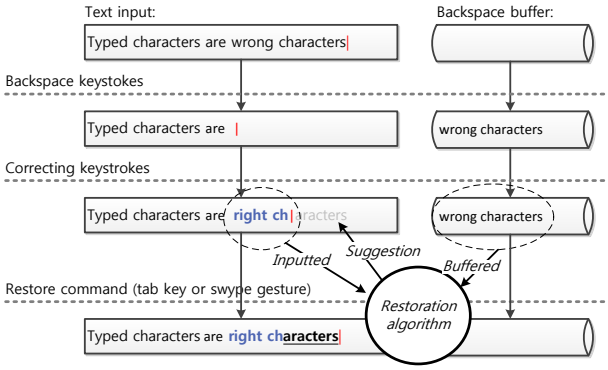
As an example, consider the following case:

Typed Text: `resd books with ebook`

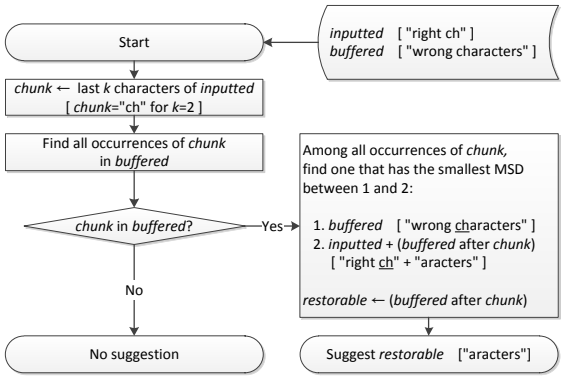
Deleted Text: ~~`resd books with ebook`~~

Retyped Text: `read bo`

In this case, *inputted* is "ead bo" and *buffered* is "esd books with ebook". *Chunk* is set to "bo" and two "bo"s are found (underlined) in *buffered*. For the first "bo", MSD between *buffered* and "ead books with ebook" is 1 and, for the second "bo", MSD between *buffered* and "ead book" is 13. Therefore, *restorable* is set to "oks with ebook".



(a)



(b)

Figure 1: *Restorable Backspace*: (a) an example sequence (A user changes "wrong characters" to "right characters".) and (b) *Restoration* algorithm (example values in brackets)

PILOT STUDY

We conducted a pilot study to observe that the proposed algorithm is working as intended and to confirm that the concept of *Restorable Backspace* is acceptable to users. Four participants (all male, ages from 26 to 29) performed a transcribing task with *Restorable Backspace* on 11" MacBook Air for 30 minutes. All participants were touch-typists. Sentences were randomly selected from the MacKenzie and Soukoreff phrase set [1]. The parameter *k* of *Restoration* algorithm was set to 2 throughout the pilot study because we wanted the algorithm to make suggestions as early as possible but, with $k = 1$, suggestions were too frequent and often incorrect.

Throughout the pilot study, 18 suggestions were accepted by participants and they expressed favorable opinions about the concept. One participant said "It feels great because my effort of typing was not wasted." From logged data and video recording, we could observe the followings:

1. *Restorable Backspace* worked well in most cases. Among the 18 accepted suggestions, 17 suggestions (for 10 insertion errors, 6 omission errors, and 1 substitution error) recovered right words successfully.
2. It did not work in some cases. For example, a participant typed "needs" instead of "knees". He deleted it and typed "knee". Then, "ds" appeared as a suggestion. This was not a correct suggestion, but he accepted it unconsciously, and corrected it again.
3. Participants often stayed stalled for a while after automatic restoration. One participant said "I lost my flow of typing." Especially, they delayed more when the final word of the restored text was incomplete.

CONCLUSION AND DISCUSSION

We proposed *Restorable Backspace* concept and implemented *Restoration* algorithm for it. All participants in a pilot study showed satisfaction about the new concept.

As mentioned in the beginning, *Restorable Backspace* will be more useful in a mobile touchscreen environment. First of all, as we discussed in the introduction, *backspace method* is more likely in this environment and therefore more correctly inputted characters will be deleted. Another reason is that users in a mobile environment focus more on the onscreen keyboard than on the text input field, and therefore errors are more likely to be detected belatedly.

Although the algorithm worked fine for most of the cases, there is a lot of room for refinement. For example, the current algorithm suggests all characters in the buffer at once. As people often treat one word at a time, suggesting one word at a time may be more effective. Also, problems observed in the second and the third observations emphasize the necessity of a word level analysis and algorithm.

ACKNOWLEDGEMENT

This work was supported by the IT R&D program of MKE/KEIT. [10039161, Development of core technologies for high-physicality control interfaces and personalized intelligent user interfaces based on Smart TV user experience]

REFERENCES

1. MacKenzie, I. S., and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. *CHI 2003 extended abstracts*.
2. Siek, K., Rogers, Y., and Connelly, K. Fat finger worries: How older and younger users physically interact with pdas. *INTERACT 2005*.
3. Soukoreff, R. W., and MacKenzie, I. S. Measuring errors in text entry tasks: an application of the levenshtein string distance statistic. *CHI 2001 extended abstracts*.