Interactive Multimodal Transcription of Text Images Using a Web-based Demo System

Verónica Romero  Luis A. Leiva  Alejandro H. Toselli  Enrique Vidal
ITI - Instituto Tecnológico de Informática
Universidad Politécnica de Valencia  Spain
[vromero,luileito,ahector,evidal]@iti.upv.es

ABSTRACT
This document introduces a web based demo of an interactive framework for transcription of handwritten text, where the user feedback is provided by means of pen strokes on a touchscreen. Here, the automatic handwriting text recognition system and the user both cooperate to generate the final transcription.

Author Keywords
Handwritten recognition, interactive framework, Web, HCI

INTRODUCTION
Current off-line automatic Handwritten Text Recognition systems (HTR) are far from being perfect and, in general, human intervention is required to check and correct the results thrown by such systems. This approach is rather inefficient and uncomfortable for the user. As an alternative to post-editing, a multimodal interactive approach is proposed here, where user feedback is provided by means of touch-screen pen strokes and/or more traditional keyboard and mouse operation [5]. User’s feedback directly allows to improve system accuracy [4], while multimodality increases system ergonomy and user acceptability. Multimodal interaction is approached in such a way that both the main and the feedback data streams help each other to optimize overall performance and usability.

This new multimodal interactive approach for transcription of text images, called MM-CATTI from here onwards, is shown to work quite well by an implemented Web-based Demo. Figure 1 shows an user interacting with the MM-CATTI system by means of a touch-screen. The online form of such MM-CATTI system allows to carry out collaborative tasks with thousands of users across the globe, thus reducing notably the overall image recognition process. Since the users operate within a web browser window, the system also provides cross-platform compatibility and requires no disk space on the client machine.

USER INTERACTION PROTOCOL
In MM-CATTI, the user is involved with the transcription process, where following a preset protocol, he/she validates and/or corrects the HTR output during the process. The protocol that rules this interaction process, is formulated in the following steps:

• The HTR system proposes a full transcription of the input handwritten text line image.
• The user validates the longest prefix of the transcription which is error-free and enters some on-line touchscreen pen strokes and/or some amendment keystrokes to correct the first error in the suffix.
• An on-line HTR feedback subsystem (or HFR) is used to decode this input into a word (or word sequence).
• In this way, a new extended consolidated prefix is produced based on the previous validated prefix, the on-line decoding word and the keystroke amendments. Using this new prefix, the HTR suggests a suitable continuation of it.
• This previous steps are iterated until a final, perfect transcription is produced.

In figure 2 we can see different interaction modes by using the web-based demo making use of pen-strokes. The user can write down the correct word directly, make a diagonal line to delete an erroneous word, make a vertical line followed by a word for inserting that word, or make a single click to ask for another suitable continuation [3].

Technology
Off-line and On-line HTR Basic Systems
Both the main off-line HTR system and the on-line HTR subsystem (HFR) employ a similar conceptual architecture, which is composed of three modules: preprocessing, feature

Copyright is held by the author/owner(s).
IUI’09, February 8-11, 2009, Sanibel Island, Florida, USA.
ACM 978-1-60558-331-0/09/02.
Figure 2. Four interaction modes to validate an error-free prefix. From left to right: substitution, single click validation, deletion, and insertion.

**Search Process Constraints based on User Feedback**

The transformation of user feedback into constraints imposed on the search process is carried out by adapting systematically the language model to cope with each consolidated prefix. This adaptation lies on building a special language model, that can be seen as the “concatenation” of a linear model which strictly accounts for the successive words in the prefix (in case it exists) and a normal n-gram that accounts for the rest of words in the suffix. In this way, the decoder should be forced to match the previously validated prefix, followed by some suffix according to the constraints of the corresponding n-gram.

**EVALUATION RESULTS**

To test the effectiveness of the MM-CATTI approach, several experiments were carried out on three corpora corresponding to different handwritten text transcription tasks: ODEC, IAMDB, and CS [5, 4, 3]. From the reported results on these corpora, and assuming for simplicity that the cost of correcting an on-line decoding error is just similar to the one provided by another on-line touchscreen interaction, the estimated human effort to produce error-free transcription using MM-CATTI is reduced by a 15% on average, regarding to the classical HTR system. So, from every 100 words misrecognized by a conventional HTR system, a human post-editor will have to correct all the 100 erroneous words, while a MM-CATTI user would correct only 85 – the other 15 are automatically corrected by the system.

**DEMO DESCRIPTION**

The proposed system coordinates client-side scripting with server-side technologies (see Figure 3). The web interface loads initially an index of all available pages in the corpus. The user then navigates to a page and begins to transcript the handwritten text images line by line. He/She can make corrections with any kind of mouse pointer (i.e: touchscreen or stylus) and also use the keyboard. Mouse strokes data are sent to the HFR and then processed by the MM-CATTI engine, while keystrokes data interact directly with the aforementioned MM-CATTI engine. Client-Server communication is made asynchronously via Ajax, providing thus a richer interactive experience, and Server-Engine communication is made through PHP. All corrections are stored in plain text logs on the server, so the user can retake them in any moment.

**REFERENCES**