Improving Dictionary-Based Disambiguation Text Entry Method Accuracy

Jun Gong
College of Computer & Information Science
Northeastern University
Boston, USA, 02115
gjoliver@ccs.neu.edu

Carole D. Hafner
College of Computer & Information Science
Northeastern University
Boston, USA, 02115
hafner@ccs.neu.edu

Peter Tarasewich
College of Computer & Information Science
Northeastern University
Boston, USA, 02115
tarase@ccs.neu.edu

Scott I. MacKenzie
Department of Computer Science
York University
Toronto, Ontario, Canada, M3J 1P3
smackenzie@acm.org

CHI 2007, April 28–May 3, 2007, San Jose, California, USA.
ACM 978-1-59593-642-4/07/0004.

Abstract
Text entry on mobile devices is problematic because of ever-decreasing device sizes. Dictionary-based keypad text entry methods are relatively effective, but still run into problems of word ambiguity, especially when used with small numbers of keys. Common text entry disambiguation methods only use word frequency information to resolve conflicts. This paper proposes a new method that also looks at semantic information (distances between word meanings). Simulations show encouraging results, suggesting potential practical applications of this method to mobile devices.

Keywords
Mobile device text entry, disambiguation, dictionary, semantic context

ACM Classification Keywords
H5.m. Information interfaces and presentation (e.g., HCI): User Interfaces.

Introduction

Although much communication in the mobile environment is achieved through voice, pictures, and even video, text entry remains an important part of human-computer interaction with mobile devices because of popular services such as text messaging.

Keypads are common on small mobile devices where there is insufficient room for a full-sized QWERTY keyboard. Besides the international standard mobile phone keypad, which distributes 3 or 4 letters across 8 keys, devices that use very few (3-4) keys are attracting more and more attentions from HCI researchers [1, 5].

Dictionary-based predictive disambiguation (DBPD) methods, such as T9™, enable users to press a key once for each desired letter. Any key sequences pressed are matched to those in a dictionary. The matching word with the highest frequency of occurrence is displayed. If there is more than one matching word, users cycle through the choices by pressing a special "next" key. Importantly, with a limited number of keys, the number of words matching a given keystroke sequence may become very large, requiring frequent presses of the "next" key.

To improve the usability of DBPD when used with small keypads, an algorithm that combines a co-occurrence based semantic language model with a disambiguation algorithm is proposed. Results from initial experimental simulations using a large word corpus are reported and discussed.

Background

Past research on DBPD optimized performance by creating keypad designs (mapping letters to keys) that reduce ambiguity (i.e., keystroke sequences that correspond to multiple words) [3]. But the resulting placement of letters on keys was, in essence, random. Recent work [2] placed alphabetical constraints on key mappings. Results showed that the alphabetically constrained designs provided good novice usability, ease of learning, and improved disambiguation efficiency.

Besides remapping keys to reduce the number of ambiguous words, algorithms based on language models can guess the "correct" word that a user desires. One way to do this is by using the word with highest frequency of appearance among all matching words. Past studies [2] suggested that word frequencies work well for disambiguating keystroke sequences with the 8-key standard mobile phone keypads, as only 1.5% of all words inputted need to be manually disambiguated. However, difficulties arise if fewer keys are used. For example, about 28% of all words need to be manually chosen with an optimized 3-key keypad.

With the shrinking size of mobile devices, as well as the need for specialized devices for disabled users, text entry using fewer keys is a worthwhile research endeavor. Dunlop [1] described a DBPD text entry system implemented on a watch with five keys. MacKenzie [5] compared several text entry

techniques that used three keys. But both methods predict relatively low text entry rates.

However, other research [e.g., 4] has shown that better disambiguation performance is achievable if a higher ordered N-Gram model or semantic information model is utilized. Therefore, the new method proposed here uses semantic information to help disambiguate keystroke sequences with multiple matching words. "Semantic relatedness" is defined as entities that are likely to co-occur [6]. A simpler semantic relatedness model might be based on words' co-occurrences in a large text corpus. Such a statistical solution is potentially easier to obtain and manipulate.

Semantic Relatedness of Word Pairs

Our co-occurrence-based semantic relatedness model (SRM) is defined as follows:

$$SEM(w_1, w_2) = \frac{C(Stem(w_1), Stem(w_2))}{\sqrt{C(Stem(w_1))} \times \sqrt{C(Stem(w_2))}}$$

Where w_1 and w_2 are any two words in the dictionary. $Stem(w_1)$ and $Stem(w_2)$ are the word stems of w_1 and w_2. (A stemming algorithm [7] derives the word stems used here). $C(Stem(w_1))$ and $C(Stem(w_2))$ are the number of times the stems of word w_1 and w_2 occur in the training corpus, respectively. $C(Stem(w_1), Stem(w_2))$ is the number of times the stems of both words w_1 and w_2 occur in the same arbitrarily defined contexts in a training corpus. $SEM(w_1, w_2)$ denotes the co-occurrence based semantic relatedness between w_1 and w_2.

Note that relatedness is built on word stems. This reduces the size of the model, and can help if sparse training data are a potential problem.

Disambiguation Method Combining Semantic Relatedness and N-Gram Models

In the following section, a new "context-based" predictive disambiguation algorithm is proposed. The purpose is to find the most probable matching word based on the semantic context available.

Definitions:

- w_i : A particular word in the dictionary.
- ks_i : The keystroke sequence of w_i.
- $match(ks_i) = \{w_1^{w_i}, w_2^{w_i}, ..., w_n^{w_i}\}$: The set of words that share the same keystroke sequence with w_i. Note that $w_i \in match(ks_i)$.
- $w_j^{w_i} \in match(ks_i)$: A word that shares the same keystroke sequence with w_i.
- $Freq(w_j^{w_i})$: The normalized frequency of $w_j^{w_i}$ among all the words in the set of $match(ks_i)$. Note that $\sum_{w_j^{w_i} \in match(ks_i)} Freq(w_j^{w_i}) = 1$. If w_i is not ambiguous, $match(ks_i)$ should only have one element $w_1^{w_i} = w_i$, therefore $Freq(w_1^{w_i}) = 1$.

Description of the Disambiguation Algorithm

The inputs to the disambiguation algorithm are the keystroke sequence ks_i of a word w_i and a set H_i of context words. The content of H_i depends on the arbitrarily defined context for the word ks_i. (E.g. H_i may contain all the inputted words preceding w_i). Note that only words already inputted and preceding the current target word can be used in H_i.

The output of the disambiguation algorithm is the English word w_i that the disambiguation method produces as the most probable matching word, given ks_i and H_i.

The validity of a word w_i given its history H_i of context words is then defined using:

1. The estimated semantic validity $SV(w_i \mid H_i)$ of a word w_i given H_i

$$SV(w_i \mid H_i) = \prod_{w' \in H_i} SEM(w_i, w')$$

2. The normalized estimated semantic validity $NSV(w_i \mid H_i)$ of w_i given H_i

$$NSV(w_i \mid H_i) = \frac{SV(w_i \mid H_i)}{\sum_{w_j \in match(ks_i)} SV(w_j \mid H_i)}$$

Therefore, the estimated validity $EV(w_i \mid H_i)$ of a word w_i given H_i, is defined as the linear combination of its normalized semantic validity and its normalized frequency:

$$EV(w_i \mid H_i) = \alpha \bullet Freq(w_i) + (1-\alpha) \bullet NSV(w_i \mid H_i)$$

where *a* (with a value between 0 and 1) specifies how much we would like to believe in the frequency component.

Given the definitions above, the disambiguation algorithm is straightforward: it simply returns the word from the candidate list with the highest estimated validity value based on the context. This is formally stated as:

$Disambiguate(ks_i, H_i)$

$return \ \arg\max_{w_k \in match(ks_i)} \{EV(w_k \mid H_i)\}$

End Disambiguate

Experiments

The Reuters corpus [8], composed of news articles, was used to implement the described semantic relatedness model during this initial work. Other corpora, such as the spoken section of BNC corpus, may prove even more valuable for capturing semantic information related to short messages, and will be tested as part of continuing work. Testing used the best 3-key keypad design found in a previous study [2], as shown in Figure 1. The disambiguation performance of using both the semantic model and the word frequencies was compared to that of using only the frequencies.

The optimized three-key keypad was used in the experiment because of the relatively higher proportion (28%) of words that cannot be directly disambiguated by frequencies alone. This leaves more "potential" for our semantic model to show a performance gain.

Key 1:	abcdef
Key 2:	ghijklmn
Key 3:	opqrstuvwxyz

Figure 1. The optimized three-key keypad

The first 2/3 of the entire corpus (~ 4 million sentences) was used to train the semantic relatedness model, and the remaining 1/3 (~ 2.7 million sentences) was used for testing.

To build a vocabulary of reasonable size, any tokens occurring less than 100 times were removed. The remaining 24,109 word tokens form the vocabulary, among which 18.31% were ambiguous words. They accounted for 49.52% of the corpus text.

The semantic relatedness model was then trained, and different *a* values were tested to find the linear combination of the semantic model and frequency model which achieved the best disambiguation performance.

Results

Some example word tokens and the semantic relatedness between them are listed in Table 1.

	DOG	SICK	DIE	DOCTOR
DOG	1	0.000670	0.001067	0.002089
SICK	0.000670	1	0.002269	0.010900
DIE	0.001067	0.002269	1	0.014694
DOCTOR	0.002089	0.010900	0.014694	1

Table 1. Relatedness between Pairs of Example Words

Ten values of α, from 0.1 to 1, in increments of 0.1, were run. Disambiguation accuracies (DA) for different values of α are shown in Figure 2. Simulations were also run to find the smallest average number of keystrokes needed to input a single character (KSPC) with the combined model. These results are shown in Figure 3.

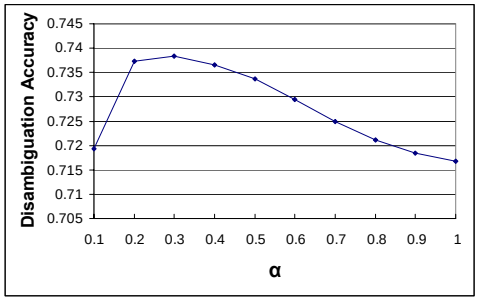

Figure 2. Disambiguation Accuracy with Different *a*

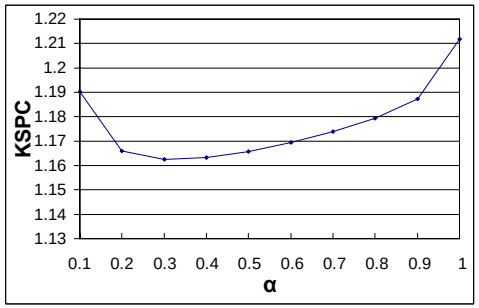

Figure 3. KSPC Values with Different *a*

Discussion

From the statistics of the Reuters corpus, 18.31% of the ambiguous words account for 49.52% percent of the entire corpus, which shows that ambiguous words are more common than non-ambiguous words.

From Figures 2 and 3, it is clear that with $a = 0.3$, the combined method achieves the highest disambiguation accuracy and the lowest KSPC value: 73.84% and 1.16 respectively.

It is interesting that with the combined method, the DA (the chance that a user will get the desired word immediately without pressing "next" key) is only improved from 71.68% to 73.84%, however the KSPC value (which specifies the average number of keystrokes for inputting a character) is reduced much more significantly from 1.21 to 1.16, suggesting that, about 25% of the extra "next" key presses can be saved. This result tells us that many desired words can move up towards the top of their candidate lists, even if they do not become the first choice. This can save a significant amount of user effort.

Finally, the KSPC value of 1.16 demonstrates a promising applicability of improved DBPD text entry methods to "ultra-small" devices with very few keys.

Conclusion and Continuing Work

This paper presents ongoing work on a new method for disambiguating keystroke sequences using a semantic relatedness model based on text co-occurrence information. An algorithm utilizing this model has been implemented and our initial simulations show promising results.

Investigations continue with this new method, particularly into its potential to improve text entry performance with different sized keypads, and into the effect of α values on the model results. Corpora that may be more suitable for capturing the characteristics of text messaging will also be incorporated and tested. Additionally, simple part-of-speech rules will be used with the method to further refine disambiguation candidates. Empirical usability studies are planned to validate the predicted disambiguation improvements found through these simulations. Since the semantic relatedness model will be implemented on real mobile devices, practical issues, including memory and processing speed requirements, will be investigated further at that time.

REFERENCES

[1] Dunlop, D. M. Watch-Top Text-Entry: Can Phone-Style Predictive Text-Entry Work with Only 5 Buttons? In *Proc. Mobile HCI 2004*, (2004), 342-346.

[2] Gong, J., and Tarasewich, P. Alphabetically constrained keypad designs for text entry on mobile devices, In *Proc. CHI 2005*, (2005), 211-220.

[3] Lesher, G.W., Moulton, B.J., and Higginbotham, D.J. Optimal character arrangements for ambiguous keyboards, In *IEEE Trans. on Rehabilitation Engineering*, 6 (1998), 415–423.

[4] Lesher, G., Moulton, B., and Higginbotham, J. Effects of ngram order and training text size on word prediction, In *Proc. of the RESNA99 Annual Conference*. (1999).

[5] MacKenzie, I. S. Mobile text entry using three keys, In *Proc. NordiCHI 2002*, (2002), 27-34.

[6] Manning, C. D., and Schutze, H. *Foundations of Statistical Natural Language Processing*, MIT Press (1999), Cambridge MA.

[7] Porter, M. F. An algorithm for suffix stripping, *Program*, 14, 3 (1980), 130-137.

[8] Reuters Corpora, http://trec.nist.gov/data/reuters/reuters.html (accessed November 30, 2006).