PREDICTIVE KEYBOARD DESIGN STUDY: EFFECTS OF WORD POPULATIONS, NUMBER OF DISPLAYED LETTERS, AND NUMBER OF TRANSITIONAL PROBABILITY TABLES

James R. Lewis
International Business Machines Corp.
West Palm Beach, Florida

David J. Allard
International Business Machines Corp.
Ft. Lauderdale, Florida

Harold D. Hudson
International Business Machines Corp.
Raleigh, North Carolina

Predictive keyboards are software keyboards that, to conserve screen real estate, display a subset of the full set of alphabetic keys at any one time, predicting the letters to display on the basis of tables of letters' transitional probabilities. Using different types of test texts (names, words, random strings), we evaluated the influence of various manipulations on the efficiency of letter selection with a predictive keyboard. The results of this study indicated that, across tested text types (names, words, random strings), (1) there was little benefit gained from adding a fourth transitional table modeling the likelihood of a letter's use as a function of a space and two preceding letters, (2) there was a potential benefit from increasing the number of displayed letters from six to eight, and (3) for a personal communicator device, an adaptive strategy would probably be less effective than using multiple sets of letter probability tables.

INTRODUCTION

As touchscreen interfaces become smaller, emulating keyboard input becomes more difficult. One solution to on-screen keyboard emulation is a predictive keyboard -- one that only displays a most-likely subset of the set of keyboard keys. This was one of the keyboard input solutions provided in the Simon™ personal communicator (Lewis, 1996), developed by IBM for BellSouth Corp. IBM Research developed the fundamental algorithms for the predictive keyboard, and the development team built the user interface for its deployment in Simon.

The Importance of Minimizing "Others"

The predictive keyboard used tables of transitional letter probabilities to guide its predictions. The software used these tables to arrange the letters of the alphabet by most-likely probability of occurrence, and displayed the six letters that it predicted as the most likely for the user to type next. If the desired letter was not in the initial set, users could touch an "Other" button to display the set of six next most-likely letters.

Some reflection on this reveals the importance of minimizing the number of times users need to touch the "Other" button when inputting text. Visual search over a set of letters is relatively fast, especially when the letters are grouped together in a small space, as is the case with the types of small devices for which a designer would consider building a predictive keyboard. Determining that the desired letter is not in the current display, touching the "Other" button, and waiting for the next set of letters to appear is relatively slow and likely to be frustrating. Thus, it is important to investigate ways to reduce the likelihood that a user will have to touch the "Other" button.

Aspects of Predictive Keyboard Design Potentially Affecting the Likelihood of Touching "Other"

The use of an adaptive strategy. As with other predictive technologies (for example, handwriting and speech recognition), the probabilities in the tables were derived from the analysis of a large corpus of training texts. Simon's training texts were, for the most part, composed of normal English words used in sentences. As designed for the Simon personal communicator, the contents of these tables stayed the same regardless of users' text inputs.

It is possible to adjust the probabilities in these tables over time as a function of the text that a user types, but this adaptive strategy would be useful only if most of the text that a user types comes from the same text population. If the text that a user types comes from different text populations, then an adaptive strategy could result in predictive tables that are suboptimal for any specific text population. For Simon, we expected people to use the predictive keyboard to type small amounts of normal text when creating e-mail and to type proper names when creating address book entries.

Thus, one issue in the design of predictive keyboards is whether the kinds of text that users type come from one or more text populations. More specifically, do the differences in the text populations affect the number of times users would have to touch an "Other" button when inputting text? If so, it would be better to base prediction on multiple sets of tables, each derived from different text populations (different sets of training data), with the active set dynamically controlled by the application to optimize predictive accuracy.

The number of keys displayed. A second issue is the number of keys to display in a predictive keyboard. The more keys displayed, the more likely it is that the desired letter will be immediately available. However, displaying more keys means potentially more visual search demand on the user (because there are more letters to search among) and, with the amount of real estate available for the predictive keyboard remaining constant, increased motor skill demand (because the buttons must be smaller). Therefore, unless displaying more keys substantially improves the likelihood of displaying the desired letter, it would be better to display a smaller number of keys. The user interface for Simon displayed six letters, but could potentially have supported the presentation of eight letters in the space available.

Number and types of transitional probability tables. A third issue is the number of tables used to predict the most likely letters. The fundamental table for prediction contained the probability of a letter given the immediately preceding two letters (with no spaces). An important secondary table identified the probability of a letter given an immediately preceding space (in other words, the likelihood of a letter's occurrence at the beginning of a word). Other transitional probability tables developed to improve prediction included one that contained the probabilities of a letter given a preceding space and one letter, and one for a preceding space and two letters.

Theoretically, the more tables used, the better the prediction should be. However, using more tables places a greater demand on a device's memory and potentially increase the time required to determine the set of letters to present, and this is especially true for tables that take into account two or more characters preceding the predicted character. Therefore, unless additional tables provide a substantial improvement in prediction (specifically, provide a measurable reduction in the frequency of requiring users to touch the "Other" button), it would be more efficient to use a smaller number of them, especially for devices with limited memory and/or processing power.

Design Issues Investigated in this Study

This paper describes an experiment designed to investigate three design issues for predictive keyboards. (1) Do the types of text that users might reasonably input with a predictive keyboard come from the same or different text populations as measured by the frequency with which users would need to touch an "Other" button when inputting sets of test texts? (2) Does displaying eight keys rather than six keys substantially increase the likelihood that the desired next key will be immediately available for typing (in other words, reduces the frequency of touching the "Other" button)? (3) Does using the additional predictive table (modeling letter usage as a function of a space and two additional letters) substantially decrease the likelihood of requiring the user to touch an "Other" button?

METHOD

To acquire test texts for this experiment, we randomly selected 100 names (first and last) from a local telephone book, 100 words from a dictionary, and 100 words from an English-language novel. As a control, we constructed 100 random 5-letter strings. (Note that these were test texts obtained independently from the training texts used to create the transitional probability tables at IBM Research – not training texts, which would typically contain many more words than a set of test texts.) We varied the number of tables used by the predictive keyboard from three to four (without and with the table for a space and two letters), and varied the number of displayed letters from six to eight.

We simulated user input by developing a program based on the Simon predictive keyboard that could accept lists of words stored in files, so users did not actually have to type the words into the predictive keyboard. This program entered the word lists for the various test texts into the predictive keyboard, kept track of the number of times a user would have to have pressed the "Other" key to display the desired letter, and provided as output the average "Others" per character for each tested word.

RESULTS

Table 1 shows the means and 95% confidence intervals for each combination of conditions in the experiment. The results of an analysis of variance showed statistically significant effects for all terms in a complete linear model representing this experimental design (see Table 2). We dropped the random set from the analysis of variance because the means from the random set were clearly different from the other sets, and including them in the analysis obscured other more meaningful comparisons among the more realistic text sets due to the large amount of measurement variability introduced by this set. The design included one between-words variable (word source) and two within-words variables (number of displayed letters and number of transition probability tables used), and all their interactions. The words themselves were a random variable; all others were fixed.

The three-way interaction (word source by number of displayed letters by number of transition probability tables used) was significant ($F(3,396) = 4.27$, $p=.006$). As shown in Figure 1, the manipulations of number of letters displayed and number of tables used affected all text types in the same direction, but with differing magnitudes.

The words from the standard text in the novel required the fewest "Other" button presses, and names (both last and first) required more. Only the standard text from the novel seemed to benefit substantially from having four rather than three transition probability tables. All text types benefited from increasing the number of displayed letters, although the standard text from the novel benefited the least.

Table 1. Experimental Results (Mean "Other" Presses per Letter, and Upper and Lower Limits of a 95% Confidence Interval Around the Average)

		Six Letters Displayed		Eight Letters Displayed	
		Tables Used		Tables Used	
Word Source	Statistic	Three	Four	Three	Four
Standard Text	Upper	0.53	0.42	0.32	0.26
	Mean	**0.45**	**0.34**	**0.26**	**0.20**
	Lower	0.37	0.27	0.20	0.14
Dictionary	Upper	0.66	0.64	0.40	0.40
	Mean	**0.58**	**0.56**	**0.33**	**0.33**
	Lower	0.40	0.49	0.27	0.27
First Names	Upper	0.85	0.81	0.57	0.53
	Mean	**0.76**	**0.73**	**0.50**	**0.47**
	Lower	0.68	0.66	0.44	0.41
Last Names	Upper	0.82	0.78	0.51	0.47
	Mean	**0.74**	**0.70**	**0.44**	**0.41**
	Lower	0.65	0.62	0.38	0.35
Random Strings	Upper	2.08	2.09	1.45	1.46
	Mean	**1.96**	**1.97**	**1.35**	**1.36**
	Lower	1.84	1.85	1.25	1.26

Table 2. Analysis of Variance on Experimental Results

Source of Variance	Sums of Squares	df	Mean Square	F-Score	P(>F)
Between Words					
Word Source	22.47	3	7.49	16.32	0.000
Words w/Groups	181.75	396	0.46		
Within Words					
Letters Displayed	22.96	1	22.96	659.07	0.000
Letters Displayed x Word Source	0.84	3	0.28	8.08	0.000
Letters Displayed x Words w/Groups	13.79	396	0.03		

Table 2. Analysis of Variance on Experimental Results (cont.)

Source of Variance	Sums of Squares	df	Mean Square	F-Score	P(>F)
Tables Used	0.56	1	0.56	19.96	0.000
Tables Used x Word Source	0.32	3	0.11	3.84	0.01
Tables Used x Words w/Groups	11.12	396	0.03		
Letters Displayed x Tables Used	0.03	1	0.03	9.97	0.002
Letters Displayed x Tables Used x Word Source	0.03	3	0.01	4.27	0.006
Letters Displayed x Tables Used x Words w/Groups	1.06	396	0.003		

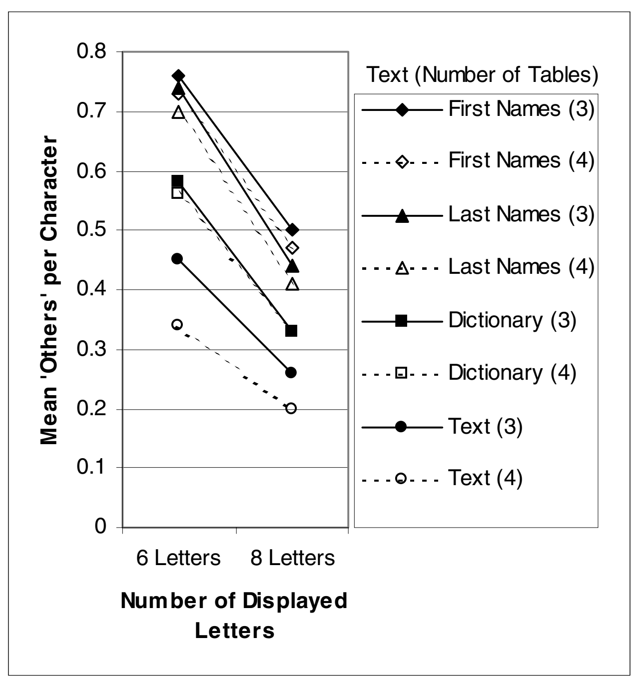

Figure 1. The Letters Displayed by Tables Used by Word Source Interaction

DISCUSSION

The purpose of this experiment was to investigate three issues related to the design of predictive keyboards. (1) Do the types of text that users might reasonably input with a predictive keyboard come from the same or different text populations as measured by the frequency with which users would need to touch an “Other” button when inputting sets of test texts? (2) Does displaying eight keys rather than six keys substantially increase the likelihood that the desired next key will be immediately available for typing (in other words, reduces the frequency of touching the “Other” button)? (3) Does using the additional predictive table (modeling letter usage as a function of a space and two additional letters) substantially decrease the likelihood of requiring the user to touch an “Other” button?

The statistically significant differences among the different text types (word sources) indicate that these words represent different word populations, with differences sufficient to affect the likelihood that a user would have to touch the “Other” button when typing from these sources. This suggests that an adaptive strategy for transition probability tables would be less effective than a strategy using multiple sets of tables because a single set of tables cannot represent the different text populations as effectively as multiple sets. This is especially important if users will type both normal text and names (names of people and names in addresses), as they would using a device like a Simon personal communicator.

In practice, it is important to balance the desire for maximized prediction accuracy with a device's memory constraints. Smaller devices might not have enough memory to accommodate multiple sets of tables. In this case, an adaptive strategy might be better than a multiple-tables strategy. If adaptation is not feasible, then the tables should appropriately reflect the kinds of text that users might type with the device (such as normal text, names, and addresses).

There is a potential benefit from increasing the number of displayed letters from six to eight, which resulted in an average 40% decrease in the number of "Others" per letter. The words from the novel received the least raw benefit from increasing the number of displayed letters. This is reasonable because the original source for the tables was standard text, so the current tables should be better at predicting letters in standard text relative to words from other sources. Therefore, the correct letter is more likely to appear in the first six letter choices for standard text, reducing the benefit of displaying two additional letters for standard text relative to other text sources. The gain for standard text was large enough (34.9% fewer "Others" per letter) to justify further investigation of the feasibility of increasing the number of displayed letters in the predictive keyboard. However, this change would require testing with users to ensure that the additional demands from increased visual search and motor skill requirement do not cancel the increased likelihood of having the desired character available immediately. Our experiences with the Simon suggest that increasing the number of letters displayed would maximize the tradeoffs among these factors. Determining the point at which increasing the number of displayed letters no longer improves performance would require additional research.

The results indicate that, across all word sources, there is little benefit gained from including the additional transition probability table. Only the normal text showed a substantial reduction in "Others" per letter (from .355 to .272, a reduction of 23%). The average reduction for the other text types was only 4%. However, because the additional table did improve predictive accuracy for the kind of text from which it was derived, it would be worthwhile to re-evaluate the value of the extra table given additional tables derived from training sets based on the different word sources. As with the issue of an adaptive strategy versus multiple table sets, dropping the extra table is reasonable if its inclusion places too much demand on limited system resources. Otherwise, it is reasonable to include it in a predictive keyboard system.

ACKNOWLEDGMENTS

The authors thank the three anonymous reviewers who provided comments for this paper. We believe that their comments and questions helped us improve the clarity of this paper. Of course, any remaining lack of clarity is entirely our fault.

TRADEMARKS

REFERENCES

Lewis, J. R. (1996). Reaping the benefits of modern usability evaluation: The Simon story. In G. Salvendy and A. Ozok (eds.), *Advances in Applied Ergonomics: Proceedings of the 1st International Conference on Applied Ergonomics -- ICAE '96* (pp. 752-757). Istanbul, Turkey: USA Publishing.