Fast Text Prediction in Wtx: the FTL rule

Gianluca Quercini
DISI - Dipartimento di Informatica Università di Genova
Via Dodecaneso 35, 16146 Genova Italy
Email: quercini@disi.unige.it
April 17, 2007

Abstract

Text entry strongly limits the interaction with mobile devices such as cellular phones and PDAs. As a result, the sheer potential carried by them risks remaining unexploited. In principle, handwriting recognition is the more suitable solution, as the interaction with mobile devices typically takes place with a tiny pen. However, several users are still reluctant to use this technology, as they perceive it inaccurate and uncomfortable. In this scenario, text prediction techniques play an important role, witnessed by the success gained by T9 on cellular phones. Wtx, the text entry tool for PDAs developed at Disi, exploits text prediction like T9 does, but with some differences. In particular, text prediction is improved by FTL rule, that requires users to comply with a new way of writing words.

1 Introduction

In the last few years mobile computing has gained a great success, witnessed by the huge amount of projects in this field [4, 2]. This is mainly due to technological improvements that turned portable devices into small Personal Computers. However, their size and the lack of an effective text entry tool heavily limit their use. In desktop PCs and notebooks the Qwerty keyboard provides a comfortable and fast tool for text entry, accepted by every user; the same can not be claimed for mobile devices. Throughout this paper I will mainly refer to pen-based mobile devices (PDAs and smartphones), for which keyboard and mouse are replaced by a tiny pen and a touch-screen. The pen is a more effective tool than the mouse for selecting objects on the screen, such as files and directories, as it guarantees a direct interaction with the objects. On the contrary, text entry is not as comfortable as on Desktop PCs. In particular, there is not a universally accepted tool for text entry; some users prefer virtual keyboards, while others may choose handwriting recognition or other tools such as, for instance, Dasher [10].

In this context, text prediction techniques play an important role, as they allow users to enter words without inserting all their characters. T9 is an example of text prediction tool; however, it is suitable in disambiguating words when writing short SMS or MMS. Tengo\footnote{http://www.tengo.net/} is an attempt of adapting T9 to PDAs; however, T9 is inefficient, especially when the words to write are not in the dictionary [8]. Wtx, a text entry tool for PDAs running a Microsoft PocketPC operating system, is based on text prediction, like T9, but works in a different way [3]. First of all, there is no need of disambiguating...
the words, as there is not a phone-like keyboard, where each key is associated to several characters (typically 3 or 4). Moreover, words matching a character sequence are shown into an area of the screen, so as users can directly select them, without the presence of a devoted key “next”\(^2\). In order to improve text prediction, I devised the so-called rule FTL (First-Third-Last), requiring users to write a word by inserting the first, the third and the last character of it [9]. In the next sections, I will discuss advantages and drawbacks of this new rule, after explaining the basic principles of Wtx.

2 Wtx

Wtx was developed at Disi with the aim of providing PDA users with a powerful tool for text entry. Unlike other well-known text entry tools, it was not thought to be fast, but to be comfortable, meaning that users should not feel tired or stressed when writing a long text. For this reason, Wtx uses a dictionary to suggest a word before all the characters have been inserted. The current version of Wtx runs in two different modes: the traditional and the FTL mode. Section 2.2 will provide more details about both running modes. Moreover, users can choose whether to use a virtual keyboard or a handwriting recognition system. Figure 1 shows the available options. The box layout allows to optimize the interface depending on whether the user is left-handed or right-handed (figure 2(d)); in the box Text Entry Method there are options to change the running mode of Wtx; and finally in the box Handwriting Engine users can choose between a virtual keyboard or a handwriting system.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{WtxOptions.png}
\caption{Wtx Options}
\end{figure}

The first version of Wtx, called WordTree [1], showed considerable differences in the interface. Words in the dictionary were organized into a forest, each tree being rooted at a letter of the

\(^{2}\)“next” is the key on the cellular keyboard used to list all the words suggested by T9, after some characters have been already inserted.
alphabet. Nodes were labelled with word prefixes and leaves with words. To insert a word, users simply had to browse the tree. The key idea underlying the choice of avoiding a virtual keyboard was that selection tasks could be comfortably performed with a pen. However, this approach proved to be unsuccessful because of the size of the tree. In order to visualize all the nodes at a given level, in fact, a scrollbar was required, with a consequent increase of the number of operations performed with the pen. The tests proved that WordTree is slower and more stressful than the Qwerty virtual keyboard. For that reason we changed the interface as explained in section 2.1.

2.1 Wtx: the interface

In order to recall the experience of WordTree (WT), we chose to identify the subsequent releases of the tool by the acronym “Wt” plus a letter of the alphabet representing the current version. For example, Wta was the first release, Wtb, the second one and so on. Since the current interface has remained unchanged for long time, we decided to adopt its name (Wtx) as the final name of the tool.

The area of the SIP is split into two regions: the composition area and the selection area (figure 2).

(a)

(b)

(c)

(d)

Figure 2: Interface of Wtx

The composition area is bound to host a text entry tool to insert the letters of the words. Users can choose among a virtual keyboard (figure 2(a), relative to the traditional mode, and 2(b), relative to the FTL mode) or a handwriting recognition system (figure 2(c)). Figure 2(d) shows an optimized interface for left-handed people. The selection area is moved to the right part of the SIP, as left-handed people would cover it with their left hand. The meaning of the keys in the last row of the virtual keyboard is the following:

- key “#”: replaces the current keyboard with one containing symbols and punctuation marks;
- key “kb”: replaces the current keyboard with one containing numbers and arithmetic symbols;
- key “nl”: inserts a newline;

3SIP (Soft Input Panel) is the area on the screen devoted to text entry on PocketPC PDAs. Its size, that can not be modified because constrained by the operating system, is 320x80 pixels.
• “sh”: shift character;
• “sp”: space character;
• “dl”: del character.

In the keyboard relative to the traditional mode, key “pl” adds an “s” at the end of the last inserted word. This turns out to be useful to form the plural of English regular nouns. This key, in FTL mode, is replaced by “tr”, that allows a temporary switch to the traditional mode. This could be useful whenever a password or a url or any word not in the dictionary has to be inserted. As far as the handwriting recognition system is concerned, a box is shown in the composition area, where users can write the letters. In the current release Wtx is interfaced to the handwriting recognition engine provided by the operating system [5]. However, this engine is not very accurate and some interfacing issues arose during the implementation. For that reason we are planning to embed into Wtx a more accurate and effective recognition system (for example that described in [7]).

![Figure 3: Wtx at work in the traditional mode](image)

The selection area shows up to 10 words obtained from the dictionary. At the start-up, when no letter has been inserted, the ten most frequent English words are shown. Whenever a letter is inserted, the selection area is updated with new words loaded from the dictionary. The dictionary used in the current release contains 13715 words; words beginning with the same letter are sorted by decreasing frequency. This way, more frequent words appear in the selection area before infrequent ones. Also in the selection area words are sorted by frequency. However, in the next releases we will choose the alphabetical order to decrease the time required for looking for them.
2.2 Wtx at work

In both running modes, Wtx, at the start-up, shows the ten most frequent words in the selection area, so that they can be inserted with a single tap. Moreover, as a word is selected from the selection area, or a space character or a new line are inserted, these words are shown again. This reduces the mean number of taps\(^4\) or strokes\(^5\) by 10%.

To insert a word in the traditional mode, user has to insert the first letter, say “c”; the ten most frequent words beginning with that letter are showed in the selection area (figure 3(a)). If the desired word does not appear, the second letter, say “a”, has to be inserted. Consequently, the selection area is updated in order to contain the ten most frequent words having “ca” as prefix (figure 3(b)). If the desired word, say “cause”, comes up, the user has to select it with the pen and Wtx sends it to the text editor along with a space character (figure 3(c)).

In the FTL mode, the situation is quite different. At the beginning, the first letter “c” must be inserted; the selection area is updated as before and Wtx sends to the editor the string “c\_”, meaning that the second letter of the word must be omitted (figure 4(a)). Next, the user has to insert the third letter of the word. The situation changes as shown in figure 4(b); finally the last character has to be inserted (figure 4(c)). If the desired word does not appear yet, the second letter must be inserted (figure 4(d)) and then the fourth, the fifth and so on. In the next section, advantages and drawback of the rule FTL are presented.

3 The FTL rule

The aim of every text prediction system is to understand the word being inserted by the user as soon as possible. Unless the word is very frequent, the system is unlikely to guess it, and thus to suggest it to the user, after just one character has been inserted. For example, in the Wtx dictionary 800 are the words beginning with “a”; it is impossible to visualize all the 800 words in the selection area. Even if that was possible, it would not be effective, as users would spend most of their time in looking for the desired word in the list. The minimization of the words loaded from the dictionary turns out to be a key approach. This can be obtained either by a text prediction system without requiring a collaboration from users, or by a system with the help of users. Most text prediction techniques use the first approach \([6]\), making things easier for users. However, the second approach could be adopted to ease text prediction, provided that the interaction required from users is not complicated. An example of collaboration between users and system is handwriting recognition; since handwriting varies from user to user, most recognition systems associate a list of stylized symbols to each alphabet letter and users have to learn and adopt them. After some training sessions, users get used to the new alphabet and help the system to improve its accuracy during the recognition process. The FTL (First-Third-Last) rule follows the same principle; usually human beings write a word by inserting the first character, then the second, then the third and so on. The FTL rule imposes to write a word by inserting the first, the third and the last character before the other ones. This approach allows a strong selection of the words in the dictionary, as it will be described in the following section. As far as usability is concerned, I performed some preliminary tests, collecting favourable users’ impressions; however, more detailed experiments will be set up and deeply analysed in future.

\(^4\)The action of pressing a key or selecting a word by the pen is referred to as tap

\(^5\)A stroke is the gesture performed with the pen in order to write a letter to be recognized
Figure 4: Wtx at work in the FTL mode
3.1 Advantages of the FTL rule with sequences of three characters

In order to evaluate the selection power of the new rule, I wrote a Java program that generates all the possible sequences of three characters ("aaa", "aab", ... , "zzz") and computes how many words in the Wtx dictionary get selected by applying both the new rule and the traditional rule. The obtained results are shown in figure 5; on the x-axis you find all the possible sequences that match at least one word (henceforth I will refer to them as *useful sequences*), whereas on the y-axis there is the number of the selected words. With the new rule (figure 5(a)) the number of useful sequences is greater than with the traditional rule (figure 5(b)).

![Figure 5: Selection power of both the traditional and FTL modes](a) ![Figure 5: Selection power of both the traditional and FTL modes](b)

As a result, the words are more equally distributed over all the sequences of characters, meaning that the number of words selected by a given sequence is reasonably low. As a matter of fact, the worst case with the new rule is relative to the sequence “dse”, that selects only 64 words, that is the 0.46% of the whole dictionary. With the traditional rule, on the contrary, the worst case leads to 202 words. Moreover, the values reported for the other sequences seem to be lower in figure 5(a) than in figure 5(b). This belief is confirmed by the distribution of the number of the words. Figure 6 shows that the 94% of the sequences of three characters selects up to 10 words, whereas this value falls down to 81% in the traditional case. The advantages of the new rule in Wtx are evident. If the 94% of the sequences selects no more than 10 words, it means that almost every word in the dictionary can be inserted with at most 4 taps, whatever is the length of the word.

4 The FTL rule in Wtx

The FTL rule brings to Wtx several advantages. First of all, it considerably decreases the negative effects of two intrinsic issues: hidden words and long words with the same prefix. A word in the dictionary is said to be hidden when it never comes up in the selection area. This is due to the fact that it has a low frequency and there are ten more frequent words that use it as prefix. In the traditional mode 373 are the hidden words, whereas only 160 are in the FTL mode. As far as long words are concerned, they get displayed in the selection area by their prefix followed by some dots (figure 4(d)). When two words have the same prefix (such as co-operation and co-operate), it turns
out to be impossible to distinguish between them. However, with the new rule the disambiguation of two words with the same prefix can be performed faster, as the last character is inserted as third character, which is not the case in the traditional mode.

A second significant improvement is in the mean tap number. Again I wrote a simple Java program that computes, for each word in the dictionary, the number of taps needed to insert it in both traditional and FTL mode (included the tap required to select the word or the space in the case of hidden word). The mean length, weighted on the frequency, of the words in the dictionary is 4.4 characters. The mean number of taps, always weighted on the frequency of the words, in the traditional mode is 2.43, and 2.32 in the original mode. This means that in both modes Wtx halves the number of taps required to insert a word. In both modes, moreover, the worst case requires 8 taps.

If these results do not seem to be much significant, those listed in table 1 highlight the goodness of the new rule. In the FTL mode the 87.9% of the words can be inserted with a number of tap

<table>
<thead>
<tr>
<th>Tap Number</th>
<th>N. of words in trad. mode (%)</th>
<th>N. of words in FTL mode (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 2</td>
<td>0.07</td>
<td>0.07</td>
</tr>
<tr>
<td>&lt; 3</td>
<td>2.31</td>
<td>1.86</td>
</tr>
<tr>
<td>&lt; 4</td>
<td>14.65</td>
<td>29.49</td>
</tr>
<tr>
<td>&lt; 5</td>
<td>64.50</td>
<td>87.85</td>
</tr>
<tr>
<td>&lt; 6</td>
<td>93.12</td>
<td>95.12</td>
</tr>
<tr>
<td>&lt; 7</td>
<td>98.75</td>
<td>99.62</td>
</tr>
<tr>
<td>&lt; 8</td>
<td>99.96</td>
<td>99.93</td>
</tr>
<tr>
<td>&lt; 9</td>
<td>100.00</td>
<td>100.00</td>
</tr>
</tbody>
</table>

Figure 6: Distribution of the words over all the sequences of 3 characters
in the traditional mode. Table 2 shows that in the 44% of the cases the FTL rule wins over the traditional rule. In most cases the FTL rule gains only a tap over the traditional mode (91.80%). On the contrary, only 7.94% of the words require less taps in the traditional mode than in FTL mode.

<table>
<thead>
<tr>
<th>No. of words (%)</th>
<th>1 tap (%)</th>
<th>2 taps (%)</th>
<th>3 taps (%)</th>
<th>4 taps (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FTL Victories</td>
<td>44.72</td>
<td>91.80</td>
<td>6.73</td>
<td>1.44</td>
</tr>
<tr>
<td>FTL Draws</td>
<td>47.34</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>FTL Defeats</td>
<td>7.94</td>
<td>93.94</td>
<td>5.69</td>
<td>0.37</td>
</tr>
</tbody>
</table>

5 Conclusions and future work

In this paper I presented a short description of a powerful and comfortable text entry tool for PDAs, named Wtx, and of a new rule that makes text prediction faster. Intensive testing is being performed in order to evaluate the level of comfort of Wtx. In particular, it is essential to evaluate the usability of the FTL rule, as it considerably increases Wtx performances. Wtx still needs some improvements: in the current dictionary the plural of regular nouns and the simple past of regular verbs are not provided. This means that the insertion of such words still require additional taps that can and must be avoided. Moreover, we used only a English dictionary so far; it would be interesting to evaluate Wtx in multi-lingual contexts. Finally, improvements in the user interface could be taken into account. The size of the box used by the handwriting recognition system could be further reduced in order to allow 15 words to fit into the selection area; FTL rule will greatly benefit from that, as more than 90% of the words could be inserted with ≤ 4 taps.

References


