WebTEM: A Web Application to Record Text Entry Metrics

Ahmed Sabbir Arif
Ryerson University
Toronto, Ontario, Canada
a.s.arif@gmail.com

Ali Mazalek
Ryerson University
Toronto, Ontario, Canada
mazalek@ryerson.ca

ISS '16, November 06-09, 2016, Niagara Falls, ON, Canada
ACM 978-1-4503-4248-3/16/11.
http://dx.doi.org/10.1145/2992154.2996791

Abstract
WebTEM is a Web application to record text entry metrics. It is developed with common Web technologies, thus works on any device with a modern Web browser, and with any keyboard. We evaluated its effectiveness in an empirical study that compared the default Apple iOS, Google Android, and Microsoft Windows Phone keyboards. Results of the study highlighted mobile users' increased dependency on autocorrections.

Author Keywords
Text entry; performance; metrics; Web app; prediction; autocorrection; predictive text; keyboard; evaluation.

ACM Classification Keywords
H.5.m. Information interfaces and presentation (e.g., HCI): Evaluation/methodology.

Introduction
Text entry techniques are usually evaluated by measuring entry speed and accuracy in transcription tasks. Participants are presented with phrases of text from a corpus that they have to transcribe as quickly and accurately as possible [2]. Yet due to the unavailability of tools to record performance metrics with different devices, operating systems, and text entry techniques, researchers are often forced to develop custom evaluation tools. Since these tools are

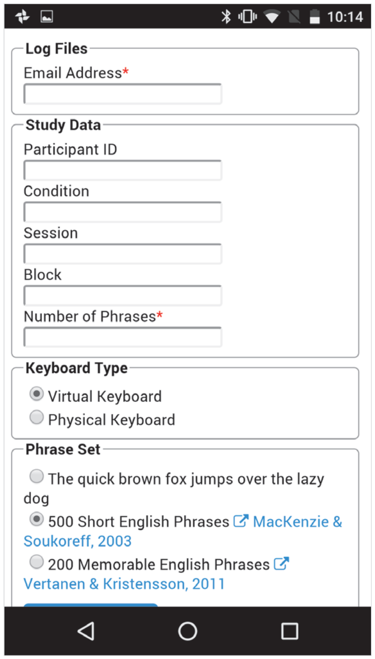

Figure 1. Partial view of WebTEM settings.

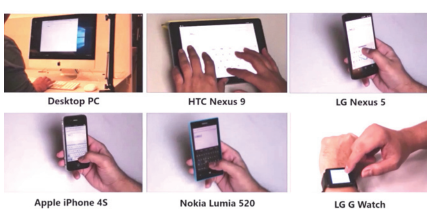

Figure 2. WebTEM on different devices.

tailored to fit the needs of a particular evaluation, usually they cannot be used in other studies. Further, developing these tools can be a challenge, since it requires, as a minimum, programming skills and an understanding of the existing corpora, metrics, and mechanism of the techniques under investigation. This can restrict researchers from quickly evaluating novel text entry techniques. Besides, when developing these tools, they have to decide which metrics to record. If calculating additional metrics becomes necessary after conducting the study, but the existing data sets do not provide the means, they have to first modify the tool and then conduct a new study. As a result, evaluation of text entry techniques is laborious, and depends largely on a researcher's programming skills and knowledge of the existing text entry research. To address this, we present WebTEM, a Web application to record text entry metrics. It is developed with common Web technologies, thus works with any device with a modern Web browser, regardless of the operating system or text entry technique. It allows researchers to select from a large set of metrics and settings. Its goal is to make text entry research easier by reducing the efforts necessary to setup a user study.

Related Work

Some applications are available to collect text entry metrics. However, they all are either specific to an operating system or text entry technique. Wobbrock and Myers [15] developed TextTest, an application that supports most keyboards, but works only on Windows PC. Castellucci and MacKenzie [4] developed TEMA that works with most text entry techniques, but runs only on Google Android. Besides, in an informal test, the current version (v2.1) failed to record metrics on devices running on Android OS 6. Arif et al. [3] developed an application that works only with QWERTY on Apple iOS.

WebTEM

WebTEM is developed with HTML5, CSS3, JavaScript, and PHP. It uses jQuery [8], Diff-match-patch [5], and Grapheme-Splitter [7] JavaScript libraries to perform Ajax requests, analyze plain text, and break strings into their individual user-perceived characters, respectively. WebTEM processes all user interactions on the client side, but periodically pushes all data to the server using PHP for faster performance and to reduce data loss. It also uses PHPMailer PHP library [11] for its integrated SMTP implementation. WebTEM allows researchers to select from a large set of metrics and settings, discussed below. It is freely accessible from http://www.asarif.com/resources/WebTEM.

Study-specific data. Researchers can enter optional session-specific data for recordkeeping, such as participant ID, condition, session, and block (Figure 1). But they must enter the total number of phrases in the session for the application to display the phrases in a randomized order. It also requires a valid email address to send all log files to researchers upon completion of the session. The device must also have access to the Internet throughout the study to avoid any data loss.

Keyboard type. Researchers must select whether they are evaluating a physical or a virtual keyboard. The application detects and records all predictive actions, such as word prediction, autocorrection, and auto-capitalization, for virtual keyboards. However, if detection of these features is desirable for a physical keyboard, researchers could select "Virtual Keyboard", regardless of the actual keyboard type (Figure 1).

Confidentiality. To maintain participant confidentiality, the application does not store any data on the server.

It generates anonymous temporary files at runtime that are automatically emailed to the researcher, and then deleted. The application also deletes all temporary files when the researcher leaves the application, such as exits, refreshes, and/or reloads the application.

Limitations. WebTEM detects all input based on the events in the input area. As a result, it cannot detect modifier and special keys, such as Shift, Ctrl, Alt, etc., for virtual keyboards that do not emulate actual key events.

Further, it detects predictions based on patterns and entry time. While in our tests it was able to detect all predictions, it may misrecognize some in some scenarios. Hence, the predictive metrics reported by the application should be treated as estimations.

WebTEM detects predictive features based on input patterns and event time. For this, we recorded different prediction patterns and event times from informal user studies with multiple devices and keyboards. Results revealed that regular taps on virtual keyboards always take more than 90 milliseconds, while automated events always take less. We used this as a threshold, together with input patterns, to detect predictive behaviors.

Phrase sets. WebTEM includes the three most popular phrase sets, two for adults [13, 14] and one for children [9]. It also includes the "The quick brown fox jumps over the lazy dog" pangram that is frequently used in quick evaluations, since it contains all letters of the English alphabet.

Performance metrics. WebTEM records all commonly used metrics, including Words per Minute (WPM) and Characters per Second (CPS) for entry speed, and Error Rate (ER), Minimum String Distance Error Rate (MSD ER), Keystrokes per Character (KSPC), Corrected Error Rate (CER), and Total Error Rate (TER) for accuracy [2, 13]. Apart from these popular metrics, it can also record the following.

Visual Scan Time (VST) signifies the time (in milliseconds) users took to visually scan a recently completed phrase, mainly to proofread, before submitting it.

Cursor Control Count (CCC) is the total number of times users repositioned the cursor using the arrow keys, the mouse, direct touch, or a digital pen to correct errors or to edit text in a text entry episode.

Backspace Count (BC) is the total number of backspaces in a text entry episode.

Prediction Rate (PR) is calculated as the ratio (%) of the total number of characters automatically entered by the predictive system and the total number of characters entered in a text entry episode. It does not account for incorrect predictions, since they are difficult to identify.

International text entry. WebTEM supports non-Latin text entry evaluations. Currently, it includes a Bengali corpus [1] and several performance metrics for non-Latin scripts, calculated using the output stream convention [12]. In the future, we will include more corpora and metrics to further accommodate the international text entry community.

Additional options. WebTEM includes the following options for researchers to customize a study condition.

Disable predictive features. This disables the browser's spell checker and the keyboard's predictive features, i.e., word prediction, autocorrection, and auto-capitalization.

Present all phrases in lowercase. This converts all uppercase letters in the presented text to lowercase.

Ignore letter case and extra spaces in metrics calculation. This ignores letter case mismatch and extra spaces in metrics calculation. This option is useful for predictive techniques that auto-capitalize words and automatically enter a space after each prediction.

Present all phrases without special characters. Some phrase sets include punctuation and other characters to increase the external validity of the work. However, if the techniques under investigation use the same mechanism to enter special characters, it is better to exclude these from the study to eliminate a potential

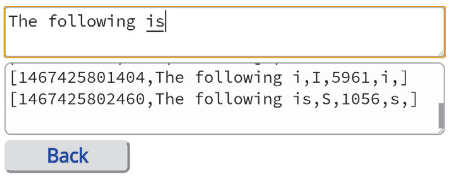

Figure 3. WebTEM works with almost all operating systems, browsers, and keyboards. Yet to verify compatibility, researchers can press the "Check" button and type a chunk of text in the input area to see all events in the aforementioned format.

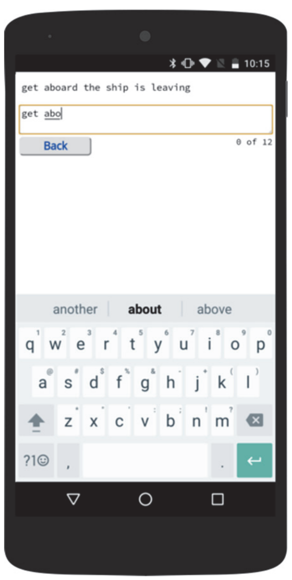

Figure 4. WebTEM is freely accessible from http://www.asarif.com/resources/WebTEM

confound. This option allows researchers to present all phrases without special characters.

Display the number of phrases entered. Participants often query the number of phrases they still have to enter, especially in longer sessions. This interrupts the study and disrupts the natural flow of text entry. To avoid this, this option displays the number of phrases entered below the input area.

Display performance summary. This displays speed and accuracy rates for the last entered phrase, and the session averages, below the input area. We, however, recommend against using this, since it makes participants aware of their performance, compromising their input behavior and reducing the external validity. Yet we included this for better comparisons between data collected using different tools, since some existing tools [4] include this feature.

Hide presented text when users start typing. In text entry studies, participants are usually asked to take the time to read, understand, and memorize the phrases before entering them. Researchers can enforce this by using this option that hides the presented phrase when the user starts typing.

Force error-free submissions. Some user studies may require participants to enter error-free text, especially when an error correction method is being evaluated. This option disables the entry of text containing errors. Selecting this also enables auditory feedback. That is, the system makes a noise when the user attempts to enter phrases with errors.

WebTEM preselects the most popular phrase set, metrics, and settings for convenience. It also stores cookies on the devices to assure that researchers do not have to reselect the options at each visit.

Log files. The application generates one information (.INFO) and two tab-delimited files (.TSV) to record all settings, timestamped events, and performance metrics, respectively. Each row of the metrics log represents a phrase and each column a metric. The last row holds average values. Events are recorded as [time,text,event,duration,insertion,deletion], where time is the number of milliseconds since Jan 1, 1970, text is the current state of the transcribed text, event is a user or system action, e.g., a tap, an autocorrection, etc., duration is the time for the action in milliseconds, and insertion and deletion are character/s entered and deleted by the action, respectively. See Figure 3.

A User Study

We tested WebTEM with multiple devices, e.g., desktop computers, tablets, smartphones, and smartwatches, operating systems, e.g., Android, iOS, Mac, Windows, and Windows Phone, browsers, e.g., Chrome, Firefox, Safari, Internet Explorer, and Microsoft Edge, and keyboards. Figure 2 shows some examples. We also tested its effectiveness in an empirical study.

Apparatus

The study used an Apple iPhone 4S, 115.2×58.6×9.3 mm, 140 g, running on iOS 9.3.2, an LG Nexus 5, 137.9×69.2×8.6 mm, 130 g, running on Android 6.0.1, and a Nokia Lumia 520, 119.9×64×9.9 mm, 124 g, running on Windows Phone 8.1 (Figure 5). The devices accessed WebTEM using their default browsers, namely Safari, Chrome, Internet Explorer, respectively. To our knowledge, no studies have compared these keyboards.

Participants

Twelve participants, aged from 18 to 33 years, average 24.6, took part in the study. Four of them were female

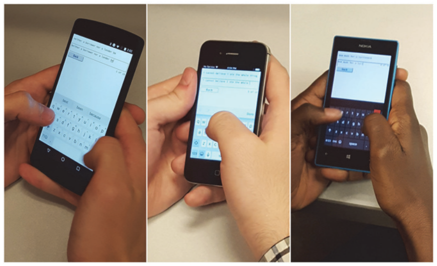

Figure 5. Participants transcribing phrases with WebTEM using the three devices and keyboards.

	Android	Apple	Windows
WPM*	30.31* σ = 10.9	29.87 σ = 11.7	24.98 σ = 8.8
ER*	6.8 σ = 13.9	1.86* σ = 4.6	4.75 σ = 2.9
TER*	10.29 σ = 9.5	9.72* σ = 11.5	13.9 σ = 9.9
PR*	37.98* σ = 30.2	6.62 σ = 8.6	29.32* σ = 37.1

Table 1. Average Words per Minute (WPM), Error Rate (ER), Total Error Rate (TER), and Prediction Rate (PR) for the three keyboards. The symbol "σ" signifies standard deviation, while "*" marks statistically significance.

and all were right-handed. They were all experienced touchscreen users. Most of them (84%) owned at least one Google Android device. The others owned either/both an Apple iOS (17%) or a Windows Phone (8%) device.

Procedure and Deign

In the study, participants were asked to transcribe ten short English phrases from the MacKenzie and Soukoreff [10] set on each device using the respective keyboard. The application displayed all phrases in lowercase and in random order. Participants were asked to hold the device and enter the phrases as they usually would on their own devices. The keyboards enabled word prediction and autocorrection. However, personalized prediction was disabled to eliminate a potential confounding factor. Participants could also gesture type on the default Android and Windows Phone keyboard, but the default iOS keyboard does not support this feature. There was no auditory feedback. Error correction was encouraged but not forced. Participants were asked to enter the phrases as fast and accurately as possible. There was no practice block, but the keyboards were demonstrated before the study. The study used a within-subjects design for the three factors: the three devices and their respective virtual keyboards. The factors were counterbalanced. In summary, the design was, 12 participants × 3 conditions × 10 phrases = 360 phrases, in total.

Results

We used a repeated measures ANOVA for all analysis. Table 1 displays the results.

Entry Speed (WPM)

An ANOVA identified a significant effect of keyboard on WPM ($F_{2,11}$ = 4.56, p < .05). A Tukey-Kramer Test revealed that Android was significantly faster than Windows.

Error Rate (ER)

An ANOVA identified a significant effect of keyboard on ER ($F_{2,11}$ = 9.71, p < .001). A Tukey-Kramer Test revealed that Apple was significantly more accurate than Android and Windows.

Total Error Rate (TER)

An ANOVA identified a significant effect of keyboard on TER ($F_{2,11}$ = 3.84, p < .05). A Tukey-Kramer Test revealed that Apple was significantly faster than Windows.

Prediction Rate (PR)

An ANOVA identified a significant effect of keyboard on PR ($F_{2,11}$ = 7.88, p < .005). A Tukey-Kramer Test revealed that PR was significantly lower for Apple than Android and Windows.

Subjective Analysis

Most participants (84%) selected Android as one of their most preferred keyboards, followed by Apple (17%) and Windows (8%). Further enquiry revealed that they preferred Android mostly due to familiarity (84% of them were Android users) and its predictive system – 60% of them felt that Android's predictive system was more reliable. Interestingly, only about 25% picked a keyboard they are unfamiliar with as one of their most preferred, again primarily for its superior predictive system (67%), secondarily for its design (33%).

Windows was the least preferred keyboard (67%), followed by Apple (25%) and Android (8%). When enquired about the reasons, most participants (75%) blamed the keyboard's unreliable predictive system. Some also complained about the device itself, such as it is smaller (25%) and its screen is not as responsive as the other devices (13%).

Additional Observations

We asked participants if they gesture type on their devices. About 67% responded that they never gesture type, while 25% responded that they sometimes do. The remaining 8% use gesture typing almost exclusively. We also observed the posture used to input text. About 84% of participants used the two-thumb, while 16% used one thumb (8% of them were gesture typists). They all held their devices in portrait position.

Support

Researchers could report bugs or request new features, performance metrics, and/or corpora for their user studies via email to textentry@asarif.com.

Acknowledgments

We thank the members of RE/Lab at Ryerson University, particularly Jason Nolan, for helping us with the user study. This work has been supported in part by the Canada Research Chairs (CRC) program.

Discussion

Results revealed that there was a significant effect of keyboard on WPM, ER, TER, and PR. Android was significantly faster, while Apple was significantly more accurate. Interestingly, PR was substantially lower for Apple, suggesting that users rarely used the keyboard's predictive features. The fact that it did not include a prediction bar like the other keyboards may have caused this. It is possible that some users did not realize that it also provides predictive features like autocorrect, thus were more careful while typing, increasing its accuracy rate. This shows how dependent users are nowadays on predictive features. Post-study survey also highlights this, where participants picked the keyboards they felt has the most reliable predictive systems as their most preferred. Yet, we suggest caution when interpreting the results considering our Android-heavy user base.

Conclusion and Future Work

We presented WebTEM, a cross-platform customizable Web application to record text entry metrics that works with almost all devices and keyboards. We demonstrated its effectiveness in a study that compared the default Apple iOS, Google Android, and Microsoft Windows Phone keyboards. Results reveled that the default Android keyboard is significantly faster and Apple is significantly more accurate. Results also highlighted mobile users' increased dependency on autocorrections. In the future, we will include more corpora and metrics for ultra-small devices, e.g., smartwatches, and non-Latin text entry. We will also include additional settings.

References

[1] Arif, A. S. and Fardeen, S. A phrase set for Bengali text entry evaluations based on actual text messages. *Ext. Abstracts CHI '16*, ACM (2016), 2992-2998.

[2] Arif, A. S. and Stuerzlinger, W. Analysis of text entry performance metrics. TIC-STH '09. IEEE (2009), 100-105.

[3] Arif, A. S., Lopez, M. H., and Stuerzlinger, W. Two new mobile touchscreen text entry techniques. *Poster GI '10*, CEUR-WS.org/Vol-588, 22-23.

[4] Castellucci, S. J. and MacKenzie, I. S. Gathering text entry metrics on Android devices. *Ext. Abstracts CHI '11*, ACM (2011), 1507-1512.

[5] Diff-match-patch. https://code.google.com/p/google-diff-match-patch

[6] Font Awesome. http://fontawesome.io

[7] Grapheme-Splitter. https://github.com/orling/grapheme-splitter

[8] jQuery. https://jquery.com

[9] Kano, A., Read, J. C., and Dix, A. Children's phrase set for text input method evaluations. *NordiCHI '06*, ACM (2006), 449-452.

[10] MacKenzie, I. S. and Soukoreff, R. W. Phrase sets for evaluating text entry techniques. *Ext. Abstracts CHI '03*, ACM (2003), 754-755.

[11] PHPMailer. https://github.com/PHPMailer/PHPMailer

[12] Sarcar, S., Arif, A. S., and Mazalek, A. Metrics for Bengali text entry research. *CHI 2015 Workshop on Text Entry on the Edge* (2015). Seoul, South Korea.

[13] Soukoreff, R. W. and MacKenzie, I. S. Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric. CHI '03, ACM (2003), 113-120.

[14] Vertanen, K. and Kristensson, P. O. A versatile dataset for text entry evaluations based on genuine mobile emails. *MobileHCI '11*, ACM (2011), 295-298.

[15] Wobbrock, J. O. and Myers, B. A. Analyzing the input stream for character-level errors in unconstrained text entry evaluations. *ACM Trans. Comput.-Hum. Interact. 13*, 4 (2006), 458-489.