# Automatically Correcting Typing Errors for People with Motor Impairments

*Shaun K. Kane and Jacob O. Wobbrock*
The Information School
University of Washington
Box 352840
Seattle, WA 98195 USA
{skane, wobbrock}@u.washington.edu

**ABSTRACT**
People with motor impairments often have difficulty entering text accurately when typing on a keyboard. These users also may have trouble correcting errors. We introduce *TrueKeys*, a system that automatically corrects typing errors as they occur. TrueKeys utilizes a word frequency list and a model of the user's keyboard layout to identify typing errors and choose appropriate corrections. We evaluated TrueKeys with 9 motor-impaired and 9 able-bodied users who completed phrase-typing trials with correction enabled and disabled. Results show that using TrueKeys significantly reduced the number of uncorrected errors for both motor-impaired (2.09% *vs*. 3.44% errors) and able-bodied users (1.03% *vs*. 1.83% errors).

**Categories and Subject Descriptors:** H.5.2 [**User Interfaces**]: Graphical User Interfaces (GUI), Input devices and strategies.

**Additional Keywords and Phrases:** Assistive technology, spelling correction, text entry, typing errors

**INTRODUCTION**
Entering text can be a difficult task for users with health conditions such as peripheral neuropathy or cerebral palsy, which may make it difficult to type quickly and accurately. Some users with motor impairments may choose to enter text using an alternative method such as speech recognition, or may use assistive hardware such as a keyguard. However, many users with disabilities choose to forego any assistive devices and instead use a standard keyboard to enter text. These users may benefit from assistive software that filters, augments, or corrects their typed input.

Previous research in assistive typing software generally has taken two approaches to reduce the burden of typing. Some methods, such as word prediction and completion systems, provide an alternate input method that may reduce the number of keystrokes needed to enter text. These systems may benefit some users, but may also require users to learn a new form of text input that may be less efficient. Word prediction systems, for example, have been found to reduce overall typing speed, rather than increase it [4].


*UIST'07*, October 7-10, 2007, Newport, Rhode Island, USA.
ACM 978-1-59593-679-2/07/0010

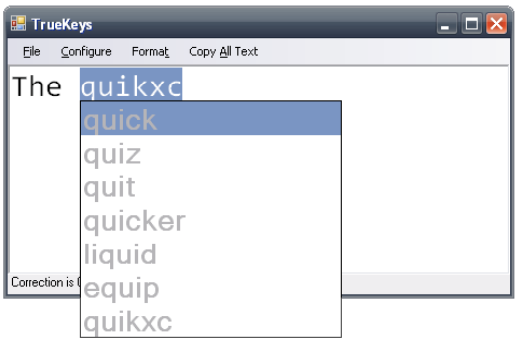


Figure 1. TrueKeys presents the user with a list of word correction options.

Another approach involves filtering the keystrokes entered by the user to remove potential errors. This enables users to type in their normal fashion, while increasing accuracy. Trewin developed two filtering systems for motor-impaired users. The *Dynamic Keyboard* monitors users' typing and automatically adjusts keyboard parameters to reduce errors [7]. *OverlapKeys* detects errors produced when a user strikes two keys at once and removes those errors from the input stream [6]. These systems can operate without user intervention, but may only be able to correct simple errors.

We complement these prior approaches with *TrueKeys*, a system that automatically corrects a variety of common typing errors. TrueKeys combines a word frequency list with a model of keyboard layout to identify and correct typing errors. TrueKeys can be used interactively as a word correction method or passively as a filter on text already entered. In this latter mode, TrueKeys corrects 82.17% of words that are incorrectly entered.

**DESIGN OF TRUEKEYS**
TrueKeys consists of two components: a word correction algorithm and a correction interface. TrueKeys' correction algorithm draws upon prior research that has shown that typing errors often involve keys that are adjacent to the intended key [3]. TrueKeys uses Damerau's version of the Levenshtein minimum string distance (MSD) metric [1] to

calculate the distance between the string entered by the user ($S_{entered}$) and a correction candidate ($S_{candidate}$). Distance is defined as the sum of string edit operations needed to turn one string into the other. The string edit operations are insertion (including an extra character), deletion (omitting a character), substitution (entering an incorrect character in place of a correct one), and transposition (swapping the position of two characters). This score estimates the probability that a user made a specific typing error [2]. We developed a weighted MSD ($MSD_w$) score that favors editing operations that involve adjacent-key errors. The $MSD_w$ score is combined with the word's frequency and bigram frequency, or the probability of the word given the previously entered word, to produce the final distance score:

$$score_{TK} = \alpha * MSD_w(S_{entered}, S_{candidate}) + \beta * f_{word} + \gamma * f_{bigram}$$

$\alpha$, $\beta$ and $\gamma$ are weighting factors derived through pilot testing. In case of multiple possible MSD values, the score is averaged across all possibilities. Correction begins when the user types a space character. The entered string is checked against a list of valid English words. Invalid words are passed on to the correction system. The system calculates the distance score for each candidate in the word list. TrueKeys chooses the candidate with the lowest distance and replaces the original string with the corrected word.

TrueKeys also includes a user interface for interacting with suggested corrections. This interface takes the form of a drop-down list controlled using the keyboard's arrow keys. When the user enters a misspelled word, TrueKeys automatically corrects the word and underlines the correction. If the system suggests an incorrect word, the user may choose from an N-best list of alternatives, including the original word. The user may select the original word to add it to the system word list. Figure 1 shows the correction interface.

## EXPERIMENTAL RESULTS

We performed an experimental evaluation of TrueKeys with 9 motor-impaired and 9 able-bodied participants. Motor-impaired participants had a range of health conditions including arthritis, cerebral palsy and Parkinson's disease. Participants transcribed 20 phrases with TrueKeys enabled and 20 phrases with TrueKeys disabled. Conditions were counterbalanced and no ordering effects were observed. We measured typing speed (words per minute) and uncorrected error rate [5] for each user (Figure 2).

Results show that uncorrected errors were lower with TrueKeys for both motor-impaired (2.09% *vs*. 3.44%) and able-bodied participants (1.03% *vs*. 1.83%). This effect of TrueKeys on errors was significant ($F_{1,16}$=4.82, p<.05). However, there was also a reduction in speed for motor-impaired users (26.20 *vs*. 30.25 wpm) and able-bodied users (67.57 *vs*. 73.85 wpm) when using TrueKeys interactively. This difference was also significant ($F_{1,16}$=17.27, p<.001). Thus, TrueKeys significantly reduced error rate for users but decreased speed somewhat.

TrueKeys provided accurate suggestions in the majority of cases. Of the 174 changes performed by TrueKeys, 109 (62.64%) were correct, and an additional 26 (14.94%) contained the correct word in the N-best list. The remaining 39 (22.41%) corrections did not contain the correct word.

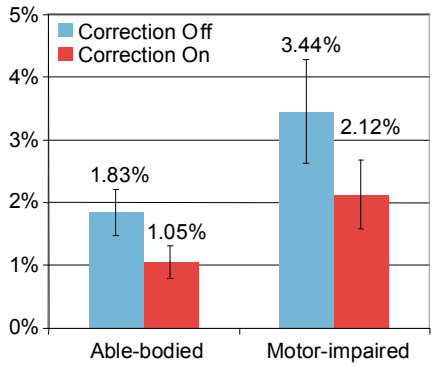


Figure 2. Uncorrected error rate for TrueKeys on and off. Lower is better. Error bars show ±1 SE.

## CONCLUSION AND FUTURE WORK

We have shown that TrueKeys is capable of reducing typing errors produced by users with and without motor impairments. However, when used interactively, TrueKeys also reduced typing speed. This trade-off may be acceptable to those users who have to expend considerable effort to correct errors. The effect on speed may also decrease as users become familiar with the system.

We plan to initiate a long-term user study of TrueKeys to observe how users adapt to the system. We also plan to supplement the TrueKeys correction model with individual users' performance data, and to evaluate TrueKeys as a text entry method for small mobile device keyboards.

## REFERENCES


1. Damerau, F.J. (1964) A technique for computer detection and correction of spelling errors. *Communications of the ACM 7* (3), 171-176.
2. Deorowicz, S. and Ciura, M.G. (2005). Correcting spelling errors by modeling their causes. *Int'l J. Applied Mathematics & Computer Science 15* (2), 275-285.
3. Grudin, J.T. (1984). Error patterns in skilled and novice transcription typing. In *Cognitive Aspects of Skilled Typewriting*, W.E. Cooper (ed). New York: Springer-Verlag, 121-143.
4. Koester, H.H. and Levine, S.P. (1996). Effect of a word prediction feature on user performance. *Augmentative and Alternative Communication 12* (3), 155-168.
5. Soukoreff, R.W. and MacKenzie, I.S. (2003). Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. *Proc. CHI '03*. New York: ACM Press, 113-120.
6. Trewin, S. (2002). An invisible keyguard. *Proc. ASSETS '02*. New York: ACM Press, 143-149.
7. Trewin, S. (2004). Automating accessibility: the dynamic keyboard. *Proc. ASSETS '04*. New York: ACM Press, 71-78.