Language Modeling for Soft Keyboards

Joshua Goodman
Microsoft Research
Redmond, WA 98052
joshuago@microsoft.com
www.research.microsoft.com/~joshuago

Gina Venolia
Microsoft Research
Redmond, WA 98052
ginav@microsoft.com

Keith Steury
Microsoft Corp.
Redmond, WA 98052
keithste@microsoft.com

Chauncey Parker
Univ. of Washington
Seattle, WA 98195
chaunce@u.washington.edu

ABSTRACT
Language models predict the probability of letter sequences. Soft keyboards are images of keyboards on a touch screen for input on Personal Digital Assistants. When a soft keyboard user hits a key near the boundary of a key position, the language model and key press model are combined to select the most probable key sequence. This leads to an overall error rate reduction by a factor of 1.67 to 1.87. An extended version of this paper [4] is available.

Keywords
Language model, soft keyboard, corrective keyboard

INTRODUCTION
Language models give probabilities of word or letter strings They are critical in applications such as speech recognition, for distinguishing phrases like "recognize speech" and "wreck a nice beach." A language model giving the relative probabilities helps correctly distinguish the two.

A soft keyboard is an image of a keyboard shown on a touch sensitive screen and used with a stylus. Soft keyboards are perhaps the fastest technique for entering information into Personal Digital Assistants (PDAs) such as Palm Pilots. Soft keyboard displays tend to be small, and, since speed and accuracy are inversely related [1], errors on soft keyboards are fairly common.

We propose a novel use for language models, reducing errors on soft keyboards. We call such keyboards *corrective keyboards*. Imagine that a user taps the center of the "q" key, and then taps the boundary of the "u" and "i" keys. Given the likelihood of "u" following "q", it is a better bet that the user intended the "u" than the "i". Similarly, imagine that the user taps the boundary between "q" and "w", and then taps the center of "u". The intended sequence was probably "qu." The relative probabilities of letter sequences help resolve the intended key sequences. We can extend this reasoning to cases when the tap is not on a boundary. If the user taps "q" centrally, and then taps inside "i" but near "u", the intended sequence was probably "qu". By combining the probability of intending to hit "u" with the probability of the actual pen down location, we can find the most likely key sequence.

This is the short version of the paper. An extended version [4] describes the models, experiments, language modeling, and other related work in much more detail.

IUI'02, January 13-16, 2002, San Francisco, California, USA.

MATHEMATICAL MODELS
We desire the most likely intended letter sequence given the observed pen down positions. Using Bayes law, we get

$$\arg\max_{\text{letter sequences}} P(\text{letter sequence} \mid \text{pen down positions}) =$$

$$\arg\max_{\text{letter sequences}} \frac{P(\text{letter sequence}) \times P(\text{pen down positions} \mid \text{letter sequence})}{P(\text{pen down positions})} =$$

$$\arg\max_{\text{letter sequences}} P(\text{letter sequence}) \times P(\text{pen down positions} \mid \text{letter sequence})$$

We can use standard language modeling techniques [3] to find the probability of a letter sequence $L_1, L_2 \ldots, L_n$:

$$P(L_1, L_2, ..., L_n) = P(L_1) \times P(L_2 \mid L_1) \times ... \times P(L_n \mid L_1 ... L_{n-1})$$

Next, make an approximation, such as

$$P(L_i \mid L_1 ... L_{i-1}) \approx P(L_i \mid L_{i-6} ... L_{i-1})$$

That is, assume that the probability of letter L_i is independent of the probabilities of letters more than 6 back. Compute $P(L_i|L_{i-6}...L_{i-1})$ by counting the number of occurrences of $L_{i-6}...L_{i-1}$ and $L_{i-6}...L_i$ in training data. We smooth this estimate by combining it with other, more robust estimates, so that any possible letter sequence has a reasonable probability. See [3] for a language modeling introduction.

We also need the probability distribution of pen down positions given letter sequences, an area not previously studied. Many factors were *not* helpful. These included modeling the correlation between position at one time and the previous time; using information from pen up position; modeling center offset correlations; and modeling correlation between typing speed and variance or mean position.

We did find three factors that *were* helpful, which we used to form better models. First, the average position for each key was not the key's center, but was slightly shifted down and towards the center of the keyboard. Second, variances in the x and y position are typically different. Third, some distributions are rotated, rather than being perpendicular to the keyboard axes; that is, there is noticeable covariance between the x and y coordinates. Based on these observations, we used a bi-variate Gaussian distribution, with the means, variances, and covariance computed

separately for each key. The probability of a sequence of positions is the product of the individual probabilities.

RESULTS

We performed a user study with 8 participants. Users typed 1000 "net" characters from a set of short prompts, where a net character is the number of characters correctly typed, minus the number of mistakes. For users to finish as quickly as possible, they needed to type with both reasonable accuracy and speed. We ran three sets of pairs of conditions per user. The first set, Set 0, consisted of 5 prompts for each of corrective and non-corrective keyboards. The following sets, Set 1 and Set 2, consisted of enough prompts for the users to complete 1000 net characters in each condition. Corrective and non-corrective conditions were presented alternately. We balanced for order and counterbalanced by gender. The subjects were told for each set whether they were using the corrective keyboard, so that they would know they could type outside key boundaries with the corrective version. 6 different prompt sets were used, and in the same order, so that a given group of prompts was used 4 times in the corrective condition, and 4 times in the non-corrective condition.

While on average, throughput was marginally faster with the corrective keyboard, the differences were not statistically significant. We compared the geometric means of the error rates. In order to check for learning effects, we compared both the first full set, the second full set, and results from pooling the two sets.

Set	ratio	p<
1	1.87	0.014
2	1.87	0.028
Both	1.72	0.005

Table 1: Error ratios and significance for various measurement methods

There were always significantly fewer errors with the corrective keyboard than with the non-corrective keyboard: between a factor of 1.72 and a factor of 1.87. Alternative methods of measuring error rate described in the full paper gave ratios as low as 1.67, which is still impressiveUsers were also given a short questionnaire to answer [4]; they consistently preferred the corrective keyboard.

DISCUSSION

There has been much previous work in text entry, but little like this. Fitts' law [1] looks at the relationship between accuracy, speed and distance, but does not give a probability distribution. A different approach to optimizing soft keyboards is redesign of the layout [6]. The work most similar to ours is work on spelling correction using probabilistic models, although the exact pen down position is ignored. In other work [7], a simple language model was applied to decoding eye-traces for use in an eye-typing input method.

Language models have been used with keyboard input when trying to disambiguate truly ambiguous input sequences [2]. In that work, a ten key keypad is used, with each key corresponding to several letters, so that each input sequence is ambiguous, and a language model is essential for choosing among the possible letter sequences. Language models can be used for text entry using the numeric keypad of a cellular phone [5]. Note, however, that commonly deployed systems, such as T9 (www.tegic.com) use a dictionary sorted by word frequency, rather than a language model.

Our results are very promising. They show that use of language models significantly reduces error rates in a soft keyboard task, by a factor of 1.67 to 1.87. These techniques could be applied to a variety of other input methodologies, such as pie menus for text input, or with eye-tracking systems [7]. Further research remains. We want to try longer tests, in order to examine learning effects. We would like to try user-specific models, which have been very helpful in speech recognition. It would be interesting to try more complex language models.

Overall, we are very pleased with the large error rate reductions we have found, and look forward to further progress applying language models to soft keyboards, as well as applying these techniques in other areas.

ACKNOWLEDGMENTS

Thanks to X.D. Huang, D. Jacoby, and Mary Czerwinski.

REFERENCES

1. Fitts, P.M. The information capacity of the human motor system in controlling the amplitude of movement, *Journal of Experimental Psychology 47* (1954), pp. 381-391.

2. Goldstein, M., Book, R. Alsiö, G., Tessa, S. Non-keyboard touch typing: a portable input interface for the mobile user. *Proceedings of CHI '99*, Pittsburgh, pp. 32-39.

3. Goodman, J. Putting it all together: language model combination, in *ICASSP-2000*, Istanbul, June 2000.

4. Goodman, J., Venolia, G., Steury, K., Parker, C. Language Modeling for Soft Keyboards. Microsoft Research technical report MSR-TR-2001-118, available from http://www.research.microsoft.com/pubs.

5. MacKenzie, I.S., Kober, H., Smith, D., Jones, T., Skepner, E. LetterWise: Prefix-based disambiguation for mobile text input. *Proceedings of the ACM Symposium on User Interface Software and Technology - UIST 2001*. New York.

6. MacKenzie, I.S. and Zhang, X.S. The Design and Evaluation of a High-Performance Soft Keyboard, *Proceeding of CHI '99*, Pittsburgh, pp. 25-31.

7. Salvucci, D.D. Inferring intent in eye-based interfaces: tracing eye movements with process models. Proceedings of CHI '99, Pittsburgh, pp. 254-261.