Word Completion Feature that Recognizes Implicit Rejections

TR 29.3812
June 24, 2004

Patrick M. Commarford
James R. Lewis

IBM Pervasive Computing

Boca Raton, Florida

Abstract

This report describes a method that provides users with an improved method for entering data into mobile handheld devices such as cell phones and personal digital assistants (PDAs) by increasing the likelihood of displaying a desired word in a word completion interface. This solution is unique in that it is sensitive to implicit rejection of the word completion candidates.

ITIRC Keywords

Handheld
Implicit rejection
Input
Mobile device
PDA
Word completion

Contents

Introduction

Text input methods for mobile devices are slow. Typically, the greater number of actions (e.g., soft keyboard taps, handwriting strokes, etc.,) a user has to take, the longer each single word entry will take to input. In other words, inputting "rhinoceros" letter-by-letter takes longer than letter-by-letter input of "or."

Current word completion features 1) use some logic to determine a candidate word or set of words, 2) display this word (these words) and 3) provide some mechanism for users to quickly select the desired word (for example, by tapping the word with a stylus). Pocket PC and Sharp Zaurus devices each provide a word completion feature.

Pocket PC allows users to turn this feature on or off, to select the number of letters (1-7) after which the system should suggest word(s), to select the number of words that should be made available (suggested) and to decide whether to automatically add a space after each word. Sharp always displays the maximum number of words, consistent with the starter letters, that can fit in a single line spanning the width of the screen.

When a user taps a next letter that is consistent with the currently displayed word(s), current methods generally continue to display the un-chosen word. In this way, these methods continue to provide users with word choices that they have already implicitly rejected (by choosing not to select them). For example, if Pocket PC users have their system set to display one word choice (the default) and the user enters the letter "P" the word completion feature might return "provide." Now if the user enters "r" the system is likely to continue displaying "provide" even though the user implicitly rejected this word. It is more likely that the user is planning to enter "prevent" or "private" but the system is not helping the user at all because it continues to display "provide." The same is true when the system suggests a set of words. For example, with Pocket PC, we obtained the results listed in Table 1.

Table 1
Sample word completion options using a Pocket PC device

Letter(s) entered	**Word completions choices**
P	Provide, Prove, Process, Provided
Pr	Provide, Prove, Process, Provided
Pro	Provide, Prove, Process, Provided

This is a minor problem (perhaps not even a problem) when people use standard keyboards for character input. Given the slow input rates of current handheld devices, though, this can dramatically degrade potential input times. Experienced handheld device users monitor the word completion candidates closely, looking for any input speed advantage they can find. Current word completion systems, however, fail to take advantage of this potential user strategy.

The present invention provides users with a better chance of reducing input time by displaying more likely-to-be-chosen words in the (set of) word completion choice(s). The primary advantage of this solution is that it is sensitive to implicit rejection of the word completion candidates.

The Solution

This invention increases the speed with which users can input text to mobile devices by replacing implicitly rejected words in a word completion display with the words determined to be next most likely. These replacements can occur with the first implicit rejection, the second, or any subsequent rejection.

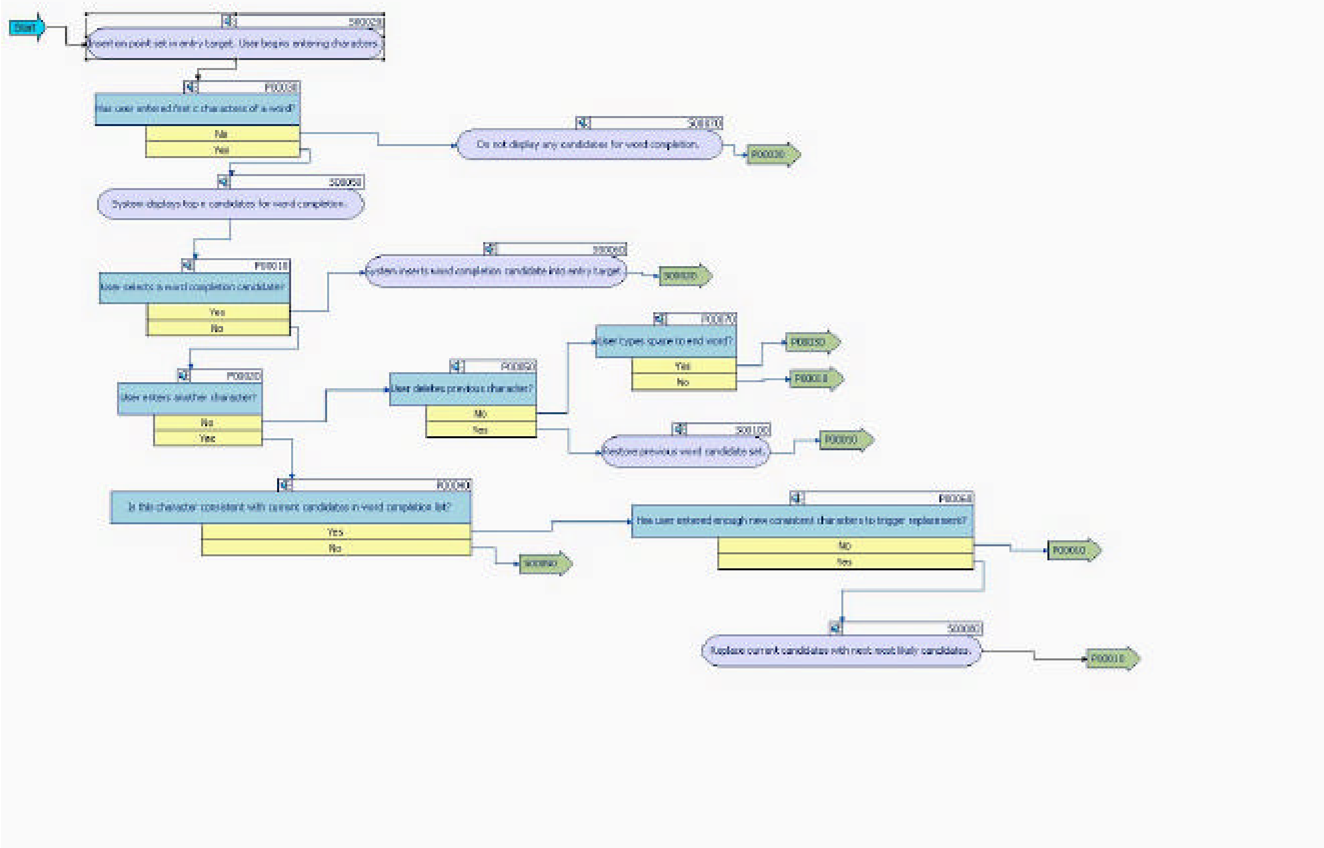
Method 1

The first method assumes that a single instance of not selecting a displayed word indicates that the user is not planning to enter that word. For example, the system might use some logic to determine that when users begin a word with "P" the most likely-to-be-entered word is "provide" and the second most likely word is "prove." Therefore, if the word completion feature is set to display a single word, it will display (and make available for selection), "provide." However, when users next enter "r" they are implicitly rejecting "provide." Therefore, the system would remove, "provide" and display "prove" when users tap the "r." Now, if the user continues by entering "o" instead of selecting "prove" from the word completion feature, the system would assume that neither "provide" nor "prove" are the to-be-entered words and would display the next most likely word beginning with "pro." The same would hold true with word completion features set to display and allow selection of more than one word. For example, assume a user taps "P" and the word completion feature returns provide, prove, process, and provided. Now the user taps "r" implicitly rejecting the four displayed words—the system would replace these four words with the four next most likely words (e.g., providing, prevent, processing, practical). Now, if the user taps "o" all four words would again be replaced (prevent and practical do not start with "pro" and were implicitly rejected; providing and processing have the correct starting letters, but have been rejected).

Method II

The second method assumes that a single instance of not selecting a displayed word indicates that a) the user is not planning on entering that word or b) the user was in the process of tapping the second letter before he/she noticed the displayed word. This method gives the user a second chance to select a particular word for which a consistent letter was chosen. In this case, when the user enters the "r" the system will continue to display the most likely choice for "p" and for "pr" (for example, "provide"). However, if the user continues by entering another consistent letter ("o"), the system would move to the next most likely word (e.g., "process") assuming that, if the user were planning to enter "Provide," he/she would have selected it after entering the "p" or after entering the "r." Those set to display multiple selections would work the same way—any choice that has been implicitly rejected twice would not be displayed. This same approach can be generalized by setting the replacement criterion to any programmed value rather than specifically to two, as in the preceding example. Note that in any case, if a user deletes the most recently input character, the system should restore the previous word candidate set. Figure 1 illustrates these processes:

Figure 1
Flow for word completion feature



A Model

Under ideal conditions (lab conditions), users can input approximately 12 corrected words per minute (CWPM) using a virtual keyboard and fewer than 7 CWPM using a handwriting recognizer on a PDA (Commarford & Lewis, 2004). To demonstrate the advantage that this technique could provide for mobile device users, consider a user who wishes to enter the word, “subjective.” A Pocket PC 2002 device for which the word completion feature is set to display a single choice, will provide the selections presented in Table 2 as the user enters letters.

Table 2
Word completion options by starter letters

Letters	**Word**
S	State
Su	Subject
Sub	Subject
Subj	Subject
Subje	Subject
Subjec	Subjects
Subject	Subjected
Subjecti	Subjective

As Table 2 illustrates, “Subjective” is the fourth choice for words beginning with “Sub.” With the current system a user will need to enter “Subjecti” before “Subjective” becomes the selectable option. At 12 CWPM, it will take a user 45 seconds to enter “Subjecti” and then tap the displayed word with a soft keyboard (9 total taps at 5 seconds per tap). At fewer than 7 CWPM, it will take approximately 80 seconds with a handwriting recognizer. If the word completion feature moved to the next selection option as each item was implicitly rejected, it would have moved from “Subject” to “Subjects” at the “b”, to “Subjected” at the “j” and to “Subjective” at the “e.” This system would allow users to enter the word with a total of six taps, saving them 15 seconds with a soft keyboard and approximately 25 seconds with a letter recognizer. Extrapolating this example to a full text passage, we would see that the feature could save users a significant amount of time.

References

Commarford, P. M., and Lewis, J. R. (2004). Models of Throughput Rates for Dictation and Voice Spelling for Handheld Devices. *International Journal of Speech Technology, 7(1),* 69-79.