Identifying and Correcting Typing Errors for People with Motor Impairments

**Correction Algorithm**
TrueKeys uses a weighted minimum string distance metric (wMSD) to classify errors. This metric counts the number of (1) correct key presses, (2) substitution errors, (3) insertion errors, and (4) deletion errors, as well as transpositions, between the entered word and a correction candidate. Errors with nearby keys are weighted as more likely than errors with distant keys. The wMSD is combined with the word frequency and the frequency of the bigram to produce the final score. The candidate with the lowest score replaces the user’s input.

**Results: TrueKeys Reduces Typing Errors**

<table>
<thead>
<tr>
<th></th>
<th>Motor-impaired</th>
<th>Non-impaired</th>
</tr>
</thead>
<tbody>
<tr>
<td>Uncorrected Error (%)</td>
<td>4.1</td>
<td>1.8</td>
</tr>
<tr>
<td>TrueKeys Error (%)</td>
<td>2.09</td>
<td>3.44</td>
</tr>
</tbody>
</table>

TrueKeys reduces uncorrected typing errors for motor-impaired (2.09% vs. 3.44%) and non-impaired users (1.03% vs. 1.83%) (p<.05).

However, TrueKeys reduced speed for motor-impaired (26.20 vs. 30.25 wpm) and non-impaired users (67.57 vs. 73.85 wpm) when using TrueKeys interactively (p<.05).

**User Study**
We evaluated TrueKeys with 9 motor-impaired and 9 non-impaired users. Motor-impaired participants had a range of health conditions including arthritis, cerebral palsy, Parkinson's disease, and peripheral neuropathy. Participants transcribed 20 phrases with TrueKeys correction enabled and 20 with correction disabled.

**Sample phrases**
my watch fell in the water prevailing wind from the east

**User Interface**
When a user mistypes a word, TrueKeys automatically replaces it with the corrected word. TrueKeys underlines the word to show that it has been changed. If the system guesses incorrectly, the user may choose from an N-best list of correction candidates using the arrow keys.

**Future Work**
We plan to conduct a longer-term study to better understand how users adapt to TrueKeys over time. We may also evaluate how variations of the TrueKeys user interface may be better for users with varying levels of typing ability.

We also plan to extend the current correction algorithm to produce user-specific error models. This would allow TrueKeys to learn users’ common error patterns over time and potentially improve correction accuracy.

Shaun K. Kane  
skane@u.washington.edu

Jacob O. Wobbrock  
wobbrock@u.washington.edu