A COMPARISON OF TWO ON-LINE HANDWRITING RECOGNITION METHODS FOR UNCONSTRAINED TEXT ENTRY BY CHILDREN

Janet C Read	Stuart MacFarlane	Chris Casey
UCLAN	UCLAN	UCLAN
Preston, UK	Preston, UK	Preston, UK
jcread@uclan.ac.uk	sjmacfarlane@uclan.ac.uk	ccasey@uclan.ac.uk
www.chici.org	www.chici.org	www.chici.org

ABSTRACT

In this paper, we describe a small study that compared real time recognition and batch recognition for on-line handwriting recognition based text input with child users aged 7 and 8. The study involved the children in writing text using a pen and graphics tablet and in error repair using the pen, keyboard and mouse. The study compared speed of use and ease of use and measures of efficiency and error handling were taken. It was found that in the real time mode there was more error repair to be done and results suggest that the children took longer to complete their tasks with this mode. Most of the children that used the two methods preferred real time recognition.

Keywords

Children, Handwriting recognition, Text input, Usability, Errors, Writing.

1. INTRODUCTION

Handwriting recognition is the process of turning handwritten work into computer text. There are two different technologies, one for off- line recognition and one for on-line recognition. This paper is only concerned with on-line recognition. In this sort of system, the user writes onto a graphics tablet with a stylus or pen. The writing is displayed on the computer screen and is recorded as an 'ink' file which contains data about position and time. At some point the ink file is 'recognised' and this results in a stream of ASCII characters. These characters are generally displayed on the screen and are thereinafter referred to as 'recognised text'. It is common for the recognised text to include errors as the recognition algorithms generally have to make sense of incomplete or 'noisy' script.

Proceedings Volume 2 of the
17th British HCI Conference
London, September 2003

1.1 Two Handwriting Recognition Methods – Real time and Batch

On-line handwriting recognition may take place in real time or as a batch process. In real time recognition, the words get recognized almost as soon as they are written; in batch recognition, the ink files are stored ready to be recognized later. Research by van Gelderen *et.al*. found that for adult users there was no difference in the time taken for repair with these different methods [1]. It has been observed that when children use handwriting recognition in real time mode, they notice when their writing is mis-recognized and they have been seen attempting to improve the character formation for subsequent versions of the problematic letter [2]. It has been hypothesised that this real time recognition may also help reinforce the child's mental model of the recognition process. However, when handwriting is recognized as a batch, there is an extra opportunity for repair before it is recognised and the child is not distracted in their writing in the same way that they are with real time recognition.

1.2 Usability Metrics for Text Input

Work on the evaluation of text input methods focuses on the three usability measures of efficiency, effectiveness and user satisfaction [3]. Efficiency is typically measured by counting characters entered per second and effectiveness is measured by making comparisons between two strings, the first being the presented text (PT) and the second the transcribed text (TT) [4]. From these two strings a CER (Character Error Rate) is derived which is the ratio of the number of errors to the number of characters in the presented text. This metric is generally converted into a percentage correctness figure and is referred to as the recognition rate. User satisfaction is measured by the use of questionnaires and by observation.

1.3 Improving the Usability of Handwriting Recognition for Text Input

The nature of any recognition-based system is that both the system and the user can cause errors. Much work is being done to improve the recognition algorithms [5]. Constraining the user can also result in improved recognition rates. Methods include making the user write in boxes, use a reduced alphabet, or use just discrete (non-cursive) writing [6], [7]. Limiting the vocabulary has also been shown to increase recognition rates [8]. An 'intelligent' system would be able to identify if a user was repeatedly constructing a letter incorrectly and would be able to provide assistance [9].

The interface can be made more efficient and more effective by providing good error recovery strategies. These include improved error discovery and the provision of effective strategies for error repair [10]. An observational study by the authors on children finding and repairing errors in a handwriting recognition based system concluded that when children had access to a pen and a keyboard they used different repair strategies which had different costs in relation to time and effort. Work by Ward *et.al*. on error repair in speech recognition interfaces concluded that the complexity of error repair strategies directly impacts on user satisfaction [11].

2. THE STUDY

This study was designed to determine the usability of these two different recognition methods for children doing unconstrained text entry. The first method was immediate recognition (real time) whereby each word that they had constructed would vanish from the screen almost as soon as it was completed and in its place there would be ASCII text. The second was batch recognition, in which the child would initiate the recognition when he had written all his words on the screen. The intention in this study was to investigate which of these two methods was most efficient, which was easiest to use (i.e. had least error repair), and whether or not the children preferred one to the other.

2.1 Method

2.1.1 Subjects

Twelve children participated in this study (7 females, 5 males). They were aged 7 and 8 and were all from one British Primary School. The children were selected to represent the full ability range within the target age group. Two of the children were left handed, the rest were right handed. One further child attempted the study but wanted to quit before he had completed both tasks. The results from his work were excluded from the recorded data

2.1.2 Hardware / Software

The equipment that was used was a HiGrade Notino laptop computer together with a Wacom Graphics tablet. The recognition software was Calligrapher Pen Office 2, which had been optimised for letter shape and was referencing a standard dictionary. The keyboard of the laptop was enabled and a 2-button mouse as well as the laptop touch pad was available for the children to use.

2.1.3 Procedure

The selected children had all used the pen and tablet on two previous occasions. They had copied writing without seeing it being recognised and without doing any repair. The children were all instructed to use the pen to initially write the text but they were told that they could use the pen, the mouse and the keyboard for error repair. Prior to this study, it was discovered that they were generally unsure about error repair and so the researcher showed each child how text could be corrected using the three devices. They were also shown the actions of the delete and backspace keys on the keyboard.

In order to provide comparable text for each child, it was decided that a piece of text would be used that they would know, this would eliminate the need for copying or for having the text read aloud to them [12], [3]. A well-known phrase from children's TV **'Bob the builder can he fix it'** was chosen as the entry text. This had no punctuation and the spelling of 'builder' was revised with each child before the start of the activity. Children found this easy to remember and it had a good selection of letters and one tricky word. They were instructed to write 'as clearly as possible'.

2.1.4 Design

The design was within-subjects single factor with test condition: recognition takes place during writing (real time) vs. recognition takes place after writing (batch). Half the children did each condition first. Both conditions were tested with the same input text and the same interface. The recognised text was presented to the children in 24-point Comic Sans Font on the computer screen. There were 23 characters to write which, once spaces were added, resulted in 29 characters in the correct representation. The time taken for each condition was recorded.

The researcher logged the different actions by the children. These were

KS Keystrokes (writing a character with the pen, typing or deleting a character at the keyboard)
EF Errors found and corrected

Three counts were taken;

CG Characters generated by the recogniser
UE Uncorrected errors in the final text
CF Characters at the end

After the two conditions the children were asked which set up they preferred. Three calculations were made on the data, these being a KSPC (Key strokes per Character) measure, a CER (character error rate) measure and a conscientiousness measure (EC) as described by Soukoreff and MacKenzie [13].

The equations for these measures were;

$KSPC = KS / CF.$
$CER = UE / CF$
$EC = EF / (EF + UE)$

2.2 Results

The table that follows shows the mean and median for each of the measures that were taken. These are displayed to two significant figures. Time is given in minutes.

	Real Time		Batch	
	Mean	Median	Mean	Median
KS	61	54	45	37
CG	45	39	32	29
CF	31	31	29	29
UE	2.3	2.0	2.3	1.5
KSPC	2.0	1.6	1.6	1.4
CER	7.7	6.9	8.1	5.3
EF	47	42	25	17
EC	7.3	5.3	3.9	2.6
Time taken	5.4	4.5	3.7	2.5

Table 1 – Results from the two conditions

A Wilcoxon test was applied to the results for time and errors fixed. This was preferred to a paired t test due to large numbers that appeared for two children who got into spirals of error repair thus making the data non normal. The results of the Wilcoxon test on the number of errors fixed were significant ($p \leq 0.05$, when $N = 12$, $T = 15$) which allows us to conclude that where the recognition happened in batch mode, there was less error repair. There was not a significant difference between the results for time. 75% of the children preferred the condition where the recognition happened in real time.

2.3 Observations from the Experiment

Children were expected to write 23 characters, however, on six occasions they wrote 24 characters as they remembered the text as 'Bob the builder can you fix it', four other trials resulted in 22 characters being written as they spelt 'builder' wrong, missing the u.

2.3.1 Spaces

There was a problem with the spaces between words in both of the conditions. When the recognition took place in batch mode, children had to insert a total of 20 spaces in the ASCII text. When the recognition took place in real time only 6 spaces had to be inserted. However, in this latter case, 19 spaces appeared where they were not wanted! The 'cost' of fixing the space errors was similar for each condition.

2.3.2 Repairing the Text

The repair process required the children to find errors and decide how to fix them. This involved them in inserting spaces and characters, and in deleting spaces and characters. As can be seen from the error conscientiousness score, over 90% of the errors made were corrected. Half the children in the study did not pick up the pen again after their initial writing; they used only the mouse and keyboard to effect repairs. Two children used the pen extensively for error repair in the real time recognition condition, but only once or twice in the batch recognition condition. Errors that remained included extra spaces, missing words, extra letters, incorrect letters, reversed letters (b for d), and incorrect capitalization.

2.3.3 Getting to the Point of Repair

Once there was a stream of ASCII text on the screen, children had to move the text cursor to arrive in the right place for editing. Different children did different things, and some children used more than one method. Only two children never used the mouse, preferring in one case the arrow keys on the keyboard, and in the other case the pen. All the other children used the mouse to move and position the text cursor. Only two children attempted to use the pen as a pointer, one exclusively and the other used it with the mouse. The children all functioned single-handedly, that is they typed with the same hand that they wrote with and used the mouse with.

3. Conclusion - Implications for the Design of Writing Interfaces for Children

While it is not possible to conclude that one recognition input method is always better than the other, the results in this paper highlight some interesting areas for future work. Most of the children in this small study preferred the method that appeared more difficult and more inefficient. When the results are examined, it can be seen that 75% of the children preferred the method in which they made and subsequently had to fix the most errors and 75% of the children preferred the method in which they had to do most work (their higher KSPC score). This is a bit of a surprise. It may be that children found real time recognition to be more fun; the immediate disappearance of the writing and its subsequent reappearance was entertaining. Further work needs to be done to establish why children liked this method of recognition. It may also be useful to establish whether or not these same findings are replicated with other recognition-based interfaces.

This study used only a short piece of text and did not allow the children the opportunity to write what they wanted. A lengthier writing activity would have allowed some evaluation of the effect of learning by the children, as they observed the recognition process. In addition, a more extended writing activity would allow some assessment of the distracting effect on the composition process that can be caused by the immediate recognition of text.

The repair strategies used by the children were not always very efficient. The two children that used the pen extensively to repair their writing both took much longer than their peers. Children need to be given effective strategies for error repair and it seems likely that a multi-modal repair system is needed. Future work will examine how easy it is for children to learn and use pen gestures for error repair and will determine whether or not this is a worthwhile strategy.

4. REFERENCES

[1] vanGelderen, T., A. Jameson, and A. Duwaer. *Text Correction in Pen-Based Computers: An Empirical Comparison of Methods*. in *CHI 93*. 1993. Amsterdam: ACM Press. p. 87 - 88.

[2] Read, J.C., S.J. MacFarlane, and C. Casey. *Oops! Silly me! Errors in a Handwriting Recognition-based Text entry Interface for Children*. in *NordiChi 2002*. 2002. Aarhus, Denmark: ACM Press. p. 35 - 40.

[3] Mackenzie, I.S. and R.W. Soukoreff, *Text Entry for Mobile Computing: Models and Methods, Theory and Practice.* Human-Computer Interaction, 2002. **17**: p. 147 - 198.

[4] Mackenzie, I.S. and R.W. Soukoreff. *A Character-Level Error Analysis for Evaluating Text Entry Methods*. in *NordiChi2002*. 2002. Aarhus, Denmark: ACM Press. p. 241 - 244.

[5] Plamondon, R. and S.N. Srihari, *On-line and Off-Line Handwriting Recognition: A Comprehensive Survey.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000. **22**(1): p. 63 - 84.

[6] Frankish, C., *Pen Input*, in *Interface Technology: The leading Edge*, J.M. Noyes and M.J. Cook, Editors. 1999, Research Studies Press: Baldcock, Hants. p. 59 - 72.

[7] Goldberg, D. and A. Goodisman. *STYLUS User Interfaces for Manipulating Text*. in *UIST'91*. 1991. South Carolina: ACM Press. p. 127 - 135.

[8] Tappert, C.C., C.Y. Suen, and T. Wakahara, *The State of the Art in On-Line Handwriting Recognition.* IEEE Transactions on Pattern Analysis and Machine Intelligence, 1990. **12**(8): p. 787 - 808.

[9] Noyes, J., *Talking and writing-how natural in human-machine interaction?* International Journal of Human-Computer Studies, 2001. **55**: p. 503-519.

[10] Mankoff, J., G. Abowd, and S. Hudson, *OOPS: a toolkit supporting mediation techniques for resolving ambiguity in recognition-based interfaces.* Computers and Graphics, 2000. **24**: p. 819 - 834.

[11] Zajicek, M. and J. Hewitt. *An Investigation into the use of Error Recovery Dialogues in a User Interface Management System for Speech Recognition*. in *IFIP Interact '90*. 1990. p. 755 - 760.

[12] Ward, D.J., A.F. Blackwell, and D.J.C. MacKay. *Dasher: A data entry interface using continuous gestures and language models*. in *UIST 2000*. 2000. New York: ACM Press. p. 129 - 137.

[13] Soukoreff, R.W. and I.S. MacKenzie. *Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric*. in *CHI2003*. 2003. Ft. Lauderdale: ACM Press. p. 113 - 120.