

Document downloaded from:

<http://hdl.handle.net/10251/39582>

This paper must be cited as:

Martín-Albo Simón, D.; Romero Gómez, V.; Toselli ., AH.; Vidal, E. (2011). Character-level interaction in multimodal computer-assisted transcription of text images. En Pattern Recognition and Image Analysis. Springer Verlag (Germany). 684-691. doi:10.1007/978-3-642-21257-4.



The final publication is available at

[http://link.springer.com/chapter/10.1007/978-3-642-21257-4\\_85](http://link.springer.com/chapter/10.1007/978-3-642-21257-4_85)

Copyright Springer Verlag (Germany)

# Character-level Interaction in Multimodal Computer-Assisted Transcription of Text Images

Daniel Martín-Albo, Verónica Romero, Alejandro H. Toselli, and Enrique Vidal \*

Instituto Tecnológico de Informática, Universidad Politécnica de Valencia, Spain  
{dmartinalbo, vromero, ahector, evidal}@iti.upv.es

**Abstract.** To date, automatic handwriting text recognition systems are far from being perfect and heavy human intervention is often required to check and correct the results of such systems. As an alternative, an interactive framework that integrates the human knowledge into the transcription process has been presented in previous works. In this work, multimodal interaction at character-level is studied. Until now, multimodal interaction had been studied only at whole-word level. However, character-level pen-stroke interactions may lead to more ergonomic and friendly interfaces. Empirical tests show that this approach can save significant amounts of user effort with respect to both fully manual transcription and non-interactive post-editing correction.

## 1 Introduction

At present time, the use of automatic handwritten text recognition systems (HTR) for the transcription of manuscript document images is far from being useful, mainly because of the unrestricted vocabulary and/or handwriting styles involved in such documents. Typically, the automatic transcriptions obtained by these HTR systems need a heavy human *post-editing* process in order to obtain transcriptions of standard quality. In practice, such a *post-editing* solution becomes rather inefficient, expensive and hardly acceptable by professional transcribers.

In previous works [7, 5], a more effective, *interactive* on-line approach was presented. This approach, called “Computer Assisted Transcription of Handwritten Text Images” (CATTI), combines the accuracy ensured by the human transcriber with the efficiency of the HTR systems to obtain final perfect transcriptions. Empirical results show that the use of CATTI systems can save a substantial quantity of human effort with respect to both pure manual transcriptions and post-editing.

So far, human corrective feedback for CATTI has been studied at two different levels: a) whole-word interactions (both typed and handwritten using an e-pen interface [7]) and b) (typed) character-level corrections [5]. According to the results of these works, keystroke corrections can save a significant quantity of human effort with respect to whole-word corrections, while multimodal, e-pen interaction seems more ergonomic for human transcribers, which is a key point in the design of friendly and usable user interfaces.

In this work, we focus on character level interaction using the more ergonomic *e-pen handwriting* modality, which is perhaps the most natural way to provide the required feedback in CATTI systems. It is important to note, however, that the use of this

---

\* Work supported by the Spanish Government (MICINN and “Plan E”) under the MITTRAL (TIN2009-14633-C03-01) research project and under the research programme Consolider Ingenio 2010: MIPRCV (CSD2007-00018), and by the Generalitat Valenciana under grant Prometeo/2009/014.

kind of non-deterministic feedback typically increases the overall interaction cost in order to correct the possible feedback decoding errors. Nevertheless, by using informations derived from the interaction process, we will show how the decoding accuracy can be significantly improved over using a plain e-pen handwriting recognizer which can not take advantage of the interaction context.

## 2 CATTI Overview

In the original CATTI framework, the human transcriber (named *user* from now on) is directly involved in the transcription process since he is responsible of validating and/or correcting the HTR outputs. The process starts when the HTR system proposes a full transcription of a feature vector sequence  $x$ , extracted from a handwritten text line image. The user validates an initial part of this transcription,  $p'$ , which is error-free and introduces a correct word,  $v$ , thereby producing correct transcription *prefix*,  $p = p'v$ . Then, the HTR system takes into account the available information to suggest a new suitable continuation *suffix*,  $s$ . This process is repeated until a full correct transcription of  $x$  is accepted by the user [7].

At each step of this process, both the image representation,  $x$ , and a correct transcription prefix  $p$  are available and the HTR system should try to complete this prefix by searching for the most likely suffix  $\hat{s}$  as:

$$\hat{s} = \arg \max_s P(s | x, p) = \arg \max_s P(x | p, s) \cdot P(s | p) \quad (1)$$

Since the concatenation of  $p$  and  $s$  constitutes a full transcription hypothesis,  $P(x | p, s)$  can be approximated by concatenated character Hidden Markov Models (HMMs) [2, 4] as in conventional HTR. On the other hand,  $P(s | p)$  is usually approximated by dynamically modifying a  $n$ -gram in order to cope with the increasingly consolidated prefixes [7]. Let  $p = p_1^k$  be a consolidated prefix and  $s = s_1^l$  a possible suffix:

$$P(s | p) \simeq \prod_{j=1}^{n-1} P(s_j | p_{k-n+1+j}^k, s_1^{j-1}) \cdot \prod_{j=n}^l P(s_j | s_{j-n+1}^{j-1}) \quad (2)$$

In order to make the system more ergonomic and friendly to the user, interaction based on characters (rather than full words) has been studied in [5] with encouraging results. Now, as soon as the user types a new keystroke (character), the system proposes a suitable continuation following the same process described above. As the user operates now at the character level, the last word of the prefix may be incomplete. In order to *autocomplete* this last word, it is assumed that the prefix  $p$  is divided into two parts: the fragment of the prefix formed by complete words ( $p''$ ) and the last incomplete word of the prefix ( $v_p$ ). In this case the HTR decoder has to take into account  $x$ ,  $p''$  and  $v_p$ , in order to search for a transcription suffix  $\hat{s}$ , whose first part is the continuation of  $v_p$ :

$$\hat{s} = \arg \max_s P(s | x, p'', v_p) = \arg \max_s P(x | p'', v_p, s) \cdot P(s | p'', v_p) \quad (3)$$

Again, the concatenation of  $p''$ ,  $v_p$  and  $s$  constitutes a full transcription hypothesis and  $P(x | p'', v_p, s)$ , can be modelled with HMMs. On the other hand, to model

$P(s \mid p'', v_p)$  we assume that the suffix  $s$  is divided into two fragments:  $v_s$  and  $s''$ .  $v_s$  is the first part of the suffix that corresponds with the final part of the incomplete word of the prefix, i.e.,  $v_p v_s = v$  where  $v$  is an existing word in the task dictionary ( $\Sigma$ ), and  $s''$  is the rest of the suffix. So, the search must be performed over all possible suffixes  $s$  of  $p$ , and the language model probability  $P(v_s, s'' \mid p'', v_p)$  must ensure that the concatenation of the last part of the prefix  $v_p$ , and the first part of the suffix,  $v_s$ , form an existing word ( $v$ ) in the task dictionary. This probability can be decomposed into two terms:

$$P(v_s, s'' \mid p'', v_p) = P(s'' \mid p'', v_p, v_s) \cdot P(v_s \mid p'', v_p) \quad (4)$$

The first term accounts for the probability of all the whole-words in the suffix, and can be modelled directly by (2). The second term should ensure that the first part of the suffix (usually a word-ending-part)  $v_s$ , will be a possible suffix of the incomplete word  $v_p$ , and can be stated as:

$$P(v_s \mid p'', v_p) = \begin{cases} \frac{P(v_p, v_s \mid p'')}{\sum_{v'_s} P(v_p, v'_s \mid p'')} & \text{if } v_p v_s \in \Sigma \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

### 3 Multimodal CATTI (MM-CATTI) at the Character Level

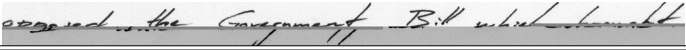
One way to increase the ergonomy and the usability in CATTI is to allow the user to provide his or her validating and/or corrective feedback by means of more comfortable peripheral devices, such as e-pen or touchscreen.

Clearly, decoding this kind of non-deterministic feedback consists in *on-line* HTR. As previously mentioned, the information available in the interaction process, allows us to boost the accuracy of this on-line HTR subsystem with respect to a conventional on-line HTR decoder (which do not make use of the interaction-derived information).

Let  $x$  be the representation of the input image and  $p'$  a user-validated prefix of the transcription. Let  $t$  be the on-line touchscreen pen strokes provided by the user. These data are related to the suffix suggested by the system in the previous interaction step,  $s'$ , and are typically aimed at accepting or correcting parts of this suffix. Using this information, the system has to find a new suffix,  $\hat{s}$ , as a continuation of the previous prefix  $p'$ , considering all possible decodings,  $d$ , of the on-line data  $t$  and some information from the previous suffix  $s'$ . That is:

$$\begin{aligned} \hat{s} &= \arg \max_s P(s \mid x, s', p', t) = \arg \max_s \sum_d P(s, d \mid x, p', s', t) \\ &\approx \arg \max_s \max_d P(t \mid d) \cdot P(d \mid p', s') \cdot P(x \mid s, p', d) \cdot P(s \mid p', d) \end{aligned} \quad (6)$$

An approximate two-step solution to this difficult optimization problem is followed (see Figure 1). In the first step, an “*optimal*” decoding,  $\hat{d}$ , of the on-line pen-strokes  $t$  is computed using only the first two terms of equation (6). After observing this decoding,  $\hat{d}$ , the user may type additional keystrokes,  $\kappa$ , to correct possible errors in  $\hat{d}$ . In the second step, the first two terms of (6) are ignored and  $d$  is replaced with  $\hat{d}$  in the last two terms. This way, a new consolidated prefix  $p = p' \hat{d}$  is obtained, which leads to a

		x						
INTER-0		$\hat{s} \equiv \hat{w}$	<b>opposite</b>	<b>this</b>	<b>Comment</b>	<b>Bill</b>	<b>in that</b>	<b>thought</b>
INTER-1	Step-1	$\hat{p}', \hat{t}$ $\hat{d}$ $\kappa$	<b>oppos<sup>e</sup></b> l e					
	Step-2	$\hat{s} \equiv \hat{s}'$	<b>oppose</b> d	<b>the</b>	<b>Government</b>	<b>Bill</b>	<b>in that</b>	<b>thought</b>
INTER-2	Step-1	$\hat{p}', \hat{t}$ $\hat{d}$ $\kappa$	<b>opposed</b>	<b>the</b>	<b>Government</b>	<b>Bill</b>	$\omega$ w	
	Step-2	$\hat{s} \equiv \hat{s}'$	<b>opposed</b>	<b>the</b>	<b>Government</b>	<b>Bill</b>	w <b>hich</b>	<b>brought</b>
FINAL		$\hat{p} \equiv \hat{T}$	<b>opposed</b>	<b>the</b>	<b>Government</b>	<b>Bill</b>	<u>w</u> <b>hich</b>	<b>brought</b> #

**Fig. 1.** Example of multimodal CATTI at character level interaction. The process starts when the HTR system proposes a full transcription of the handwritten text image  $x$ . Then, each interaction consists in two steps. In the first step the user handwrites some touchscreen to amend the suffix proposed by the system in the previous step. This defines a correct prefix  $\hat{p}'$ , which can be used by the on-line HTR subsystem to obtain a decoding of  $\hat{t}$ . After observing this decoding,  $\hat{d}$ , the user may type additional keystrokes,  $\kappa$ , to correct possible errors in  $\hat{d}$ . On the second step, a new prefix is built from the previous correct prefix  $\hat{p}'$ , the decoded on-line handwritten text,  $\hat{d}$ , and the typed text  $\kappa$ . Using this information, the system proposes a new potential suffix. The process ends when the user enters the special character “#”. System suggestions are printed in boldface and typed text in typewriter font. In the final transcription,  $\hat{T}$ , underlined italic characters are those which were typed by the user.

formulation identical to (1). These two steps are repeated until  $p$  is accepted by the user as a full correct transcription of  $x$ .

Assuming whole-word e-pen feedback, this approach was studied and tested in [7], with good results. Here we consider single-character e-pen strokes, which we think may lead to improved productive and usability. Therefore, we henceforth assume that  $\hat{d}$  consists of a single character. As in section 2, the prefix  $p'$  is divided into two parts:  $p''$  (fragment of  $p'$  formed by complete words) and  $v'_p$  (the last incomplete word of  $p'$ ). Therefore the first step of the optimization (6) can be written as:

$$\hat{d} = \arg \max_d P(t | d) \cdot P(d | p'', v'_p, s') \quad (7)$$

where,  $P(t | d)$  is provided by a morphological (HMM) model of the character  $d$  and  $P(d | p'', v'_p, s')$  can be approached by a language model dynamically constrained by information derived from the interaction process. Equation (7) may lead to several scenarios depending on the assumptions and constraints adopted for  $P(d | p'', v'_p, s')$ . We examine some of them bellow.

The first and simplest scenario corresponds to a naive approach where any kind of interaction-derived information is considered; that is,  $P(d | p'', v'_p, s') \equiv P(d)$ .

In a slightly more restricted scenario, we take into account just the information from the previous off-line HTR prediction  $s'$ . The user interacts providing  $t$  in order to correct the wrong character of  $s'$ ,  $e$ , that follows the validated prefix  $p'$ . Clearly, the er-

aneous character  $e$  should be prevented to be a decoding on-line HTR result. This *error-conditioned model* can be written as  $P(d | p'', v'_p, s') \equiv P(d | e)$ .

Another, more restrictive scenario, using the information derived from the validated prefix  $p'$ , arises when we regard the portion of word already validated ( $v'_p$ ), i.e.  $P(d | p'', v'_p, s') \equiv P(d | v'_p, e)$ . In this case the decoding should be *easier* as we know beforehand what should be a suitable continuation of the part of word accepted so far.

Finally, the most restrictive scenario corresponding to the additional consideration of the information provided by  $p''$ , is left for future studies.

### 3.1 Dynamic Language Modelling for Character-level MM-CATTI

Language model restrictions are implemented on the base of  $n$ -grams, depending on each multimodal scenario considered. As mentioned above, the simplest scenario is that which does not take into account any information derived from the interaction. In this case,  $P(d)$  can be modelled directly using uni-grams. This is the *baseline* case.

The second case,  $P(d | e)$ , only considers the first wrong character. The language model probability is given by

$$P(d | e) = \begin{cases} 0 & \text{if } d = e \\ \frac{P(d)}{1-P(e)} & \text{if } d \neq e \end{cases} \quad (8)$$

The next scenario, given by  $P(d | v'_p, e)$ , the on-line HTR subsystem counts not only on the first wrong character but also on the last incomplete word of the validated prefix  $v'_p$ . This scenario can be approached in two different ways: using a character language model or a word language model. In the first one, the on-line HTR subsystem uses a modified character  $n$ -gram model:

$$P(d | v'_p, e) = \begin{cases} 0 & \text{if } d = e \\ \frac{P(d|v'_{p_{k-n+2}})}{1-P(e|v'_{p_{k-n+2}})} & \text{if } d \neq e \end{cases} \quad (9)$$

In the second approach (10), we use a word language model to generate a more refined character language model. This can be written as:

$$P(d | v'_p, e) = \begin{cases} 0 & \text{if } d = e \\ \frac{P(d|v'_p)}{1-P(e|v'_p)} & \text{if } d \neq e \end{cases}$$

where:

$$P(d | v'_p) = \frac{P(v'_p, d)}{\sum_{d'} P(d', v'_p)} = \frac{\sum_{v_s} P(v'_p, d, v_s)}{\sum_{v_s} \sum_{d'} P(v'_p, d', v_s)} \quad (10)$$

being  $v'_p d v_s$  an existing word of  $\Sigma$ .

## 4 Off- and On-line HTR System Overview

Both the off-line and on-line HTR systems employ a similar conceptual architecture composed of three modules: *preprocessing*, *feature extraction* and *recognition*. The first

two entail different well-known standard techniques depending on the data type, but the last one is identical for both systems. The Off-line HTR preprocessing involves skew and slant corrections and size normalization operation [8]. On the other hand, on-line handwriting preprocessing encompasses repeated points elimination and noise reduction. Regarding feature extraction, the off-line case converts the preprocessed text into a sequence of 60-dimensional feature vectors, whereas the on-line preprocessed coordinates are transformed into a sequence of 7-dimensional feature vectors [6].

As explained above, the recognition process is similar in both cases. Characters are modelled by continuous density left-to-right HMMs with a Gaussian mixture per state. Each lexical word is modelled by a Stochastic Finite-State automaton, and text sentences are modelled using bi-grams with Kneser-Ney back-off smoothing. All these finite-state models can be easily integrated into a single global model in which decoding process is efficiently performed by the Viterbi algorithm.

## 5 Experimental Framework

For test the effectiveness of MM-CATTI at character level different experiments were carried out. The corpora and the performance measures used are explained below.

### 5.1 Assessment Measures

Some types of measures have been adopted to assess the performance of character-level transcription. On the one hand, to make the post-editing process more accurately comparable to CATTI at character level, we introduce a *post-editing autocompleting* approach. Here, when the user enters a character to correct some incorrect word, the system automatically completes the word with the most probable word on the task vocabulary. Hence we define the *Post-editing Key Stroke Ratio* (PKSR), as the number of keystrokes that the user must enter to achieve the reference transcription, divided by the total number of reference characters. On the other hand, the effort needed by a human transcriber to produce correct transcriptions using CATTI at character level is estimated by the *Key Stroke Ratio* (KSR), which can be defined as the number of (character level) user interactions that are necessary to achieve the reference transcription of the text image considered, divided by the total number of reference characters. These definitions make PKSR and KSR comparable and the relative difference between them gives us a good estimate of the reduction in human effort that can be achieved by using CATTI at character level with respect to using a conventional HTR system followed by human autocompleting postediting. This *estimated effort reduction* will be denoted as “EFR”.

Finally, since only single-character corrections are considered, the conventional classification error rate (ER) will be used to assess the accuracy of the on-line HTR feedback subsystem under the different constraints entailed by the MM-CATTI at character level interaction process.

### 5.2 Corpus

The character level CATTI was evaluated on the IAMDB corpus. For the MM-CATTI, the on-line UNIPEN corpus was employed to simulate the user touchscreen interactions.

The IAMDB [3] is a publicly accessible corpus composed of 1,539 scanned text pages, handwritten by 657 different writers. No restriction was imposed related to the writing style or with respect to the pen used. The database is provided at different segmentation levels: characters, words, lines, sentences and page images. Here we use sentence-segmented images. Each sentence is accompanied by its ground truth transcription as the corresponding sequence of words. To better focus on the essential issues of the considered problems, no punctuation marks, diacritics, or different word capitalizations are included in the transcriptions. From 2,324 sentences that forms the corpus, 200 were used as test, leaving the rest as training partition.

The UNIPEN corpus [1] comes organized in several categories: lower and uppercase letters, digits, symbols, isolated words and full sentences. For our experiment, three UNIPEN categories were used: *Ia* (digits), *Ic* (lowercase letters) and *Id* (symbols). Three arbitrary writers were chosen as test partition and 17 as training data [7].

### 5.3 Results

Different experiments have been carried out to assess the feasibility and potential of CATTI at character level. Two types of results are reported for CATTI at character level: the PKSR (first column of table 2) and the KSR (second column of table 2). The 12.5% of KSR corresponds to a total of 1,627 characters that the user has to correct. In the MM-CATTI at character level these characters would have to be handwritten by the user on the touchscreen. It is simulated here using character samples belonging to a same writer from the UNIPEN corpus.

As we mentioned earlier, the introduction of multimodal interactivity leads, on the one hand, to an ergonomic and easier way of working, but on the other hand, to a situation where the system has to deal with non-deterministic feedback signals. Therefore, two of the most important concerns here is the accuracy of the on-line HTR subsystem and the determination of how much this accuracy can be boosted by taking into account informations derived from the interaction process. Table 1 reports the writer average

**Table 1.** On-line HTR subsystem error rates for the four language models: plain character uni-gram (CU, *baseline*), error conditioned character uni-gram (CU<sub>e</sub>), prefix-and-error conditioned character bi-gram (CB<sub>e</sub>) and prefix-and-error conditioned word uni-gram (WU<sub>e</sub>). The relative accuracy improvements for CU<sub>e</sub>, CB<sub>e</sub> and WU<sub>e</sub> are shown in the last three columns. The same GSF value (15) is used for all the cases. All values are in percentages.

Error Rate				Relative Improv.		
CU	CU <sub>e</sub>	CB <sub>e</sub>	WU <sub>e</sub>	CU <sub>e</sub>	CB <sub>e</sub>	WU <sub>e</sub>
7.0	6.9	6.7	5.0	1.4	4.3	28.6

feedback on-line recognition error rate of characters considering the different scenarios studied in section 3. As observed, feedback decoding accuracy increases significantly as more interaction derived constraints are taken into account. In addition, Table 1 also shows the relative accuracy improvements obtained respect to the baseline case.

As a final overview, Table 2 summarizes all the CATTI and MM-CATTI results obtained in this work. The third and fourth columns show the MM-CATTI KSR for the



**Table 2.** From left to right: PKSR obtained with the post-editing autocompleting approach, KSR achieved with CATTI at character level and KSR obtained with the *baseline* and best scenarios for MM-CATTI approach. EFR for KSR of CATTI with respect to PKSR and for KSR for the two scenarios of MM-CATTI with respect to PKSR. All results are in percentages.

PKSR	CATTI	MM-CATTI		EFR		
	KSR	CU-KSR	WU <sub>e</sub> -KSR	CATTI	MM-CATTI (CU)	MM-CATTI (WU <sub>e</sub> )
15.8	12.5	13.4	13.1	20.9	15.2	17.1

baseline as well as the best scenarios. This values are calculated under the simplifying assumption that the cost of keyboard-correcting a feedback on-line decoding error is similar to that of another on-line touchscreen interaction step. That is, each correction is counted twice: one for the failed touchscreen attempt and another for the keyboard correction itself. According to these results, the expected user effort for the best MM-CATTI approach is only barely higher than that of CATTI.

## 6 Conclusions

In this paper, we have studied the character level interaction in the CATTI system presented in previous works using pen strokes handwritten on a touchscreen as a complementary means to introduce the required CATTI correction feedback. From the results, we observe that the use of this more ergonomic feedback modality comes at the cost of a reasonably small number of additional interaction steps needed to correct the few feedback decoding errors. The number of these extra steps is kept very small thanks to the ability to use interaction-derived constraints to considerably improve the on-line HTR feedback decoding accuracy.

## References

1. Guyon, I., Schomaker, L., Plamondon, R., Liberman, M., Janet, S.: UNIPEN Project of On-Line Data Exchange and Recognizer Benchmarks. In: Proc. of the 14th International Conference on Pattern Recognition. pp. 29–33. Jerusalem (Israel) (1994)
2. Jelinek, F.: Statistical Methods for Speech Recognition. MIT Press (1998)
3. Marti, U.V., Bunke, H.: A full English sentence database for off-line handwriting recognition. In: Proc. of the ICDAR'99. pp. 705–708. Bangalore (India) (1999)
4. Rabiner, L.: A Tutorial of Hidden Markov Models and Selected Application in Speech Recognition. Proc. IEEE 77, 257–286 (1989)
5. Romero, V., Toselli, A.H., Vidal, E.: Character-level interaction in computer-assisted transcription of text images. In: Proc. ICFHR 2010, pp. 539–544. Kolkata, India (November 2010)
6. Toselli, A.H., Pastor, M., Vidal, E.: On-Line Handwriting Recognition System for Tamil Handwritten Characters. In: Proc. IbPRIA'07, LNCS, vol. 4477, pp. 370–377. Springer-Verlag
7. Toselli, A.H., Romero, V., Pastor, M., Vidal, E.: Multimodal interactive transcription of text images. Pattern Recognition 43(5), 1814–1825 (2010)
8. Toselli, A.H., Romero, V., i Gadea, M.P., Vidal, E.: Preprocessing and feature extraction techniques for multimodal interactive transcription of text images. Tech. rep., Instituto Tecnológico de Informática, <http://prhlt.iti.es> (2008)