Punctuation on Touchscreen Keyboards: Analyzing Use and Evaluating Input Techniques

Sana Malik
Department of Computer Science
University of Maryland
College Park, MD 20740
maliks@cs.umd.edu

Leah Findlater
College of Information Studies
Human-Computer Interaction Lab
University of Maryland
College Park, MD 20740
leahkf@umd.edu

Abstract
Non-alphanumeric symbols are rarely considered in text input research even though some punctuation is more frequent than the least common English letters. In this paper, we first evaluate punctuation frequency in two contrasting sources (Twitter and Google N-Grams). We then present a controlled study to compare existing techniques for punctuation input on touchscreens: (1) the *status quo*, where punctuation symbols are stored on an alternate keyboard layer, and (2) an approach where users *draw* punctuation symbols atop the QWERTY keyboard itself [2]. Our findings show patterns in punctuation use (e.g., comparing mobile and desktop input), and highlight the cost of mode-switching to enter punctuation marks.

Author Keywords
touchscreen, text input, gestures

ACM Classification Keywords
H.5.2 [Information interfaces and presentation]: User interfaces - *input devices & strategies*.

Introduction
People are increasingly using touchscreen devices for text entry. As a result, a substantial amount of research has focused on improving touchscreen text input, such as incorporating language models to improve typing accuracy

Symbol	Twitter Mobile	Twitter Desktop	Google N-gram
.	1.694	1.748	1.151
@	1.221	1.258	0.000
!	0.940	0.813	0.013
'	0.550	0.446	0.200
_	0.527	0.499	0.001
,	0.401	0.532	0.000
:	0.381	0.344	0.087
#	0.377	0.350	0.000
?	0.338	0.362	0.032
"	0.205	0.110	2.284
-	0.185	0.193	0.217
)	0.181	0.228	0.140
<	0.095	0.100	0.001
>	0.094	0.106	0.002
(	0.089	0.087	0.140
*	0.075	0.072	0.008
&	0.055	0.044	0.005
;	0.048	0.051	0.096
/	0.042	0.046	0.019
^	0.017	0.023	0.003
=	0.016	0.025	0.002
~	0.013	0.020	0.001
$	0.010	0.012	0.005
|	0.007	0.007	0.001
\	0.005	0.003	0.001
+	0.005	0.006	0.001
%	0.004	0.004	0.006
]	0.002	0.005	0.010
{	0.002	0.001	<.001
}	0.002	0.001	<.001
[	0.002	0.003	0.010
`	0.001	0.002	<.001

Table 1: Punctuation frequencies as percentage of total characters (minus spaces); ordered by frequency in Twitter Mobile.

[3], calculating optimal keyboard layouts to reduce movement time [9], and proposing alternatives to standard tapping on QWERTY keyboards (e.g., [5, 6]). These approaches, however, focus primarily on alphabetic input. Non-alphanumeric symbols such as punctuation marks are rarely considered, even though some punctuation marks have been shown to be more frequent than some English letters [9]. Indeed, the arguably most popular phrase set for text input evaluations in recent years—Mackenzie and Soukeroff's set of 500 phrases [8]—contains no punctuation symbols.

In this paper, we explore two aspects of non-alphanumeric symbol input on touchscreens. First, we evaluate the frequency of punctuation symbols in two contrasting corpora: Twitter data and the Google N-gram corpus [7]. Within the Twitter data, we also compare mobile versus desktop usage, in contrast to Zhai et al.'s [9] similar analysis of news articles and chat room transcripts that made no such distinction. Our findings show that many symbols occur more frequently than some letters, especially on mobile devices.

Second, we present a controlled lab experiment with 10 participants that compares two existing techniques for entering punctuation symbols on touchscreens: a gestural approach that we had previously proposed but did not formally evaluate [2] (Figure 1), and a *status quo* moded-keyboard approach where the user switches between alternate keyboard layouts (Figure 2). The gestural approach allows users to draw punctuation symbols atop the keyboard itself without having to switch to an alternate layout. While we previously conducted a preliminary evaluation of the two approaches, this study allows us to evaluate detailed performance aspects (e.g., in switching from letter to punctuation input modes).

Punctuation Symbol Frequency

To understand how non-alphanumeric symbols are used in comparison to letters, we first examined character frequency in two contexts: on Twitter and in the Google N-gram corpus (Version 1) [7]. The two corpora are distinct in that the Google corpus reflects traditional English spelling and grammar, whereas the Twitter corpus is more informal and forces an abbreviated style of text that has a unique form of spelling and grammar. Additionally, because we were particularly interested in how touchscreen keyboards may affect character usage compared to traditional keyboards, we analyzed tweets from mobile Twitter clients separately from tweets from desktop Twitter clients.

The Google corpus contains frequencies of words appearing in English books published between 1538–2008. In total, it contained 472,764,897 unigrams. We generated the Twitter corpus in June 2012 using the Twitter API, which uniformly samples 1% of the public stream. While the Twitter API provides information about the user's language, this data is not always reliable. Instead, we filtered tweets to those most likely to be in English by limiting allowed characters to ASCII 33–126; we also removed URLs from the analysis, since these are often automatically generated or pasted in. Finally, the mobile versus desktop categorization was done using a manually created list of popular mobile Twitter clients. This resulted in two datasets: 57,662 desktop tweets and 173,876 mobile tweets. For each corpus, we counted the overall usage of English characters (letters and symbols, ASCII 33–126), and ranked the characters by their frequency. Spaces made up 15.6% and 15.5% of the mobile and desktop Twitter data, respectively. However, since spaces were not present in the Google corpus (nor were commas), we excluded them from all analyses.

Letter	Twitter Mobile	Twitter Desktop	Google N-gram
e	9.34	9.52	11.58
a	9.15	9.25	7.52
o	7.09	7.36	7.07
t	7.04	6.82	8.57
i	6.52	6.44	7.08
n	6.15	6.02	6.74
s	5.19	5.26	6.15
h	4.60	4.51	4.71
l	4.38	4.35	3.82
r	4.24	4.37	5.86
m	3.18	3.15	2.38
d	3.12	3.14	3.55
u	3.10	3.17	2.55
y	2.74	2.64	1.55
g	2.60	2.41	1.75
c	2.02	2.09	3.13
k	2.00	2.00	0.52
w	1.95	1.86	1.55
b	1.85	1.75	1.40
p	1.64	1.72	2.00
f	1.42	1.48	2.23
v	0.80	0.87	0.99
j	0.57	0.54	0.16
z	0.27	0.28	0.09
x	0.27	0.29	0.22
q	0.09	0.15	0.11

Table 2: Letter frequencies as percentage of total characters (minus spaces); ordered by frequency in Twitter Mobile.

Punctuation and letter frequencies for the three corpora are shown in Tables 1 and 2; numbers (0–9) made up an additional 1.1% in each of the Twitter corpora and 1.7% in the Google corpus. In the Google corpus, non-alphanumeric symbols comprised 4.4% of characters, and 6 punctuation symbols (**. - ' ( ) ;**) appeared more frequently than the least frequent letter (**q**). Punctuation symbols were more frequent in both Twitter corpora, comprising 7.5% of characters in the mobile tweets and 7.6% in desktop. There were, however, differences between the mobile and desktop tweets. First, the mobile tweets were shorter than the desktop tweets on average (49.9 vs. 54.9 characters). As well, the mobile tweets contained more unique symbols that were at least as frequent as **q**: there were 14 such symbols for mobile (**. @ ! ' _ , : # ? " -)** < >) versus 11 for the desktop (**. @ ! , _ ' ? # :)** -).

The increased use of punctuation marks in the informal tweets versus the more formal Google corpus is similar to a distinction that Zhai et al. [9] found between chat room logs (informal) versus online news articles (formal): symbols appeared more frequently in the chat room corpus than in news articles. Additionally, **?** and **!** appeared in the top 20 characters for the chat room corpus, but not for the news articles (as they did for the Twitter versus Google book corpora).

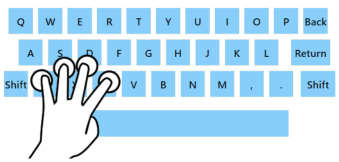
Evaluating Methods for Punctuation Input

In the second phase of this work, we evaluated two existing input techniques for entering punctuation symbols: a QWERTY keyboard augmented with gestural input, where users can draw the punctuation symbol they would like to enter, and a moded keyboard that allows uses to toggle between two keyboard layouts, an alphabetic layout and a layout with punctuation and number keys. In a preliminary study of these two keyboards [2] six participants entered 18 pairs of words with punctuation marks for each interface and no performance differences were found between the moded and gesture keyboards. In contrast, the current study included ten participants and longer typing tasks, allowing us both to confirm the previous findings and to conduct a detailed examination of the cost to switch between letter and punctuation input modes for the keyboards.

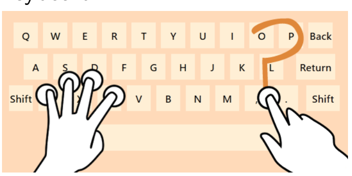
The Moded and Gestural Keyboard Interfaces

The keyboard interfaces were implemented for a Microsoft Surface 2 tabletop. The *gesture keyboard* is an implementation of the 10-finger keyboard described in our previous work [2], updated to work for the Surface 2 (Figure 1). It is a standard QWERTY keyboard that allows users to draw punctuation symbols *atop* the keyboard when "gesture mode" is activated. To draw a symbol, the user enters gesture mode by placing 4+ fingers down with either hand, and draws the symbol with the other hand using a single- or multi-touch gesture. Informed by the character analysis of the Twitter and Google corpora, ten punctuation marks were included for the study: **' - $ # " ? ! @ & %**. The period and comma were not included because they commonly appear on the first layer of touchscreen keyboards (e.g., Apple's iPad).

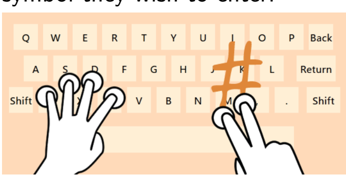
Single-touch gestures are defined for each punctuation mark, while multi-touch options are also provided for **$, "** and **#**. For example, to enter the pound sign (#), a user can draw either four separate strokes, or use two fingers to draw a pair of lines at time (Figure 1c). In addition to the gestures defined in the original study [2], we found that users preferred using a "dot" gesture (as opposed to an open circle) in question marks and percent signs, so we added these options as well.



(a) Users enter gesture mode by placing four fingers on the keyboard.



(b) The keyboard then changes color, and users can draw the symbol they wish to enter.



(c) Some symbols (e.g., pound) also allow multi-touch gestures.

Figure 1: The gesture keyboard supports single- and multi-touch gestures.

The *moded keyboard* is an implementation of the *status quo* solution for punctuation input on touchscreens (Figure 2). It uses the same QWERTY layout as the gesture keyboard, but additionally has two keys labeled "?123" that toggle between the alphabetic keys and an alternate keyboard layer. The alternate layer was modeled after the Android operating system keyboard, with the exception of the last row, where we removed the comma and period keys and added an additional "ABC" toggle on the right side of the keyboard.

We also implemented but did not formally evaluate a *combined keyboard* that allows users to enter punctuation marks with gestures *or* by switching to the alternate layer.

Method
Ten participants, aged 21–30 ($M = 24.8, SD = 2.9$), were recruited through on-campus mailing lists. Two participants were female and two were left-handed (8 were right-handed). All participants had used a touchscreen device at least once before, most commonly a phone or tablet, and four had some experience with a gesture input keyboard (e.g., Swype). Participants were compensated $15 for their time.

The study employed a 2x2 within-subject factorial design, with factors of: *Keyboard* (moded vs. gestural, as described above) and *Task* (phrases vs. randomly generated words). For the phrases task, we adapted the MacKenzie and Soukoreff phrase set [8] by adding exactly two punctuation marks to 100 phrases, one near the middle of the phrase and one at the end (e.g., `physics & chemistry are hard!`). The pairs of punctuation marks were chosen with a uniform distribution from the set supported by the gesture keyboard, except for the single and double quotations, which were always paired with themselves (e.g., `video camera with a good "zoom"`).

The randomly generated words task was not as realistic but allowed us to collect more punctuation input in a short period of time. For that task we created 5-character words from a uniform probability of each character being a letter or symbol (e.g., `%va$k`).

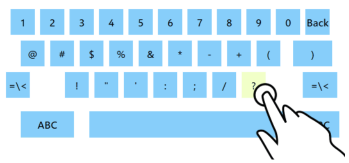
Study sessions lasted 60-90 minutes. For each keyboard, participants completed 20 training trials, which were repetitions of each of the 10 symbols, followed by 40 trials each for the phrases and randomly generated words tasks (i.e., 40 phrases and 40 words). The orders of presentation for *Task* and *Keyboard* were fully counterbalanced. Between each keyboard condition, participants completed Likert scales based on the NASA TLX [4] and answered open-ended questions about their thoughts on the keyboard. Finally, at the end of the session, users briefly used the combined keyboard, typing up to 10 phrases.

In total, participants entered 28,486 characters. To analyze overall typing speed, we ran a two-way repeated measures ANOVA (*Keyboard* x *Task*). For corrected and uncorrected error rates, which violate the normality assumption of parametric tests, we used Wilcoxon signed ranks tests to compare the two keyboards within each task. For each of the switching costs we analyzed (e.g., from letter to punctuation mark), paired two-tailed t-tests were used to compare speed across the two keyboards.

Results
Confirming the preliminary results [2], overall speed and error rates were similar for both the moded and gesture keyboard. Results for each task are shown in Table 3. No statistically significant main or interaction effects of *Keyboard* were found for any of these measures.

To compare the cognitive cost of switching into and out of punctuation mode for the two keyboards, we considered

(a) To switch to the symbol key layout, a user presses the "?123" key.



(b) The symbol key layout is based on Android keyboards.

Figure 2: The moded keyboard uses an alternate layer of keys for punctuation.

four types of input pairs: punctuation to letter, letter to punctuation, punctuation to space, and space to punctuation. We considered letters and spaces separately since the space bar is present on the alternate layout of the moded keyboard. The average speed for switching *into* punctuation mode was similar for both conditions whether from a letter (moded = 1862ms, gesture = 1789ms) or from a space (moded = 1691ms, gesture = 1871ms). However, switching *out* of punctuation mode and back to a letter key was significantly faster for the gesture keyboard than the moded keyboard: 913ms versus 1632ms on average ($t(9) = 6.15$, $p < .001$). In contrast, switching from punctuation mode back to the space key was significantly faster with the moded keyboard than the gesture keyboard: 556ms vs. 819ms ($t(9) = 3.62$, $p = .006$).

For the NASA TLX scales, t-tests revealed no significant differences between the two keyboards on any of the six scales (mental, physical and temporal demand; performance; effort; frustration). Average responses ranged from performance ratings of 6.1 and 6.3 out of 20 for the gesture and moded keyboards, respectively (where 1 is perfect performance), to 12.1 out of 20 for both keyboards in terms of mental demand (where 1 is low demand). Participants seemed to favor the alternate keyboard for punctuation with many curves and the gesture keyboard for punctuation with straight lines; seven participants listed at least one of **@ % $ &** as being problematic on the gesture keyboard, and four participants listed at least one of **' " #** to be more difficult to enter on the alternate keyboard than the gesture keyboard.

Overall, six participants preferred the gesture keyboard, while four preferred the moded keyboard. A participant who preferred the gesture keyboard said, "The gesture keyboard is easier to use as you reduce the amount of click [tapping] to switch; I enjoyed hand written commands" and two others said they felt the gesture keyboard made typing "more enjoyable" and "like a game." Those who preferred the moded keyboard liked that it was similar to what they were used to on other devices. One participant said that they found the moded keyboard less confusing: "Sometimes for gesture, I would try to draw in a letter." Preliminary feedback of the combined keyboard suggests that uses appreciate having both options available. One user noted, "Gestures is great first option, but if [the recognizer] doesn't work the first time, you can switch to alternate layout."

Discussion and Conclusion

Our findings motivate the need for future work on punctuation input for touchscreen keyboards, both by demonstrating the frequency of punctuation use in one extreme of mobile text input (Twitter) compared to traditional English, as well as by highlighting the costs of switching between punctuation mark input and letters/spaces for two existing touchscreen keyboard designs. The **@** and **#** symbols were unsurprisingly common in the Twitter data since they play a specific role in tweets. However, even apart from **@** and **#**, punctuation was more common in the Twitter data than the Google corpus, particularly for mobile devices.

The comparison of two existing methods for entering punctuation symbols reveals tradeoffs in the cost to switch between letters/spaces and punctuation marks. Compared to the *status quo* moded keyboard, switching from punctuation to a letter was faster for the gesture keyboard, likely because the letter keys always remain visible and thus require less visual search. The moded keyboard, however, was more efficient for entering a space

Measure	Gesture		Moded	
	M	*SD*	*M*	*SD*
WPM	23.0	3.4	23.9	4.6
Uncorrected Error (%)	0.4	0.2	0.8	1.5
Corrected Error (%)	10.5	4.2	9.1	4.0

(a) Phrase task.

Measure	Gesture		Moded	
	M	*SD*	*M*	*SD*
WPM	9.7	2.3	10.9	1.9
Uncorrected Error (%)	0.4	0.4	0.5	0.7
Corrected Error (%)	6.6	2.4	2.8	2.7

(b) 5-character word task.

Table 3: Performance results from gesture and moded keyboards in the experiment.

after a punctuation mark; the spacebar is on all layers of that keyboard and users do not have to return to the first layer to input it. The relatively poor performance of switching from punctuation to space for the gesture keyboard (where the space was also always shown) suggests that there is a cognitive cost in transitioning from drawing a gesture to tapping again on the keyboard. This cost was also reflected by the participant who commented that he sometimes tried to draw a letter.

Typing speeds were relatively slow with both keyboards, partly due to the difficulty of typing punctuation compared to regular text, but also due to limitations of the technology we used. Touch tracking with the Microsoft Surface 2's vision system was substantially worse than the Surface 1.0 that we had used in preliminary work [2], to the point where it would often skip keystrokes when the user typed too quickly. As a result, it would be useful to replicate this study with a capacitive touchscreen device. Additionally, the $N gesture recognizer [1] used in our gesture keyboard is only meant for quick prototyping; performance would likely improve with a more sophisticated recognizer.

The gesture keyboard can be extended in various ways. Because it is meant to augment traditional touchscreen keyboard, we also implemented a prototype version that allow users to enter symbols using either method. The keyboard has the same layout and functionality as the moded keyboard, but also allows users to enter gesture mode. Additionally, the keyboard was implemented on a touchscreen tabletop. While this is practical for a ten-finger keyboard, they are currently not as widely used as touchscreen tablets and handheld devices. Future work includes evaluating how this technique could be extended to smaller devices, with fewer than ten fingers.

Acknowledgments
We thank Rajan Zachariah for providing insight and feedback on this research.

References

[1] Anthony, L., and Wobbrock, J. O. A lightweight multistroke recognizer for user interface prototypes. In *Proc. Graphics Interface (GI'10)* (2010), 245–252.

[2] Findlater, L., Lee, B., and Wobbrock, J. O. Beyond qwerty: augmenting touch-screen keyboards with multi-touch gestures for non-alphanumeric input. In *Proc. CHI'12* (2012), 2679–2682.

[3] Goodman, J., Venolia, G., Steury, K., and Parker, C. Language modeling for soft keyboards. In *Proc. IUI'02* (2002), 194–195.

[4] Hart, S., and Staveland, L. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. *Human Mental Workload 1* (1988), 139–183.

[5] Kristensson, P.-O., and Zhai, S. Shark2: a large vocabulary shorthand writing system for pen-based computers. In *Proc. UIST'04* (2004), 43–52.

[6] Li, F. C. Y., Guy, R. T., Yatani, K., and Truong, K. N. The 1line keyboard: a qwerty layout in a single line. In *Proc. UIST'11* (2011), 461–470.

[7] Lin, Y., Michel, J.-B., Lieberman, E. A., Orwant, J., Brockman, W., and Petrov, S. Syntactic annotations for the google books ngram corpus. In *ACL (System Demonstrations)* (2012), 169–174.

[8] MacKenzie, I. S., and Soukeroff, R. W. Phrase sets for evaluating text entry techniques. In *Extended Abstracts of CHI'03* (2003), 754–755.

[9] Zhai, S., Hunter, M., and Smith, B. A. Performance optimization of virtual keyboards. *Human-Computer Interaction 17* (2002), 89–129.