. 3. WSR, Effort-Reduction (E-R) and WCR as a function of the maximal number of mouse-clicks allowed by the user before writing the correct word. The first point (0) correspond to the conventional CATTI, and the point S correspond to the single-click interaction discused in section 4.

According to table 2, the estimated human effort to produce error-free transcription using the new kind of interaction is significantly reduced with respect to using a conventional HTR system or the conventional CATTI. In the ODEC task, the new interaction mode can save about 22% of the overall effort, whereas the conventional CATTI would only save 10.3%. In the CS corpus, the reduction achieved is about 18%, instead of 4% obtained with the conventional CATTI.

Fig. 3 shows the WSR, the Effort-Reduction (E-R) and the WCR as a function of the maximal number of mouse-clicks allowed by the user before writing the correct word. The first point (0) corresponds to the results of the conventional CATTI, and the point "S" corresponds to the the single-click interaction considered in the previous table. A good trade-off is obtained when the maximum number of clicks is around 3, because a significant amount of expected human effort is saved with a fairly low number of extra clicks per word.

7 Remarks and Conclusions

In this paper, we have proposed a new way to interact with the CATTI systems presented in previous works. In conventional CATTI the user finds and corrects a first error and thereby validates an error-free transcription prefix which is used by the system to propose a hopefully better transcription continuation. Now, the mouse-click with which the user implicitly indicates the point where an error has occurred is used by the system to attempt to correct the error pointed to. It is worth noting that alternative (n-best) suffixes could also be obtained with the conventional CATTI system. However, by considering the rejected words to propose the alternative suffixes, the interaction methods here studied are more effective and (hopefully) more comfortable for the user. Moreover, using the new single-click interaction method, a second alternative suffix is obtained without extra human effort. A simple implementation of this system using word-graphs has been described and some experiments have been carried out.

In spite of the extreme difficulty of the corpora used in the experiments, the obtained results suggest that this new kind of interaction can speed-up, facilitate and save significant amounts of human effort in the handwritten text transcription process.

Acknowledgements

Work supported by the EC (FEDER), the Spanish MEC under the MIPRCV "Consolider Ingenio 2010" research programme (CSD2007-00018), the iTransDoc research project (TIN2006-15694-CO2-01), and by the Universitat Politècnica de València (FPI fellowship 2006-04)

References

1. A. H., V. Romero, L.R., Vidal, E.: Computer assisted transcription of handwritten text. In: Proc. of ICDAR 2007, Paraná (Brazil), (IEEE Computer Society) 944–948
2. V. Romero, A. H., L.R., Vidal, E.: Computer assisted transcription for ancient text images. In Proc. of ICIAR 2007 Vol. 4633 (2007) 1182–1193
3. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press (1998)
4. Rabiner, L.: A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. Proc. IEEE 77 (1989) 257–286
5. Rodriguez, L., Casacuberta, F., Vidal, E.: Computer Assisted Speech Transcription. In: Proc. of ibPRIA 2007. LNCS, Girona (Spain) (2007)
6. Barrachina, S., et al.: Statistical approaches to computer-assited translation. In: Computational Linguistic. (2006) I
7. Liu, P., Soong, F.: Word graph based speech recognition error correction by handwriting input. Proc. of the 8th Int. Conf. on Multimodal interfaces (2006) 339–346
8. A. Ogawa, K.T., Itakura, F.: Balancing acoustic and linguistic probabilites. Proc. IEEE Conf. Acoustics, Speech, and Signal Processing (1998) 181–184
9. Toselli, A.H., et al.: Integrated Handwriting Recognition and Interpretation using Finite-State Models. Int. Journal of Pattern Recognition and Artificial Intelligence 18 (2004) 519–539
10. Romero, V., Pastor, M., Toselli, A.H., Vidal, E.: Criteria for handwritten off-line text size normalization. In: Proc. of VIIP 06, Palma de Mallorca, Spain (2006)
11. Kneser, R., Ney, H.: Improved backing-off for n-gram language modeling. In Proc. of ICASSP 1995 **1** (1995) 181–184
12. Toselli, A.H., Juan, A., Vidal, E.: Spontaneous Handwriting Recognition and Classification. In: Proc. of ICPR 2004. Volume 1., Cambridge, United Kingdom (2004) 433–436