Improving the Robustness to Input Errors on Touch-based Self-Service Kiosks and Transportation Apps

Frode Eika Sandnes[1,2(✉)]

[1]Department of Computer Science, Oslo Metropolitan University, Oslo, Norway
[2]Faculty of Technology, Kristiania University College, Oslo, Norway
frodes@oslomet.no

Abstract. Text input is cumbersome on public self-service kiosks and apps. Applications often provide prefix-based text suggestions to speed up input, but the input must be correct. However, application specific lists are usually limited in size. For example, the Norwegian rail network comprise 324 stations. This can be exploited in the input process. Several approaches are explored herein for more flexible text input that are robust to errors.

Keywords: self-service kiosks, travel apps, text entry, error correction

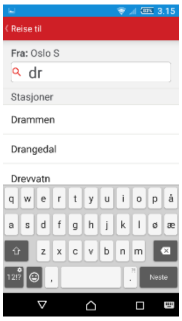
1 Introduction

Public self-service kiosks have become an important part of the transport infrastructure in large cities allowing unmanned service of stations. Apps have also emerged as an alternative [1] and the two often coexist side by side. One common step is the input of destination names using text entry.

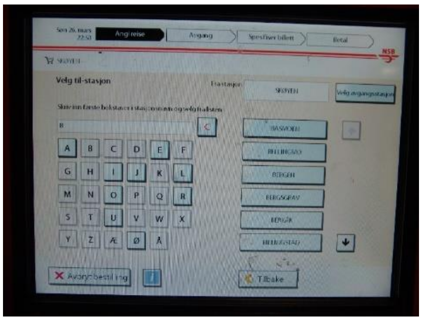
Moderate use of text input is a useful principle applicable to both touch-based self-service kiosks and smartphone apps. In some situations, one may avoid the problem altogether by other means such as QR-codes [2]. When selecting from a small list of items, such ten station names, the interface can rely on recognition and selection (see Fig. 1 (a)). However, with larger item lists that cannot be fitted on a simple screenful, such as a few hundred station names, it may be impractical to perform interaction through recognition. In these instances, the relevant item can be recalled through text input (see Fig. 1 (c) and (d)). However, text input on public self-service kiosks can be challenging for some users if they have limited experience using the system and feel pressure from being observed by onlookers. Also, often public kiosks have low-quality displays, dated technology or damaged displays making it hard to hit the exact targets. Visitors such as tourists may not have the local knowledge needed to select destinations with unintuitive names such as “Gardemoen” to get “Oslo Aiport”.

The input of text via smartphone apps is also challenging for many users due to the small screen real-estate with small virtual keys, making it challenging to hit the targets (see Fig. 1 (b)).

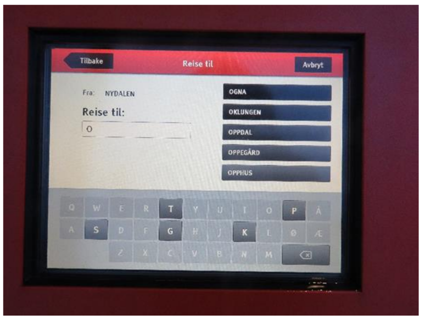
(a) Taiwan High Speed Rail).



(b) NSB travel app.



(c) NSB ticket kiosk (alphabetical, 2006).



(d) NSB ticket kiosk (QWERTY, 2016).

Fig. 1. Selecting travel destination.

Public kiosks and smartphone can display a similar amount of information on a single screen. Although the kiosk displays are larger than smartphone displays they are viewed from longer distances. Also, one must prevent information overload.

Text input can be challenging for individuals with disabilities. With reduced vision one may not be able to perceive the information. With motor disabilities such as tremor one may hit incorrect targets. Fitts' law explains the relationship between the time to hit a target, the distance to a target and the size of the target [3]. Users with cognitive disabilities, such as dyslexia, may have problems both decoding and encoding text [4, 5], and thus benefit from query-building aids [6].

Although the item lists are too large to be displayed on one screen, the lists can still be relatively small. When a list contains few textual elements (a few hundred) one may exploit the fact that the limitations allow fewer valid choices and hence provide stronger error correcting power than is possible with full dictionaries.

This study explored several strategies for exploiting small language models with the overall goal of achieving universal accessibility [7]. Four methods are investigated, namely how to compensate for (a) motor input error due to accidentally hitting neighboring targets, (b) the effort needed to input long words, (c) cognitive disabilities causing incorrect letter orderings and (d) a mixture of motor input error and cognitive errors. The methods are enhancements of the prefix method and users without reduced motor or cognitive function should not notice the added functionality. The issue of low vision is not addressed although transportation is a major challenge for individuals with reduced vision [8]. Kiosk interfaces targeted at users with low vision can for instance employ gestures with audio feedback to facilitate input [9, 10]. Moreover, gestures are usually relative movements and hence the problem of hitting absolute targets is avoided altogether. Alternatively, tactile feedback can be used [11].

2 Background

There has been much research into various aspects of self-service kiosks such as readiness for the technology [12], accessibility issues [13], ergonomic issues [14], service design [15], prototyping [16], evaluation techniques [17], etc. Serf-service kiosks spans a range of areas such as food and catering [18], ATMs [19], healthcare [20], navigation in hospitals [21], vending machines [22], interactive public displays [23] and for public transport and ticket purchase [24]. A self-service kiosk is usually placed in a public space [25] and is meant to be usable by all and there has thus been some focus on the universal accessibility of such kiosks [26].

In previous work we have focused on making kiosks accessible to blind users, especially given the task of selecting stations [10] and adapting dynamically to the needs of the user [27]. We have also explored the effect of keyboard layout for the same task, whereby presenting the users with a familiar layout is less likely to impose unnecessary stress [28]. There is a vast literature on text entry in general [29, 30, 31].

There is a long history of research into spell checking and correcting from the more classic methods using dictionaries [32] to newer methods using Hidden Markov Models [33] and online texts [34]. Attempts have also been made to study the mistakes made by dyslexic users [35] and users with reduced motor function [36]. Geometric models have been used to relax the need to hit targets exactly [37].

3 Method

The language model comprises a list of 324 train stations acquired from timetables. The station names were converted to lowercase to simplify processing.

3.1 Motor Errors: Fuzzy Keys

To compensate for motor input errors, it is assumed that the user is using a virtual QWERTY keyboard. Moreover, it is assumed that simple motor errors are caused by missing the target, that is, the user hits one of the neighboring keys. For simplicity, it is assumed that horizontal errors are more frequent than vertical errors, especially as virtual smartphone keyboards in portrait mode have tall and narrow keys (see Fig. 1 (c)).

For each keypress the two horizontally neighboring keys are considered as alternative intended keystrokes. For example, if the user enters the character *C*, the character *X* on the left and character *V* on the right are included into the set of potential input characters. Next, all combinations of the three possibilities for each keypress are combined and each combination is checked against the list of words. All the words with matching prefixes are added to the word suggestion list. Clearly, with one-character input there are 3 possibilities, with two input characters there are 9 possibilities, with 3, 4, 5, 6, 7 and 8 characters there are 27, 81, 243, 729, 2187 and 6561 possibilities, respectively. Imagine the user intends to write “Oslo”, but starts with the right hand in a wrong position and writes “Iski”. The four characters give 81 possibilities including “Uiau”, “Uiai”, “Uiao”, “Uisu” and also the correct word “Oslo”. The correct string is selected when correctly identified in the list of words.

3.2 Reducing Input Effort: Abbreviations

Prefix-based input works well with the English language where most words are not combined with other words. In other languages, such as Norwegian, compound words are built by combining words, and more words thereby share the same prefix. The user must enter long prefixes before a reliable suggestion can be made. To solve this problem abbreviation input was proposed [38]. With abbreviation-based input the user only inputs selected characters at various positions in the word. This was implemented by using the least common subsequence (LCS) algorithm.

Imagine we want to input “Lundamo”. There are four words that share the same 3 letter prefix, namely Lundamo, Lunde, Lunden and Lunner. Three of these also share the same 4 letter prefix “Lund”. With abbreviation-based input the user may enter the four consonants “lndm” to retrieve “LuNDaMo”.

3.3 Cognitive Errors: Bag-of-letters

Cognitive errors can cause characters to appear in the wrong sequence. To recover from such input errors the following strategy was performed. An entered prefix is matched against all the words in the list of words. A count of matching characters is maintained. If the number of matching characters equal the length of the prefix the associated word is included in the list of suggestions. Imagine the user intends to input “Oslo”, but accidentally inputs “Olso”. Since all the characters in Olso also are in Oslo the word is correctly identified.

3.4 Motor and Cognitive Errors: Fuzzy-bag

An attempt was made to combine fuzzy-key and bag-of-letters. This was achieved by generating the set of potential input characters sequences using fuzzy key. Next, each of the contenders were matched against the entries in wordlist using bag-of-letters. Imagine we wish to input “Oslo”, but we accidentally place the right hand incorrectly, and also accidentally swaps the two middle characters, and thus input “Iksi”.

3.5 Measurements

The four methods described herein were evaluated according to the number of available suggestions at each step of the input process. The average and the maximum number of suggestions were recorded by systematically inputting the list to the system one character at a time in offline mode while recording the suggestions at each step.

Table 1. Mean and maximum number of suggestions (M = mean, mx = maximum).

	prefix		abbreviation		fuzzy key		Bag-of-letters		fuzzy bag		Levenshtein	
N	M	mx	M	mx	M	mx	M	mx	M	mx	M	mx
1	23	50	98	202	81	322	98	202	0	0	0	0
2	5	14	24	86	24	322	36	123	0	2	5	14
3	2	4	6	32	7	71	16	65	0	4	9	24
4	1	4	2	10	3	71	8	42	1	9	14	33
5	1	3	1	4	1	23	4	26	2	20	23	50
6	1	2	1	2	1	7	3	16	5	28	7	18
7	1	1	1	1	1	2	2	9	7	42	3	7
8					1	2	1	7	7	63	2	4
9					1	1	1	3	10	43	1	3
10							1	3	7	19	1	2
11							1	2	8	19	1	1
12							1	1	5	7		
13									6	7		
14									7	7		
15									7	7		

4 Results

Table 1 presents the results. The data shows that the prefix method leads to the fewest suggestions. When only one key is pressed there are on average 23.4 suggestions (maximum 50). With two characters the maximum is reduced to 14 suggestions with a mean of 4.7. With three characters there is a maximum of 4 suggestions and a mean of 1.7 suggestions. In other words, the prefix method allows most stations to be entered with three keystrokes or less, although with no tolerance for errors.

Abbreviations result in more than twice as many suggestions with one key compared to the prefix method, and with three characters the mean number of suggestions is reduced to a manageable 6.1 (maximum 32). With four keystrokes this is reduced to a maximum of 10 suggestions with a mean of 2.1 suggestions. The freedom to construct a word from abbreviations comes at the expense of more suggestions.

Compensating for motor errors by incorporating neighboring keys gives a maximum of 322 suggestions (mean 81) with just one keypress. One needs to enter six characters to safely arrive at a maximum of 7 suggestions (mean of 1.2). This approach requires a larger keystroke sample to determine the word. Thus, the number of keystrokes is traded for decreased precision requirement.

With cognitive error compensation the number of alternatives with one input character gives a maximum of 202 suggestions (mean 97). This method converges slower and 7 keystrokes are needed to achieve a maximum of 9 suggestions (mean 1.8).

The combined method is different from the other methods in that it requires all the characters in the word to be entered. There are no entries for one keystroke since there are no one-letter station names, but there are a couple of two-letter stations. The mean number of suggestions is largest for words with 9 characters, although this mean is less than 10. The maximum number of suggestions of 63 is associated with 8 keystroke sequences.

An approach based on Levenshtein distances were included for completeness, where the string was compared to the prefixes of all words with the same length and all words with a Levenshtein distance less than 4 were included in the suggestion list. The levenshtein distance incorporates insertions, deletions and insertions and can thus compensate for both motor and cognitive errors. The Levenshtein approach works well with short distances but gives many suggestions for 4 and 5 keystrokes (maximum of 33 and 50 suggestions, respectively). However, users probably expect to have a reliable suggestion by the third keystroke and this approach appears to be quite a good balance as well as its simple implementation.

The robustness to both motor error and cognitive errors comes at the expense of longer suggestion lists. However, for someone who is very uncertain about how to input a word may find this method useful.

5 Conclusions

Several strategies for making the input of destinations on self-service kiosks and mobile apps have been discussed, namely fuzzy key, abbreviations, bag-of-letters, fuzzy bag and Levenshtein distances. The flexibility and robustness to input errors comes at the expense of more input in terms of keystrokes. Overall, suggestions based on Levenshtein differences appears to provide a good balance of simplicity, few keystrokes and robustness to errors.

References

1. Dickinson, J. E., Ghali, K., Cherrett, T., Speed, C., Davies, N., Norgate, S.: Tourism and the smartphone app: Capabilities, emerging practice and scope in the travel domain. Current Issues in Tourism **17**, 84-101 (2014)
2. Huang, Y. P., Chang, Y. T., Sandnes, F. E.: Ubiquitous information transfer across different platforms by QR codes. Journal of Mobile Multimedia **6**, 3-13 (2010)
3. MacKenzie, I. S., Buxton, W.: Extending Fitts' law to two-dimensional tasks. In: Proceedings of the SIGCHI conference on Human factors in computing systems, pp. 219-226, ACM (1992)
4. Eika, E.: Universally designed text on the web: towards readability criteria based on antipatterns. Stud. Health Technol. Inform. **229**, 461–470 (2016)
5. Eika, E., Sandnes, F.E.: Authoring WCAG2.0-compliant texts for the web through text readability visualization. In: Antona, M., Stephanidis, C. (eds.) UAHCI 2016. LNCS, vol. 9737, pp. 49–58. Springer, Cham (2016)
6. Berget, G., Sandnes, F.E.: Do autocomplete functions reduce the impact of dyslexia on information searching behaviour? A case of Google. J. Am. Soc. Inf. Sci. Technol. **67**, 2320–2328 (2016)
7. Whitney, G., Keith, S., Bühler, C., Hewer, S., Lhotska, L., Miesenberger, K., Sandnes, F.E., Stephanidis, C., Velasco, C.A.: Twenty five years of training and education in ICT Design for All and Assistive Technology. Technology and Disability **3**, 163-170 (2011)
8. Sandnes, F.E.: What do low-vision users really want from smart glasses? Faces, text and perhaps no glasses at all. In: Miesenberger, K., Bühler, C., Penaz, P. (eds.) ICCHP 2016. LNCS, vol. 9758, pp. 187–194. Springer, Cham (2016).
9. Gomez, J.V., Sandnes, F.E.: RoboGuideDog: guiding blind users through physical environments with laser range scanners. Procedia Comput. Sci. **14**, 218–225 (2012)
10. Sandnes, F.E., Tan, T.B., Johansen, A., Sulic, E., Vesterhus, E., Iversen, E.R.: Making touch-based kiosks accessible to blind users through simple gestures. Universal Access in the Information Society **11**, 421-431 (2012)
11. Lin, M.W., Cheng, Y.M., Yu, W., Sandnes, F.E.: Investigation into the feasibility of using tactons to provide navigation cues in pedestrian situations. In: Proceedings of the 20th Aus-

tralasian Conference on Computer-Human Interaction: Designing for Habitus and Habitat, pp. 299-302. ACM (2008)
12. Liljander, V., Gillberg, F., Gummerus, J., Van Riel, A.: Technology readiness and the evaluation and adoption of self-service technologies. Journal of Retailing and Consumer Services **13**, 177-191 (2006)
13. Petrie, H., Darzentas, J. S., Power, C.: Self-Service Terminals for Older and Disabled Users: Attitudes of Key Stakeholders. In: International Conference on Computers for Handicapped Persons, pp. 340-347, Springer (2014)
14. Day, P., Rohan, C., Coventry, L., Johnson, G., Riley, C.: Reach modelling for drive-up self-service. In: Contemporary ergonomics and human factors 2010: proceedings of the International Conference on Contemporary Ergonomics and Human Factors 2010. Contemporary Ergonomics. pp. 47-55, Taylor & Francis, London (2010)
15. Darzentas, J., Darzentas, J. S.: Systems thinking in design: service design and self-services. Form Akademisk-forskningstidsskrift for design og designdidaktikk **7**, (2014)
16. Schreder, G., Smuc, M., Siebenhandl, K., Mayr, E.: Age and computer self-efficacy in the use of digital technologies: An investigation of prototypes for public self-service terminals. In: International Conference on Universal Access in Human-Computer Interaction, pp. 221-230, Springer (2013)
17. Sandnes, F. E., Jian, H. L., Huang, Y. P., Huang, Y. M.: User interface design for public kiosks: an evaluation of the Taiwan high speed rail ticket vending machine. Journal of information science and engineering **26**, 307-321, (2010)
18. Shiba, Y., Sasakura, M.: Visual Interface and Interaction Design for self-service orders at a Restaurant. In: 2016 20th International Conference Information Visualisation, pp. 230-235, IEEE (2016)
19. van Schaik, P., Petrie, H., Japp, J.: The ATM speaks: the design and evaluation of an automatic teller machine with voice output. Advancement of Assistive Technology **3** (1997)
20. Lyu, Y., Vincent, C. J., Chen, Y., Shi, Y., Tang, Y., Wang, W., Liu, W., Zhang, S., Fang, K., Ding, J.: Designing and optimizing a healthcare kiosk for the community. Applied ergonomics **47**, 157-169 (2015)
21. Wright, P., Soroka, A., Belt, S., Pham, D. T., Dimov, S., De Roure, D., Petrie, H.: Using audio to support animated route information in a hospital touch-screen kiosk. Computers in human behavior **26**, 753-759 (2010)
22. Siebenhandl, K., Schreder, G., Smuc, M., Mayr, E., Nagl, M.: A user-centered design approach to self-service ticket vending machines. IEEE Transactions on Professional Communication **56**, 138-159 (2013)
23. Azad, A., Ruiz, J., Vogel, D., Hancock, M., Lank, E.: Territoriality and behaviour on and around large vertical publicly-shared displays. In: Proceedings of the Designing Interactive Systems Conference, pp. 468-477, ACM (2012)
24. Min, X. G., Li, G. Z., Wan, T.: Improve on the ticket vending machine for the railway. In: Advanced Materials Research **422**, pp. 35-38, Trans Tech Publications (2012)
25. Tikka, H., Viña, S., Jacucci, G., Korpilahti, T.: Provoking the city—touch installations for urban space. Digital Creativity **22**, 200-214 (2011)
26. Schreder, G., Siebenhandl, K., Mayr, E., Smuc, M.: The Ticket Machine Challenge: Social Inclusion by Barrier-free Ticket Vending Machines. Generational Use of New Media, 129-148 (2012)
27. Hagen, S., Sandnes, F. E.: Toward accessible self-service kiosks through intelligent user interfaces. Personal and Ubiquitous Computing **14**, 715-721 (2010)
28. Sandnes, F. E.: Effects of common keyboard layouts on physical effort: Implications for kiosks and Internet banking. In The proceedings of Unitech2010: International Conference on Universal Technologies, pp. 91-100, (2010)

29. Sandnes, F. E., Huang, Y. P.: Chording with spatial mnemonics: automatic error correction for eyes-free text entry. Journal of information science and engineering **22**, 1015-1031 (2006)
30. Sandnes, F. E.: Evaluating mobile text entry strategies with finite state automata. In: Proceedings of the 7th international conference on Human computer interaction with mobile devices & services, pp. 115-121. ACM (2005)
31. Sandnes, F.E., Jian, H.L., Pair-wise variability index: Evaluating the cognitive difficulty of using mobile text entry systems. In: International Conference on Mobile Human-Computer Interaction. LNCS, vol. 3160, pp. 347-350. Springer Berlin Heidelberg (2004)
32. Peterson, J. L.: Computer programs for detecting and correcting spelling errors. Communications of the ACM 23, 676-687 (1980)
33. Li, Y., Duan, H., Zhai, C.: A generalized hidden markov model with discriminative training for query spelling correction. In: Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval, pp. 611-620, ACM (2012)
34. Duan, H., Hsu, B. J. P.: Online spelling correction for query completion. In: Proceedings of the 20th international conference on World wide web, pp. 117-126, ACM (2011)
35. Bruck, M.: The word recognition and spelling of dyslexic children. Reading Research Quarterly, 51-69 (1988)
36. Kane, S. K., Wobbrock, J. O., Harniss, M., Johnson, K. L.: TrueKeys: identifying and correcting typing errors for people with motor impairments. In: Proceedings of the 13th international conference on Intelligent user interfaces, pp. 349-352, ACM (2008)
37. Kristensson, P. O., Zhai, S.: Relaxing stylus typing precision by geometric pattern matching. In Proceedings of the 10th international conference on Intelligent user interfaces, pp. 151-158, ACM (2005)
38. Sandnes, F. E.: Reflective text entry: a simple low effort predictive input method based on flexible abbreviations. Procedia Computer Science **67**, 105-112 (2015)