Typing Efficiency and Suggestion Accuracy Influence the Benefits and Adoption of Word Suggestions

Quentin Roy, Sébastien Berlioux, Géry Casiez, Daniel Vogel

Typing Efficiency and Suggestion Accuracy Influence the Benefits and Adoption of Word Suggestions

Quentin Roy
quentin@quentinroy.fr
School of Computer Science, University of Waterloo
Waterloo, Canada

Sébastien Berlioux
sebastien.berlioux@etu.univ-nantes.fr
School of Computer Science, University of Waterloo
Waterloo, Canada

Géry Casiez[*][†]
gery.casiez@univ-lille.fr
Univ. Lille, CNRS, Inria, Centrale Lille, UMR 9189 CRIStAL
Lille, France

Daniel Vogel
dvogel@uwaterloo.ca
School of Computer Science, University of Waterloo
Waterloo, Canada

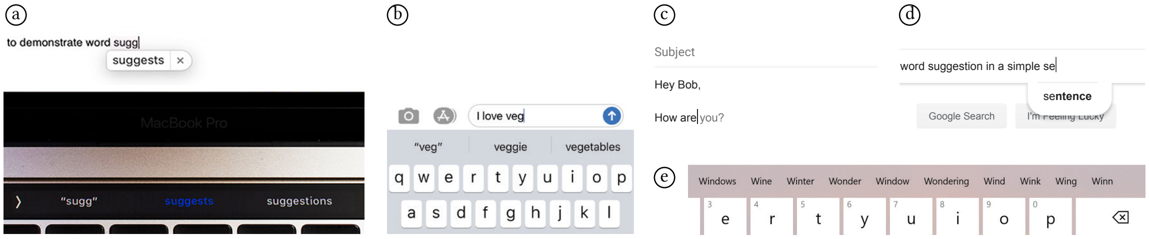

Figure 1: Examples of word suggestions: ⓐ Apple Mail on desktop, suggestions are shown both on the touch bar on top of the keyboard, and underneath the insertion point; ⓑ iOS Messages on phone; ⓒ Google Mail on desktop; ⓓ Google Search on desktop; ⓔ Windows 10's SwiftKey keyboard on desktop. Note that on both the touch bar and iOS messages, there is often only two word suggestions as the left option is used to prevent automatic correction of the word prefix.

ABSTRACT

Suggesting words to complete a given sequence of characters is a common feature of typing interfaces. Yet, previous studies have not found a clear benefit, some even finding it detrimental. We report on the first study to control for two important factors, word suggestion accuracy and typing efficiency. Our accuracy factor is enabled by a new methodology that builds on standard metrics of word suggestions. Typing efficiency is based on device type. Results show word suggestions are used less often in a desktop condition, with little difference between tablet and phone conditions. Very accurate suggestions do not improve entry speed on desktop, but do on tablet and phone. Based on our findings, we discuss implications for the design of automation features in typing systems.

KEYWORDS

text-entry; word prediction

[*] Also with School of Computer Science, University of Waterloo.
[†] Also with Institut Universitaire de France.

CHI '21, May 8–13, 2021, Yokohama, Japan

1 INTRODUCTION

Text input is one of the most common tasks on desktops, laptops[1], tablets, and phones. Physical and soft keyboards remain the main input modality for text input even as speech-to-text is an increasingly viable alternative [33]. However soft keyboards, as implemented on most modern phones and tablets, have much lower entry speeds [27]. As an attempt to improve this, intelligent text entry techniques are integrated into typing interfaces; the two most common are auto correction and word suggestions. This paper focuses on the latter.

A word suggestion interface has two common forms: an inline suggestion to complete a partially typed word or a prediction for the next word; or buttons to enable selection among multiple word suggestions (e.g. Fig. 1). Inline suggestions are more frequently used on desktops. Multiple suggestions presented in a "bar" of buttons just above the keyboard is common on mobile devices, but also exist on desktops, for example using the Touchbar on some Apple models. Regardless of interface, suggestions are typically updated after each keystroke.

Millions of users are daily exposed to word suggestions. Despite their ubiquity, our understanding of their use, the performance gain they offer on different devices, and how accurate they need to be useful, remains incomplete. Understanding how people use these suggestions, and how their use can be improved, has a tremendous

[1] In the rest of this paper, we will often use the term "desktop" to refer both desktop or laptop computers.

impact not only for industry, but for the text entry research community. Previous work has shown that current implementations are infrequently used, with people picking one suggestion every 63 characters [11]. As Palin et al. noted, the efficacy of using word suggestions is unclear since it has a cognitive and perceptual load: users have to switch their attention from the keyboard and the text being typed to attend to the suggestions [27, 31]. They further argue that the usefulness of word suggestions depends on many factors, including the efficiency of the text entry method, accuracy of word suggestions, and user experience. Previous work has mixed results regarding word suggestion benefits [1, 21], some even found them detrimental [27, 31, 37]. However, the effect of suggestion accuracy on usage, performance, and satisfaction remains unclear, nor has the impact of text entry method efficiency on suggestion usage been formally investigated in a controlled experiment.

We investigate the relationship between device typing efficiency and accuracy of word suggestions during text entry in a 36 participant mixed-design experiment. Participants performed a text transcription task. To control for typing efficiency, they used three different devices, desktop, tablet, and phone. We controlled for accuracy between subjects by manipulating how frequently a beneficial suggestion is presented as the participant types.

Our results show suggestion usage does increase with higher accuracy, but the resulting text entry speed does not improve much, and only for the highest accuracy value on mobile devices. Satisfaction, however, was greatly influenced by accuracy. We also found that natural entry speed without suggestions was a better predictor of suggestion use than device: fast typists use fewer suggestions than slow typists, even though fast typists could theoretically save the same number of keystrokes to further increase their speed. These results demonstrate that even highly accurate word suggestions will not compensate for inefficient text entry methods, and are of little benefit for accomplished typists.

2 BACKGROUND AND RELATED WORK

Before describing our experiment, we summarize related work about word suggestions and a related topic of acceptable accuracy.

2.1 Word Suggestion Interfaces

Text prediction systems attempt to predict the next word, or the next few words that a user will type. They may be used for automatic error correction ("autocorrect") [7, 13, 15, 40], but our focus is on their application to word suggestions. Anson et al. differentiate between two types of suggestions: word completions, which are suggestions for a partially typed word, and word predictions, which are suggestions for the next word each time a typist completes a word [1]. Most word suggestion systems today include both forms, but most early systems focused on word completion.

Word suggestions were originally designed for Augmentative and Alternative Communication systems (AAC) to help people with disabilities to communicate, and most of the early literature focuses on users with special needs [12, 15, 37]. However, word suggestions are now ubiquitous on phones [27, 31], and increasingly implemented in applications for desktop computers too.

On desktop computers, suggestions are generally displayed inline with user's input, after the insertion point (like Google Mail), or underneath it (like Apple Pages). Fig. 1 shows examples of word suggestions from commercial systems. If several suggestions are proposed, they may also be shown in a contextual menu, which is common in code editors. Most frequently, they are grouped in a dedicated area, for example a bar displayed at the top of the keyboard of mobile devices, or on the touch bar of recent Apple laptops. Typically, three words are suggested, the most likely one in the middle. However, some systems show more; for example the Microsoft SwiftKey keyboard[2] on Windows 10 can suggest more than ten words (see Fig. 1e).

One important aspect of the design of word suggestions is how frequently they should be updated. Quinn and Zhai investigated this, and found users prefer suggestions to be updated after each keystroke even though it slows them down [31]. We implemented this update strategy because it is how most commercial systems work today.

Another important aspect of their design is how many suggestions should be shown. Swiffin et al. investigated the effect of the number of words suggested [35]. They found keystroke saving started to plateau after 5 suggestions. Later, Venkatagiri found that while 15 word suggestions reduced the number of keystrokes on an Augmentative and Alternative Communication program compared to only 5 suggestions, it had no effect on entry speed [38]. While it remains unclear how many word suggestions is optimal, most modern systems propose 3, with the notable exception of Windows 10's SwiftKey keyboard. In our study, we implemented a 3-word suggestion bar.

2.2 Word Suggestion Benefits

When correct, a word suggestion provides a shortcut to typing the whole word, saving keystrokes. However, word suggestions incur cognitive and perceptual loads that lower these benefits [21, 22, 27, 31]. In fact, analyzing data from 37,370 participants, Palin et al. found using word suggestions resulted in lower text entry speed on average [27]. Similar results had been previously observed for people with physical impairments. For example, in a 1996 study, Koester and Levine found that even for mouth-stick typing, word suggestions actually decreased text entry speed for participants with spinal cord injuries [21]. The authors conclude that the cognitive cost of word suggestions overwhelmed their benefits. One must note that text prediction algorithms improved since the time of their study. A more recent study from Wobbrock and Myers found word suggestions created a significant improvement with their EdgeWrite trackball text entry technique for special need users [42].

A different, but interesting effect that word suggestions do have is that they can change the way users write. For example, Arnold et al. found the use of word suggestions, or more generally text prediction, tends to encourage predictable writing [4].

To summarize, previous work indicates that the use of word suggestions tends to vary across text entry systems, but the effect of device typing efficiency on suggestion use has not been formally investigated in a controlled experiment.

[2] https://swiftkey.com

2.3 Acceptable Accuracy

A reasonable question to consider is, "how accurate must word suggestions be so they are perceived as acceptable and useful?" A perfectly accurate suggestion system would certainly be very acceptable. But this implies even the first word would be suggested before typing anything, and the second word immediately suggested after that, and so on. Obviously such a perfect system requiring no user input beyond selecting every first ranked suggestion is impossible. Consequently, users always face a certain level of inaccuracy, but how much is acceptable before suggestions are ignored completely and no typing efficiency gained? This question is complex, since factors other than accuracy can impact word suggestion use, for example, even a user's emotions [16].

Word suggestion systems rely on a dictionary of possible words, and in their simplest form, present words from this dictionary that are near the word prefix already entered. This basic approach can be improved further by weighting the likelihood of a word as a function of its frequency in the target language, and even further by estimating its likelihood as a function of the previous words using language modelling [19, 26].

Bi et al. proposed a tool to assess the efficiency of word suggestions and correction algorithms by automatically replaying user input [10]. However, while it is useful to evaluate accuracy, it cannot be used to investigate user behaviour when faced with a new typing system, or more or less accurate suggestions.

The effect of word suggestion accuracy has received little attention in the literature. Notably, Trnka et al. [37] investigated the effect of two different word prediction algorithms, one "basic" that relies on frequency of word series, and one "advanced" that employs natural language modelling. They found the advanced algorithm significantly improved entry speed with a soft keyboard. In addition, participants saved 93.6% of the keystrokes that could be saved with the advanced algorithms, but only 78.2% with the other. Said another way, the potential of the advanced algorithm was more utilized than the simpler one. However, Trnka et al.'s experiment only included word predictions, not word completion. Also, they did not formally control accuracy, and did not take into account the effect of the text entry method.

Suggestion algorithms have improved substantially, and more recent works show that better algorithms lead to better performance, making them worth their cost for people with special needs [37]. Still, these results do not appear to generalize for other populations, even if many users report they like word suggestions [31]. Banovic et al. investigated the effect of autocorrect accuracy for spelling mistakes on phones [7]. They found higher accuracy enables users to make more typing mistakes, and increase typing speed. They conclude that improving autocorrect accuracy is worth pursuing. However, Banovic et al. did not investigate word suggestions, or the effect of typing efficiency. It remains unclear how accurate word suggestions need to be to become useful.

From a higher-level perspective, word suggestions are a classic form of automation. Parasuraman and Riley define automation as "the execution by a machine agent (usually a computer) of a function that was previously carried out by a human" [28]. When automating a task, some amount of inaccuracy is practically unavoidable. Low accuracy unavoidably impacts user trust in the automation, and as a result, reduces how often users rely on the automation [24, 29, 44]. Kay et al. proposed a survey instrument to measure the *acceptable accuracy* of a classifier [20]. They notice that the accuracy of classifiers is perceived quite differently depending on the function of their application. For example, the perceptions of a house alarm texting the owner when a possible intrusion is detected is quite different than an alarm calling the police. However, they did not consider the impact of any manual interface control.

Roy, et al. investigated the trade-off between machine automation and user manual control [32]. They introduced the notion of *controllability* of an automated task, which they defined as "how much a user is 'in control' of the process", and to "what extent they can control the automation or alter its result". In other words, controllability is strongly related to how difficult it is to execute a task manually. They use a simple synthetic robot placement operation as the automated task, and manipulated the accuracy of the automation, and the amount of effort required to fix its inaccuracies using manual controls. Their participants demonstrated a strong tendency to rely on manual controls to fix the automation inaccuracies, rather than trying the automation again. We build on this previous work, but focus on a task where users can choose to use the automation or not. This is different than Roy et al.'s task where automation performs the task initially, and users can choose how to fix its inaccuracies. Our work also explore controllability in a more ecological automatable task.

3 OPERATIONALIZING THE ACCURACY OF WORD SUGGESTIONS

In this section, we create a user-centred definition of accuracy for a word suggestion interface and the method we use to "operationalize" word suggestion accuracy as an independent variable in a transcription task. We adopt the general strategy of Roy et al. who operationalized the accuracy of their synthetic automation system by controlling how often it produces a result that achieved the user's goal [32]. For example, 50% accuracy means the system could complete the task for the user half of the time, or could complete half of the user's task. In the case of a text transcription task, the user's goal is to enter characters. So a word suggestion interface with 50% accuracy should be able to accurately suggest half of the characters a user needs to type.

3.1 Accuracy as Keystroke Saving

A common metric to evaluate word suggestion benefits is "keystroke saving" (KS) [35–37]. This is the ratio of the number of keystrokes (excluding edits) that are avoided by using suggestions (N_{sk}), to the number of characters in the phrase ($|P|$):

$$KS = \frac{N_{sk}}{|P|} \tag{1}$$

3.1.1 Defining word suggestion accuracy. In a transcription task, we know in advance exactly what will be typed, so we can operationalize the accuracy of word suggestions around this keystroke saving definition. We define accuracy A_P as the maximum keystroke savings offered by the suggestions for a phrase P with $|P|$ characters (including spaces):

$$A_P = \max KS = \frac{\max N_{sk}}{|P|} \tag{2}$$

Table 1: Example of distribution of keystrokes that can be saved for the phrase "`the rationale behind the decision `" with the corresponding computation of phrase accuracy and SD.

	`the␣`	`rationale␣`	`behind␣`	`the␣`	`decision␣`
maximum saved keystrokes	3	6	2	3	3
accuracy of suggestion (A_w)	3/4	6/10	2/7	3/4	3/9

Standard deviation of the accuracies for words in the phrase: $\mathrm{SD}(A_w) = \mathrm{SD}\ (3/4,\ 6/10,\ 2/7,\ 3/4,\ 3/9) = 0.22$
Accuracy of the phrase: $A_P = 17/34 = 0.5$

max N_{sk} is determined by simulating a user entering text without errors, and selecting correct suggestions as early as possible (i.e. as soon as they are available). A word suggestion system with a high accuracy will show the correct suggestion sooner (ideally after entering the first letter of a word), compared to a system with a low accuracy.

In a similar way, the accuracy of a specific suggestion for a word (A_w) can be defined as the ratio of the maximum number of keystrokes that can be saved, to the number of characters in that word. For convenience, we include a space entered at the end of the word to be a required character. The main reason is that we automatically insert a space after each selected suggestion, and this eases the comparison with words entirely typed using a keyboard (more details in the results section). Also, in contrast to previous work [37], keystrokes used to accept suggestions are ignored (for example tapping on the suggestion bar on mobile devices). This is because we are interested in measuring the contribution of suggestions independently from the interface technique used to trigger them. Ignoring suggestion interface keystrokes makes our results easier to compare with other ways of triggering word suggestions, for example using a technique that may require several keystrokes, or the use of a mouse.

3.1.2 Controlling word suggestion accuracy. To control the accuracy of word suggestions in an experiment, we have to decide how many keystrokes should be saved for each word a participant will have to type. These potentially saved keystrokes determine when the correct suggestion is first shown as a word is typed. As an example, if three keystrokes should be saved from the word "`calm␣`", the correct suggestion would need to appear once the user typed "`ca`", effectively saving keystrokes for "`l`", "`m`", and the following whitespace character. If four keystrokes should be saved for the word "`yet␣`", the correct suggestion should be shown as soon as the previous word is completed. Conversely, if no keystrokes can be saved, the correct suggestion should never be offered.

How many saved keystrokes should be offered for an entire phrase (max N_{sk}) depends on the expected accuracy for this phrase (Equation 2). Note that it is often not possible to reach the exact expected accuracy for a phrase. For example, if there is an odd number of characters to type, one cannot save exactly half of them.

Distributing the potentially saved keystrokes among the words of a phrase is not trivial either. Our first approach was to ensure all words benefit from approximately the same accuracy of word suggestions: for each word w, $A_w \approx A_P$. Specifically, we made the standard deviation of the accuracy for words, $\mathrm{SD}(A_w)$, close to zero. However, unlike in the real world, the appearance of the correct suggestion was then highly predictable. Indeed, if for each word $A_w = .75$, the correct suggestion always appeared when approximately a quarter of the word is typed. Enforcing a higher standard deviation, $\mathrm{SD}(A_w)$, makes it uneven, increasing external validity. Fixing this standard deviation for each phrase, for example $\mathrm{SD}(A_w) = 0.2$, increases internal validity. Table 1 shows an example of a distribution of potentially saved keystrokes.

To summarize, potentially saved keystrokes need to be distributed among the words of each phrase so that:

1. The accuracy of the phrase A_P is as close as possible to the expected accuracy;
2. The standard deviation of the accuracy for words ($\mathrm{SD}(A_w)$) is as close as possible to a chosen non-zero value, for example 0.2.

A brute force tree traversal algorithm is sufficient to find the distributions optimizing these criteria.

3.2 Suggestion Ranking Evolution

When more than one suggestion is displayed, the position of the correct suggestion among all the suggestions also needs to be realistically operationalized. There are two ways to analyze this behaviour on commercial systems: video recording with image processing, which is tedious; or directly using a word suggestion API. We analyzed the behaviour of the Apple macOS Catalina word suggestions using the `NSSpellChecker` API [2].

Using the API, we wrote a small utility to log the ranking of suggested words after each keystroke when a simulated user types every phrase from Mackenzie and Soukoreff's set [25] without errors. To avoid potential adaptation of the suggestion engine, a default blank macOS user account was used. Analysis of the logs show that a correct suggestion is often immediately ranked first the moment it appears, or it very quickly moves up in the ranking. Also, the length of the suggested word has an effect. the shorter the length, the fewer the keystrokes required before the suggestion reaches first rank.

To model the general behaviour of how suggestions transition through rankings, we compute a matrix of probabilities capturing the likelihood of a correct suggestion moving from one rank to another. Given the impact of word length on suggestion rank, we compute this matrix for each suggestion word length (see examples in Fig. 2). This created 13 matrices since all words in Mackenzie and Soukoreff's phrases are between 1 and 13 characters.

These transition matrices can be used to operationalize the evolution of a correct suggestion's ranking. For example, if a maximum of 4 keystrokes may be saved from the word "`doctor␣`", "`doctor`" will first be suggested after the three letters "`doc`" have been typed.

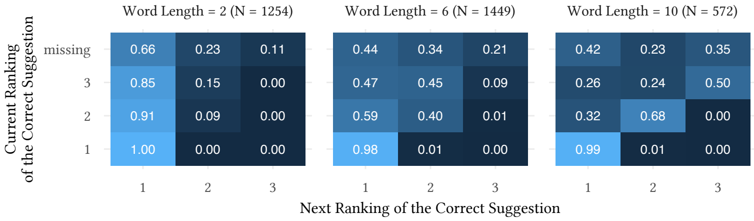

Figure 2: Examples of suggestion ranking transition matrices modelling the behaviour of macOS's suggestion engine. These matrices were calculated for each possible word length.

According to the second transition matrix of Fig. 2, there is a 39% probability it is immediately ranked first, 35% second, and 26% third. If it is ranked third, and the user types "t", there is a 45% probability it moves second. Interestingly, at any point in time, there is virtually no chance the rank of the correct suggestion decreases if the user types a correct character.

The transition matrices computed from the logs, and source code for the logger, are available from an associated project page[3].

4 EXPERIMENT

The goal of our experiment is to investigate the relationship between typing efficiency, and the accuracy of word suggestions. A transcription task with word suggestions grouped in a bar interface is used. Typing efficiency is controlled by varying the device used for the task: a desktop computer, a landscape-oriented tablet, or a phone used with one hand. Accuracy is the theoretical maximum number of saved keystrokes from suggestions using the method just described, with the placement and evolution of a correct suggestion in the interface based on transition matrices.

Our analysis focuses on how often suggestions are used as a function of the two main factors, and also their effect on task performance and user satisfaction. Inspired by Roy et al.'s results [32], we made the following hypotheses: (H1) Word suggestion usage increases with suggestion accuracy; (H2) Word suggestion usage is greater on a phone than a tablet, and greater on a tablet than a desktop.

Due to emergency measures resulting from the 2020 COVID-19 pandemic, the experiment was facilitated remotely using a web application and live video-conferencing.

4.1 Participants

We analyze the data of 36 participants recruited using mailing lists and social networks (ages 18 to 53, average 29.8, SD = 8.3, 13 self-declared as female, and 23 as male). Participants were required to have access to a smartphone, a touchscreen tablet, and a laptop or desktop computer equipped with a physical keyboard using a QWERTY mapping. They received a $10 gift card for the one hour study.

Based on a pre-questionnaire, 31 (86.1%) participants reported spending more than 4 hours typing on a desktop computer in the past 7 days. Only 2 participants (5.6%) reported similar habits for tablets, and 12 (33.3%) for phones. Six (16.7%) participants reported spending more than 1 hour typing with one hand on a phone in the past 7 days. Regarding their estimated word suggestion usage, 7 (19.4%) participants reported using more than 30 word suggestions when typing on a desktop over the past 24 hours. On a tablet, no participants reported having used more than 30 word suggestions in the past 24 hours, 30 (83%) reported using less than 10. On a phone, 10 participants (27.8%) reported using more than 30 word suggestions during the past 24 hours.

At the start of the session, the participant transcribed 10 phrases without suggestions to measure their natural typing speed. On average, participants typed 76.5 words per minutes (wpm) on desktop (SD = 0.6), 35.0 wpm on tablet (SD = 1.2), and 29.9 wpm on phone (SD = 1.1).

4.2 Apparatus

Participants used their own devices. The average screen diagonal of their desktop was 424 mm (16.7 inches, SD = 6 mm), 255 mm for their tablet (10.0 inches, SD = 2 mm), and 142 mm for their phone (5.6 inches, SD = 1 mm). We were not able to gather statistics about processing power because of browser security measures.

The experiment software was developed as three parts. A *client web application* ran on all devices. This dynamic page guided the participant through the protocol steps and presented the transcription and word suggestion interfaces to complete the measured tasks. Importantly, the tablet and phone version of the client provided an embedded keyboard that had to be used. This gave us needed control over the word suggestions and the bar interface, and controlled the type of keyboard that was used. An *administration server* implemented in JavaScript delivered static assets to the client, and enabled the experimenter to monitor and remotely control the client to facilitate the experiment and troubleshoot any issues. A *suggestion server* implemented in Go computed the word suggestions for the client. All source code is available from the project page[3].

The embedded client keyboard was designed after iOS and iPadOS's keyboards (see Fig. 3). On the phone, the Shift technique [39] was implemented to reduce the effect of occlusion. This shows a preview of the key being pressed on top of the finger, and is consistent with most implementations of phone keyboards.

4.3 Task

Each trial was a transcription task using phrases from Mackenzie and Soukoreff's 500-phrases set [25], which was designed to be representative of the English language. Participants were instructed to type as fast and precisely as possible. To insert or modify an existing character, participants had to delete all characters between

[3] https://ns.inria.fr/loki/WordSuggestions

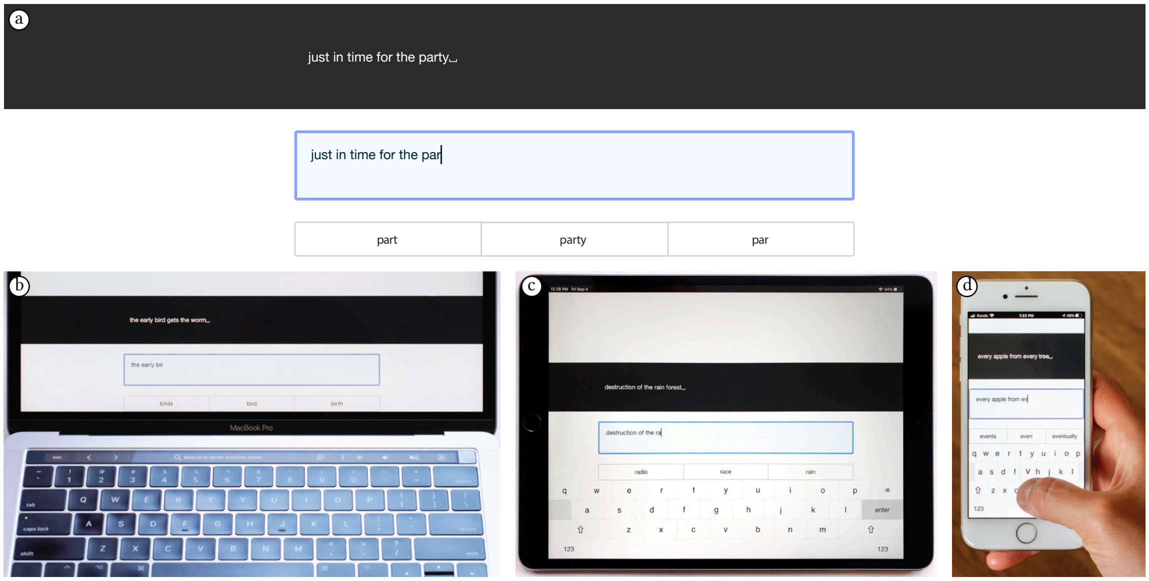

Figure 3: ⓐ User interface for the experiment, ⓑ laptop condition, ⓒ tablet condition, and ⓓ phone condition. Phrases to copy appear in the dark grey area in the middle. Participant input appears in the blue box underneath it.

the insertion point and that character using the backspace key (i.e. they could not use cursor keys or position the cursor by pointing). To complete a trial, participants had to transcribe exactly the stimulus phrase, which required all errors have to be fixed. While previous study about word suggestions fully prevented errors [31], recent work showed a user's aversion for errors has a significant effect on their typing behaviour [6, 7]. Consequently, we did not prevent errors in our experiment. Any characters could be entered, but the phrase set only uses alphabetical characters, mostly lowercase (99.8%).

Because a whitespace character is appended after each accepted suggestion, we added a terminal whitespace to key in at the end of every phrase. This ensures that the suggestion behaviour is equally beneficial for each word of the phrase, including the last one.

The same basic user interface was used for all devices (Fig. 3). For tablet and phone, the suggestions bar appeared just above the keyboard, like commercial interfaces. For desktop, the bar was positioned at the very bottom of the display, near the physical keyboard to approximate the position of the Apple Touchbar when used to display word suggestions. For all devices, the phrase stimulus and text entry field was positioned near the word suggestion bar. Selecting a suggestion on phone or tablet used direct touch. On the desktop, to avoid using the mouse or cursor keys, and to best approximate the position and action of direct input on a Touchbar, keys 1, 2 and 3 were used to select suggestions. Even if inline suggestions are more common on desktop, we used a suggestion bar on all three devices to avoid confounding the experiment.

When typing on the tablet, participants were told to lay it on a table in landscape orientation. The intention was to enable the most efficient 10-finger touch typing style for the tablet condition. When typing on the phone, participants were told to use only their dominant hand, both to hold the phone, and to type using a thumb. In addition, they held the phone in the air without supporting their hand. These constraints were designed to reduce typing efficiency with phone, in an attempt to explore a larger segment of the typing efficiency scale. This condition is consistent with real-world habits of 12.7% of phone users according to a 2018 study [11], and 36% according to a 2012 study [17].

The accompanying video figure demonstrates the task on all three devices as well as a task tutorial each participant completed.

4.4 Suggestions

After each keystroke, the current word prefix is sent to a server that evaluates matching English words. We use Google Android Jelly Bean's dictionary [8], and replicate its scoring algorithm [9]. The word with the best score is suggested, with the two following exceptions:

1. the target word w is always suggested as soon as possible, in accordance to A_w,
2. any other suggestions that may save keystrokes are prevented.

As an example to illustrate the second point, assume the next word the participant has to type is "select" and that this word has been assigned two potential saved key strokes ($A_w = 2/6$). At first, suggesting "selected␣" is prevented as it could save $6 - 3 = 3$ keystrokes in total if accepted by the participant: 6 saved keystrokes because "select" would not need to be typed anymore, -3 for "e", "d" and the automatically inserted whitespace that will need to be deleted. However, once the participant has typed "sel", "selected␣" would not save keystrokes anymore as $3 - 3 = 0$. Therefore it would not be prevented if it is ranked high on the suggestion scoring algorithm.

Our software enforces a minimum 150 ms delay before updating suggestions after each keystroke. In addition, trials for which this delay exceeded 300 ms due to network latency were removed from the analysis. A 150 ms to 300 ms delay before suggestions is lower than the 350 ms to 850 ms delays we measured with popular commercial systems[4], so our study generalizes to more responsive suggestion systems in the future.

We operationalized the accuracy of these suggestions as discussed in section 3. However, the ranking evolution slightly deviated from what we modelled due to an implementation inconsistency. This primarily affected the first ranking when the correct word moved from not being suggested to being suggested and the ranking of less frequent words with more than 8 characters. Since our transition modelling captures only one example of many word suggestion implementations, such a small deviation would not undermine the ecological validity. In addition, we found suggestions were as likely to be used regardless of their ranking (see Section 5.1.1). An explanation of the deviation and the exact transition matrices used in the experiment are available on the project's webpage[5].

We targeted a standard deviation of the accuracy for words $\mathrm{SD}(A_w) = 0.2$ as it is the highest standard deviation value that can be reached for $A_P = 0.1$ or 0.9, according to the König-Huygens theorem. This is consistent with the Apple macOS `NSSpellChecker` API for which we measured $\mathrm{SD}(A_w) = 1.79$ on average with 1 suggestion, and $\mathrm{SD}(A_w) = 1.76$ with 3 suggestions. We allowed a margin of error of 0.025 for the accuracy of the phrase (A_P), and of 0.1 for $\mathrm{SD}(A_w)$. Only 6 out of the 1,500 PHRASE × ACCURACY combinations did not allow these conditions to be met and were removed.

4.5 Procedure

The experiment was divided into two parts. During the first part, participants measured the size of their three displays, performed a short 10-trial natural typing speed test without word suggestions, completed a demographic questionnaire, then went through a short interactive tutorial demonstrating the task.

The second part of the experiment was divided into three sections, one for each device condition. Each section had 3 practice trials, 20 trials recorded for analysis, a subjective questionnaire (explain in results below), and a NASA-TLX questionnaire [18].

A dedicated interface was used by the experimenter to control each participant's progress, and remotely launch the different parts that composed the experiment.

4.6 Design

We used a mixed design with two independent variables:

- DEVICE { DESKTOP, TABLET, PHONE } (within-subject),
- ACCURACY { 0.1, 0.5, 0.9 } (between-subject).

DEVICE was within-subject to reduce inter-participant variability. Its order was counter-balanced using a balanced Latin square. ACCURACY was between-subject to avoid strong carryover order effects if administered within-subject: a participant first exposed to a low accuracy condition is less likely to use a high accuracy condition after, with the inverse if exposed to a high accuracy first. ACCURACY levels were chosen to explore the full ACCURACY spectrum. The potential keystroke saving of state-of-the-art word suggestion algorithms is less than 46% [14]. We measured 55% (SD = 14%) potential keystroke saving with Apple's `NSSpellChecker` API with one suggestion on Mackenzie and Soukoreff's phrase set [25], 65% (SD = 12%) with two suggestions, and 69% (SD = 11%) with three suggestions.

Participants were assigned to ACCURACY levels to minimize between group differences of mean natural entry speeds. The greatest pairwise difference between ACCURACY groups was 1.7 wpm for phone, 2.0 for tablet, and 0.5 for desktop. We did not specifically balance age or gender. In practice, they were relatively well distributed among accuracy values. The largest average age difference was 4.4 years between accuracy 0.5 and accuracy 0.1. There were only 2 fewer females and 2 more males for accuracy 0.1 than across the two other conditions.

Phrases to copy were randomly sampled without replacement from the Mackenzie and Soukoreff phrase set [25]. This was specifically designed to contain letter frequencies matching the English language. Our random sampling has similar letter frequencies to the entire phrase set, and there is little difference between conditions: $f(\text{'e'}) = 0.13$, $f(\text{'t'}) = 0.09$, $f(\text{'o'}) = 0.08$, and so on. This is also consistent with letter frequencies in the English language [34].

In summary: we recorded 20 PHRASE × 3 DEVICE × 3 ACCURACY × 12 participants per ACCURACY condition = 2,160 trials.

5 RESULTS

Before analysis, 11 trials out of 2,160 were removed because suggestions took more than 300 ms to be updated from the server. Four additional trials were removed because the webpage lost focus. All other trials were kept, including trials during which participants mistyped (errors had to be fixed). We provide the data and R-markdown analysis notebooks on the project's webpage[5].

5.1 Objective Measurements

In the analysis to follow, ANOVA was used. For each measure, trials were aggregated by participant and factors being analyzed. Because all measures exhibited non-normality of the residuals, and non-homogeneity of their variances, we applied an Aligned Rank Transform beforehand [41]. Tukey HSD post hoc tests were used for pairwise comparisons of main effects. Interaction Contrasts were used for cross-factor comparisons in case of interaction.

5.1.1 Suggestion Usage.

Suggestion Usage is the mean number of suggestions used during trials. As illustrated in Fig. 4, suggestions were used slightly more on phones (1.9 suggestions per trials) than tablets (1.5), and hardly used at all on desktop computers even when accuracy was high (0.3). *Suggestion Usage* increases from less than 0.5 suggestions per trials at 0.1 ACCURACY to more than 3.3 at 0.9 ACCURACY for both tablet and phone. It does not exceed 0.9 SUGGESTIONS per trials on desktop, even at 0.9 ACCURACY.

[4] These estimates are by manually inspecting video screen recordings from iOS, iPadOS and macOS. On macOS, we used the accessibility keyboard to display a copy of the touch bar on the main screen, allowing it to be recorded. On iOS and iPadOS, we used the native screen recording feature. Our analysis revealed that 200 ms generally elapse from the moment a character is added to user's input, to the start of the animation updating suggestions. On iOS and iPadOS, the update animation lasted from 150 ms to 650 ms in our recordings; during most of this time, suggestions are not readable. On macOS, there is no animation. However, the suggestions are not updated after each keystroke but only when the user pauses for approximately 350 ms.

[5] https://ns.inria.fr/loki/WordSuggestions

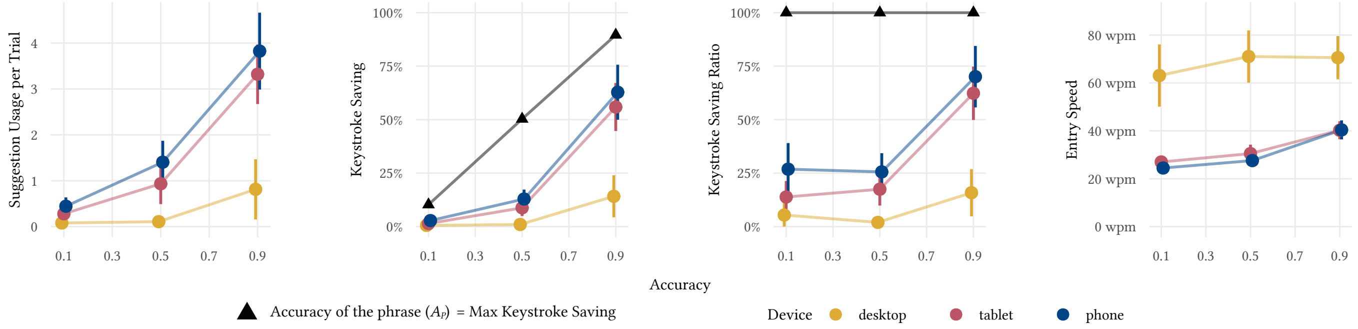

Figure 4: Use and contribution of the suggestions, and entry speed. Effect sizes show .95 confidence interval. The keystroke saving ratio is the keystroke saving by the accuracy of the phrase.

These observations are supported by statistical test. There is a significant main effect of DEVICE ($F_{2,66}$ = 80.5, $p < .0001$), with significant differences between every pairs, all $p < .0001$ except [TABLET, PHONE] ($p < .05$). There is also an effect of ACCURACY ($F_{2,33}$ = 32.8, $p < .0001$), with significant differences between all pairs, $p < .05$ for ACCURACY [0.1, 0.5], $p < .001$ for ACCURACY [0.5, 0.9], and $p < .0001$ for ACCURACY [0.1, 0.9]. Finally, we found a DEVICE × ACCURACY interaction ($F_{4,66}$ = 24.4, $p < .0001$). The difference between DESKTOP and TABLET or DESKTOP and PHONE is larger for ACCURACY 0.5 than ACCURACY 0.1 ($p < .05$), for ACCURACY 0.9 than ACCURACY 0.5 ($p < .0001$), and for ACCURACY 0.9 than ACCURACY 0.1 ($p < .0001$). No difference between TABLET and PHONE were found regardless of ACCURACY.

We found no effect of a correct suggestion's first ranking on *Suggestion Usage* (p = 0.59). The correct suggestion was not more likely to be used regardless whether it was first ranked first, second, or third in the bar interface.

5.1.2 *Keystroke Saving.*

Each used suggestion does not contribute equally to the task as they may have been used to save a different amount of keystrokes. *Keystroke Saving* is the ratio of number of saved keystrokes using suggestions excluding editing (N_{sk}), by the number of characters to type ($|P|$) [36]:

$$KS = \frac{N_{sk}}{|P|} \tag{3}$$

Note that the keystroke saving is at most equal to the accuracy of word suggestions for the phrase: $\max KS = A_P$. Compared to *Suggestions Usage*, *Keystroke Saving* is a more accurate measure of the added value of word suggestions.

As illustrated in Fig. 4, *Keystroke Saving* followed a pattern similar to *Suggestions Usage*, but tighter, blurring the differences between phone (26%) and tablet (22%). Only 5% of the keystrokes were saved on desktop. *Keystroke Saving* raises from 2% at 0.1 accuracy to an average 44% at 0.9. However, it barely reaches 15% on desktop for 0.9 accuracy, while it exceeds 55% for tablet, and 62% for phone.

These observations are supported by a significant main effect of DEVICE ($F_{2,66}$ = 66.6, $p < .0001$), with significant differences for [DESKTOP, TABLET], and [DESKTOP, PHONE] ($p < .0001$). There is an effect of ACCURACY too ($F_{2,33}$ = 49.8, $p < .0001$), with significant differences between every pairs, $p < .0001$ for ACCURACY [0.1, 0.9] and ACCURACY [0.5, 0.9], and $p < .01$ for ACCURACY [0.1, 0.5]. Finally, there is a DEVICE × ACCURACY interaction ($F_{4,66}$ = 28.9, $p < .0001$). The difference between DESKTOP and PHONE is larger for ACCURACY 0.5 than 0.1 ($p < .05$), for ACCURACY 0.9 than ACCURACY 0.5 ($p < .0001$), and for ACCURACY 0.9 than 0.1 ($p < .0001$). The difference between DESKTOP and TABLET is larger for ACCURACY 0.9 than for ACCURACY 0.5 ($p < .0001$), and for ACCURACY 0.9 than for ACCURACY 0.1 ($p < .0001$). No difference between TABLET and PHONE were found regardless of ACCURACY.

5.1.3 *Entry Speed.*

We measured *Entry Speed* in words-per-minute (wpm), where "word" means 5 characters [3, 43], calculated as follows:

$$S = \frac{|P| - 1}{T} \times \frac{60}{5} \tag{4}$$

$|P|$ is the number of characters in the phrase to transcribe, and T the interval of time in seconds from the moment the first input character is entered to the moment the transcribed text matches the target phrase. One character is subtracted from $|P|$ in the calculation of *Entry Speed* because T is only measured from the first input character.

As illustrated in Fig. 4, participants were more than twice as fast with their desktop (68.2 wpm) than with their tablet (32.6 wpm) or phone (30.8 wpm). Higher accuracy did slightly increase entry speed, in particular for the phone and tablet, but its effect was small, even at its highest. *Entry Speed* increased from (38.2 wpm) at 0.1 accuracy to (50.4 wpm) at 0.9.

Statistical support is from a significant effect of DEVICE on *Entry Speed* ($F_{2,66}$ = 118.0, $p < .0001$). Pairwise comparisons detected significant differences for [DESKTOP, TABLET], and [DESKTOP, PHONE] ($p < .0001$). There is also an effect of ACCURACY ($F_{2,33}$ = 8.6, $p < .0001$), with a single pairwise difference for ACCURACY [0.1, 0.9] ($p < .001$).

5.1.4 *Keystroke Saving and Natural Entry Speed.*

We explored the correlation between *Natural Entry Speed* (without suggestions) measured at the beginning of the experiment, and *Keystroke Saving*. Results are shown at the top of Fig. 5. *Keystroke Saving* decreases with *Natural Entry Speed*, regardless of DEVICE: Kendall's taus are $r_\tau = -.33$, $p < .01$ for ACCURACY 0.1, $r_\tau = -.62$, $p < .0001$ for ACCURACY 0.5, and $r_\tau = -.56$, $p < .0001$ for ACCURACY 0.9.

5.1.5 *Entry Speed Improvement.*

Finally, we investigated the effect of word suggestions on entry speed compared to natural entry speed. *Entry Speed Improvement* is the difference in *Entry Speed* with suggestions, as measured during the experiment; and *Natural Entry*

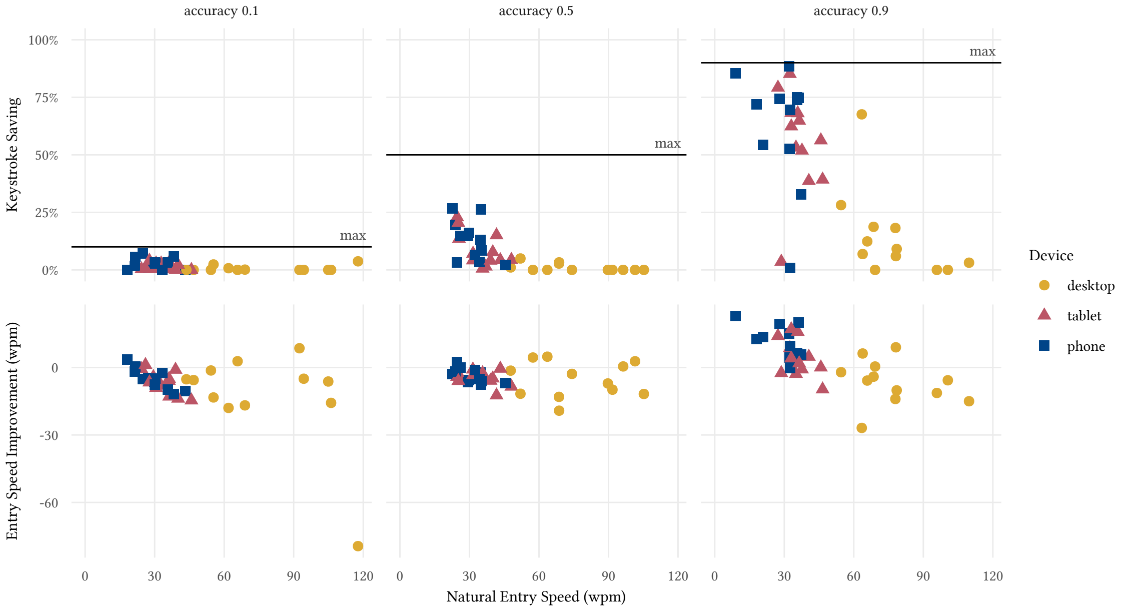

Figure 5: Keystroke Saving (top) and Entry Speed Improvement (bottom) as a function of Natural Entry Speed. Participant S25, the fastest typist, lost 79 wpm with 10% accurate word suggestions. Unlike other fast typists, they saved 4% of their keystrokes, i.e. 36% of the keystrokes that could be saved, even as accuracy was very low.

Table 2: Subjective Questionnaire

Dependent Variable	Assertion
Perceived Accuracy	The word suggestions are accurate
Perceived Keyboard Efficiency	The use of the keyboard is efficient in this task
Satisfaction	The controls (keyboard and word suggestions) are satisfactory for the completion of the task
Suggestion Disruptivity	The word suggestions are distracting

Speed, as measured at the beginning of the experiment. Results are shown at the bottom of Fig. 5. Most of the time, word suggestions negatively impacted *Entry Speed*. They were only beneficial for slow typists and when very accurate. Except for ACCURACY 0.5, the faster the typist while typing without suggestions, the more their entry speed will be reduced with word suggestions: Kendall's taus are $r_\tau = -.36$, $p < .01$ for ACCURACY 0.1, and $r_\tau = -.50$, $p < .0001$ for ACCURACY 0.9.

5.2 Subjective Questionnaire

Table 2 provides the questionnaire's assertions with associated dependent variables. Answers were collected on a seven-point Likert scale, "Strongly Agree" to "Strongly Disagree". In the analysis below, "$\ll$" indicates significant difference with $p < .05$ or lower. The results of our questionnaire scale are shown in Fig. 6.

We also applied an Aligned Rank Transform on our subjective measurements, which are all ordinal, in order to investigate interactions. Tukey HSD post hoc tests were used for pairwise comparisons of main effects. Interaction Contrasts were used for cross-factor comparisons in case of interaction. The results of ANOVA significance tests are provided in Table 3.

5.2.1 Perceived Accuracy. Our results demonstrate that the operationalization of ACCURACY matches the perception of users. We found significant *Perceived Accuracy* differences for the following ACCURACY values: 0.1 $\ll$ 0.5 $\ll$ 0.9.

5.2.2 Perceived Keyboard Efficiency. This partially supports our operationalization of typing efficiency. Participants perceive a desktop computer as more usable for a typing task than both phone and tablet. Surprisingly, there is little difference between one-handed typing on a phone and typing with two hands on a tablet. We found { TABLET, PHONE } $\ll$ DESKTOP.

5.2.3 Satisfaction. Our results show *Satisfaction* increases with ACCURACY, and participants were more satisfied on desktop than tablet or phone. We found ACCURACY { 0.1, 0.5 } $\ll$ 0.9 and { TABLET, PHONE } $\ll$ DESKTOP.

5.2.4 Suggestion Disruptivity. Participants found accurate suggestions less disruptive than inaccurate suggestions: ACCURACY { 0.1, 0.5 } $\ll$ 0.9.

5.3 NASA-TLX

Few significant results were detected from our NASA-TLX data (see Fig. 6). The results of ANOVA tests after Aligned Rank Transform are provided in Table 3. *Frustration Level* and *Physical Demand* were

Table 3: Statistical significance of the questionnaire and NASA-TLX of Experiment 3

	DEVICE		ACCURACY		DEVICE × ACCURACY	
	$F_{2,99}$	p	$F_{2,99}$	p	$F_{4,99}$	p
Questionnaire						
Perceived Accuracy	0.32	.73	74.36	<.0001	2.09	.09
Perceived Usability	22.09	<.0001	0.16	.85	0.23	.92
Satisfaction	5.38	<.01	7.61	<.001	1.79	.14
Suggestions' Disruptivity	2.66	.08	8.73	<.001	1.25	.29
NASA-TLX						
Effort	3.56	<.05*	0.40	.67	0.47	.76
Frustration Level	8.01	<.001	0.95	.39	0.75	.56
Overall Performance	1.88	.16	3.05	.05	3.49	<.05
Mental Demand	2.80	.07	1.56	.22	0.31	.87
Physical Demand	8.64	<.001	3.04	.05	1.26	.29
Temporal Demand	2.16	.12	2.76	.07	0.55	.70

*No significant pairwise differences

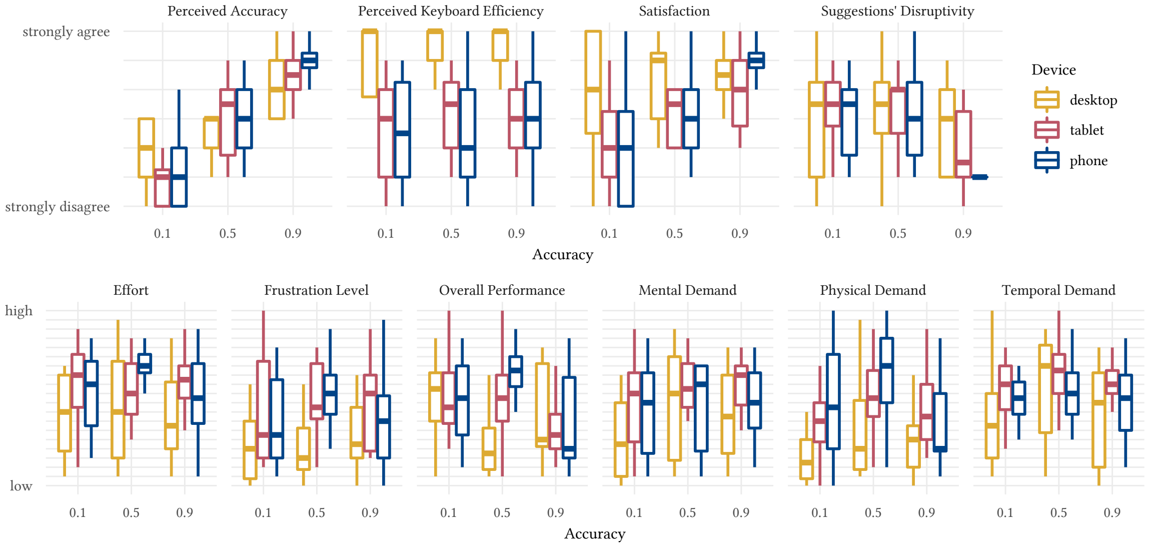

Figure 6: Subjective questionnaire (top) and NASA-TLX (bottom) of Experiment 1

both lower with desktop than with phone and tablet: DESKTOP ≪ { TABLET, PHONE } for both measures.

6 DISCUSSION

Our results demonstrate how reliance on word suggestions not only depends on their accuracy, but also on typing efficiency. We have shown that the use of word suggestions, and their contribution to the completion of a typing task, increases with suggestion accuracy but decreases relative to how fast it is to type on a device. This confirms H1 (word suggestion usage increases with their accuracy) and H2 (word suggestion usage is greater on the phone than tablet, and greater on tablet than phone). However, we expected participants to be able to type much faster with two hands on a tablet than with one hand on a phone, so we also expected suggestion usage on a phone to be well below the tablet. We did measure a significant difference, but we were surprised how small it is. Likely because of this, we found no significant differences in terms of keystroke saving. We believe this is explained because natural entry speed without suggestions was unexpectedly similar in the two conditions: 30 wpm for PHONE and 35 wpm for TABLET. As shown in Fig. 5, we observed a strong correlation between natural entry speed and keystroke saving. The slower typing speed with two hands on tablet compared to one hand on phone is most likely due to experience and training: our demographic data revealed participants were much more used to typing on a phone than a tablet.

This is also supported by how participants perceived phone and tablet typing efficiency similarly.

An important take away from H2 is that acceptable accuracy results may not translate from one typing system to another. For example, while many studies on word suggestions were performed with Augmentative and Alternative Communication systems (AAC) [12, 15, 22, 37], their results are not applicable to phones or desktops. Greater generalizability may only be achieved by considering typing efficiency.

Our results also show that the effect of accuracy on the contribution of word suggestions to the completion of a phrase is non-linear: the lower the accuracy is, the less the potential of suggestions is put into use (Fig. 4). In the highest accuracy condition of our experiment (Accuracy 0.9) suggestions were used at 50% of their full potential across devices, up to 70% on the phone. However, for levels Accuracy 0.1 and 0.5, only 15% of the keystrokes that could potentially be saved with suggestions were actually saved. And this was a very low 5% on desktop.

This hints that most users get a sense of how much suggestions can help them. They tend to disregard them if accuracy is low or if they can type quickly, but they use them more if accuracy is higher or if they type slowly. In a more systematic way, this confirms previous work. Buschek et al. observed that suggestion use was highly individual [11]. Later, Palin et al. note that slower typists use more suggestions [27].

Beyond usage, even if keystroke saving steadily increases with word suggestion accuracy, the effect is smaller than that observed by Banovic et al. with automatic error corrections [7]. In fact, our results indicate that word suggestions are almost always detrimental to entry speed except for a slow typist when accuracy is very high (see Fig. 5). It is worth noting that the potential keystroke saving of state-of-the-art word suggestion algorithms from the literature is less than 46% [14]. However, we measured 55% potential keystroke saving with Apple's `NSSpellChecker` API when one suggestion was provided for phrases in the Mackenzie and Soukoreff set, and 69% with three suggestions. However, subjective ratings expose a slightly different pattern: satisfaction increases with accuracy, even in the lower accuracy range where it does not improve performance. This trend confirms previous observations from Quinn et al., who noticed that users value assistance, even when it impedes their performance [30]. A higher-quality dictionary will increase accuracy, for example one that becomes tailored to phrases commonly used by the typists. Our results suggest that this may increase user satisfaction, but it may not create a real benefit in text entry performance.

From a higher level perspective, as discussed in section 2.3, automating, or partially automating a user's task is often done with the intention to improve usability. Word suggestion interfaces are an instance of such automation. The success of an automation depends on its purpose, but also its accuracy. The question is, how accurate does it need to be? Roy et al. showed that both automation accuracy and the "controllability" of a manual interface impact how users choose to fix automation inaccuracies [32]. Our results can be integrated in this framework, and tend to confirm their results: the more usable a manual interface is, the less an automation system will be used even if it is highly accurate. In our case, the manual interface is the keyboard, and the automation is the word suggestions. We suspect this pattern likely generalizes to other common interfaces and tasks beyond word suggestions and text entry.

Overall, our results further show that dedicating resources to improving usability is always a winning bet for user performance. While improving accuracy will result in higher satisfaction, it may not ultimately increase performance, and more critically, it is unlikely to compensate for poor usability like poor typing interfaces or devices. In the context of word suggestions, academic and industry resources may be best allocated for optimizing input efficiency—for example with the help of faster text entry techniques such as ShapeWriter gesture typing [23]—rather than improving word suggestions. This has important implications for the design of related text entry techniques used on millions of devices.

7 LIMITATIONS

While the devices we used have strong ecological validity, it does not provide a formal control for levels of typing efficiency. Considering natural entry speed helps, but we could not control it either. A more controlled and synthetic experiment that tightly controls typing efficiency would further contribute to our understanding of the impact of typing efficiency on suggestion usage and entry speed.

Some participants reported being less efficient than usual with the keyboard integrated in our mobile application because it differs from the one they are used to use with their device. While this does not put into question our pattern of results, and we did measure natural entry speed using our keyboard, it indicates the entry speed measured in our experiment under-represents the population of trained typists on phone and tablet.

Operationalizing accuracy may sometimes feel surprising to participants. In the 0.9 condition, some noted the word suggestions were "unnaturally accurate". On the contrary, in the 0.1 condition, some indicated they felt like the system was avoiding the correct word on purpose (it was). A few participants also noted the system would occasionally not suggest common words, like "`the`"; and surprisingly, it would correctly suggest much less common words, like "`racketball`". This reveals an unavoidable bias in our participants regarding their prior experience and expectations of word suggestion systems.

Finally, this experiment investigated a suggestion bar, but word suggestions may be presented in different ways. For example, on a desktop, suggestions are often inline with the document input, after or under the insertion point (see Fig. 1). It remains unclear if the use of inline suggestions follows the same pattern as our results. In addition, suggesting more than three words increases the chance of suggesting the right one, and as a result increases accuracy. However, a previous study showed it also increases cognitive load, quickly offsetting the benefits [35, 38]. We investigated a three-word suggestion bar in our experiment, though in practice, commercial systems often show only two suggestions. The first position on the bar is often used to validate the word prefix as it currently is, preventing automatic correction that would otherwise trigger if the user pressed space (see Fig. 1a and 1b). It remains unclear how varying the number of suggestions would impact the trend we observed.

8 CONCLUSION

On desktop computers, we found word suggestions never improve entry speed, even when extremely accurate. They do on phones and tablets, but typically only when accuracy is very high. However, accuracy improves user satisfaction. These results have direct implications for the design of typing systems, and could justify prioritizing decisions and resources for industry and research.

An important take away from our work is that a useful accuracy for word suggestions strongly depends on the device used to type. This is implies that results for the acceptable accuracy of word suggestions does not generalize from one typing system to another, unless typing efficiency is considered.

Our results also open directions for future work. For example, the effect of standard deviation of the accuracy for words remains unclear. Smaller values may help users better estimate when an appropriate suggestion will become available. Likewise, some modern suggestion algorithms propose not only a single word, but an entire sentence [5]. This is powerful but difficult to do accurately. The effect of this strategy, and its accuracy on suggestion use, have yet to be formally investigated.

ACKNOWLEDGEMENTS

This work was made possible by NSERC Discovery Grant 2018-05187, the Canada Foundation for Innovation Infrastructure Fund 33151 "Facility for Fully Interactive Physio-digital Spaces," Ontario Early Researcher Award ER16–12–184, and LAI Réapp.

REFERENCES

[1] Denis Anson, Penni Moist, Mary Przywara, Heather Wells, Heather Saylor, and Hantz Maxime. 2006. The Effects of Word Completion and Word Prediction on Typing Rates Using On-Screen Keyboards. *Assistive Technology* 18, 2 (2006), 146–154. https://doi.org/10.1080/10400435.2006.10131913

[2] Apple. 2020. NSSpellChecker: An interface to the Cocoa spell-checking service. https://developer.apple.com/documentation/appkit/nsspellchecker Accessed: 2020-09-07.

[3] Ahmed Sabbir Arif and Wolfgang Stuerzlinger. 2009. Analysis of Text Entry Performance Petrics. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, Vol. 43. 100–105.

[4] Kenneth C. Arnold, Krysta Chauncey, and Krzysztof Z. Gajos. 2020. Predictive Text Encourages Predictable Writing. In *Proceedings of the 25th International Conference on Intelligent User Interfaces* (Cagliari, Italy) *(IUI '20)*. Association for Computing Machinery, New York, NY, USA, 128–138. https://doi.org/10.1145/3377325.3377523

[5] Kenneth C. Arnold, Krzysztof Z Gajos, and Adam T. Kalai. 2016. On Suggesting Phrases vs. Predicting Words for Mobile Text Composition. In *Proceedings of the 29th Annual Symposium on User Interface Software and Technology (UIST '16)*. ACM, New York, NY, USA, 603–608. https://doi.org/10.1145/2984511.2984584

[6] Nikola Banovic, Varun Rao, Abinaya Saravanan, Anind K. Dey, and Jennifer Mankoff. 2017. Quantifying Aversion to Costly Typing Errors in Expert Mobile Text Entry. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (Denver, Colorado, USA) *(CHI '17)*. Association for Computing Machinery, New York, NY, USA, 4229–4241. https://doi.org/10.1145/3025453.3025695

[7] Nikola Banovic, Ticha Sethapakdi, Yasasvi Hari, Anind K. Dey, and Jennifer Mankoff. 2019. The Limits of Expert Text Entry Speed on Mobile Keyboards with Autocorrect. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services* (Taipei, Taiwan) *(MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA, Article 15, 12 pages. https://doi.org/10.1145/3338286.3340126

[8] Android Jelly Bean. 2012. List of English words with their adjusted frequency of appearance. https://android.googlesource.com/platform/packages/inputmethods/LatinIME/+/jb-release/dictionaries/en_us_wordlist.xml Accessed: 2019-07-08.

[9] Android Jelly Bean. 2012. Word prediction scoring algorithm. https://android.googlesource.com/platform/packages/inputmethods/LatinIME/+/jb-release/native/jni/src/correction.cpp#1098 Accessed: 2019-07-08.

[10] Xiaojun Bi, Shiri Azenkot, Kurt Partridge, and Shumin Zhai. 2013. Octopus: Evaluating Touchscreen Keyboard Correction and Recognition Algorithms Via. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Paris, France) *(CHI '13)*. Association for Computing Machinery, New York, NY, USA, 543–552. https://doi.org/10.1145/2470654.2470732

[11] Daniel Buschek, Benjamin Bisinger, and Florian Alt. 2018. ResearchIME: A Mobile Keyboard Application for Studying Free Typing Behaviour in the Wild. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (Montreal QC, Canada) *(CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–14. https://doi.org/10.1145/3173574.3173829

[12] Patrick W Demasco and Kathleen F McCoy. 1992. Generating Text from Compressed Input: An Intelligent Interface for People with Severe Motor Impairments. *Commun. ACM* 35, 5 (may 1992), 68–78. https://doi.org/10.1145/129875.129881

[13] Mark Dunlop and John Levine. 2012. Multidimensional Pareto Optimization of Touchscreen Keyboards for Speed, Familiarity and Improved Spell Checking. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '12)*. ACM, New York, NY, USA, 2669–2678. https://doi.org/10.1145/2207676.2208659

[14] Andrew Fowler, Kurt Partridge, Ciprian Chelba, Xiaojun Bi, Tom Ouyang, and Shumin Zhai. 2015. Effects of Language Modeling and Its Personalization on Touchscreen Typing Performance. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems (CHI '15)*. ACM, New York, NY, USA, 649–658. https://doi.org/10.1145/2702123.2702503

[15] Nestor Garay-Vitoria and Julio Abascal. 2006. Text prediction systems: a survey. *Universal Access in the Information Society* 4, 3 (mar 2006), 188–203. https://doi.org/10.1007/s10209-005-0005-9

[16] Surjya Ghosh, Kaustubh Hiware, Niloy Ganguly, Bivas Mitra, and Pradipta De. 2019. Does Emotion Influence the Use of Auto-Suggest during Smartphone Typing?. In *Proceedings of the 24th International Conference on Intelligent User Interfaces* (Marina del Ray, California) *(IUI '19)*. Association for Computing Machinery, New York, NY, USA, 144–149. https://doi.org/10.1145/3301275.3302329

[17] J.E. Gold, J.B. Driban, N. Thomas, T. Chakravarty, V. Channell, and E. Komaroff. 2012. Postures, typing strategies, and gender differences in mobile device usage: An observational study. *Applied Ergonomics* 43, 2 (2012), 408 – 412. https://doi.org/10.1016/j.apergo.2011.06.015 Special Section on Product Comfort.

[18] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index); Results of Empirical and Theoretical Research. In *Human Mental Workload*, Peter A Hancock and Najmedin Meshkati (Eds.). Advances in Psychology, Vol. 52. North-Holland, 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

[19] Dan Jurafsky. 2000. *Speech & language processing.* Pearson Education India.

[20] Matthew Kay, Shwetak N. Patel, and Julie A. Kientz. 2015. How Good is 85%?: A Survey Tool to Connect Classifier Evaluation to Acceptability of Accuracy. *Proceedings of the ACM Conference on Human Factors in Computing Systems (CHI 2015)* (2015), 347–356. https://doi.org/bqd5

[21] Heidi Horstmann Koester and Simon Levine. 1996. Effect of a word prediction feature on user performance. *Augmentative and Alternative Communication* 12, 3 (1996), 155–168. https://doi.org/10.1080/07434619612331277608

[22] H H Koester and S P Levine. 1994. Modeling the speed of text entry with a word prediction interface. *IEEE Transactions on Rehabilitation Engineering* 2, 3 (1994), 177–187. https://doi.org/10.1109/86.331567

[23] Per Ola Kristensson and Shumin Zhai. 2004. SHARK2: a large vocabulary shorthand writing system for pen-based computers. In *UIST '04*. ACM Press, New York, New York, USA, 43–52. https://doi.org/10.1145/1029632.1029640

[24] John D. Lee and Katrina A. See. 2004. Trust in automation: Designing for appropriate reliance. *Human factors* 46, 1 (2004), 50–80.

[25] I Scott MacKenzie and R William Soukoreff. 2003. Phrase Sets for Evaluating Text Entry Techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems (CHI EA '03)*. ACM, New York, NY, USA, 754–755. https://doi.org/10.1145/765891.765971

[26] Christopher D Manning, Christopher D Manning, and Hinrich Schütze. 1999. *Foundations of statistical natural language processing.* MIT press.

[27] Kseniia Palin, Anna Maria Feit, Sunjun Kim, Per Ola Kristensson, and Antti Oulasvirta. 2019. How Do People Type on Mobile Devices? Observations from a Study with 37,000 Volunteers. In *Proceedings of the 21st International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI '19)*. Association for Computing Machinery, New York, NY, USA. https://doi.org/10.1145/3338286.3340120

[28] Raja Parasuraman and Victor Riley. 1997. Humans and Automation: Use, Misuse, Disuse, Abuse. *Human Factors* 39, 2 (1997), 230–253. https://doi.org/10.1518/001872097778543886

[29] Raja Parasuraman, Thomas B Sheridan, and Christopher D Wickens. 2000. A model for types and levels of human interaction with automation. *IEEE Transactions on systems, man, and cybernetics-Part A: Systems and Humans* 30, 3 (2000), 286–297.

[30] Philip Quinn and Andy Cockburn. 2016. When Bad Feels Good: Assistance Failures and Interface Preferences. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA,

4005–4010. https://doi.org/10.1145/2858036.2858074

[31] Philip Quinn and Shumin Zhai. 2016. A Cost-Benefit Study of Text Entry Suggestion Interaction. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems (CHI '16)*. ACM, New York, NY, USA, 83–88. https://doi.org/10.1145/2858036.2858305

[32] Quentin Roy, Futian Zhang, and Daniel Vogel. 2019. Automation Accuracy Is Good, but High Controllability May Be Better. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems - CHI '19*. ACM Press, New York, New York, USA, 1–8. https://doi.org/10.1145/3290605.3300750

[33] Sherry Ruan, Jacob O. Wobbrock, Kenny Liou, Andrew Ng, and James A. Landay. 2018. Comparing Speech and Keyboard Text Entry for Short Messages in Two Languages on Touchscreen Phones. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* 1, 4, Article 159 (Jan. 2018), 23 pages. https://doi.org/10.1145/3161187

[34] Robert L Solso and Joseph F King. 1976. Frequency and versatility of letters in the English language. *Behavior Research Methods & Instrumentation* 8, 3 (1976), 283–286.

[35] Andrew Swiffin, John Arnott, J. Adrian Pickering, and Alan Newell. 1987. Adaptive and predictive techniques in a communication prosthesis. *Augmentative and Alternative Communication* 3, 4 (1987), 181–191. https://doi.org/10.1080/07434618712331274499 arXiv:https://doi.org/10.1080/07434618712331274499

[36] Keith Trnka, John McCaw, Debra Yarrington, Kathleen F McCoy, and Christopher Pennington. 2008. Word prediction and communication rate in AAC. In *Proceedings of the IASTED International Conference on Telehealth/Assistive Technologies (Telehealth/AT'08)*. ACTA Press Anaheim, CA, USA Baltimore, MD, 19–24.

[37] Keith Trnka, John McCaw, Debra Yarrington, Kathleen F McCoy, and Christopher Pennington. 2009. User Interaction with Word Prediction: The Effects of Prediction Quality. *ACM Trans. Access. Comput.* 1, 3 (feb 2009), 17:1–17:34. https://doi.org/10.1145/1497302.1497307

[38] Horabail Venkatagiri. 1994. Effect of window size on rate of communication in a lexical prediction AAC system. *Augmentative and Alternative Communication* 10, 2 (1994), 105–112. https://doi.org/10.1080/07434619412331276810 arXiv:https://doi.org/10.1080/07434619412331276810

[39] Daniel Vogel and Patrick Baudisch. 2007. Shift: A Technique for Operating Pen-Based Interfaces Using Touch. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI '07)*. Association for Computing Machinery, New York, NY, USA, 657–666. https://doi.org/10.1145/1240624.1240727

[40] Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain Text Entry on Mobile Devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. ACM, New York, NY, USA, 2307–2316. https://doi.org/10.1145/2556288.2557412

[41] Jacob O. Wobbrock, Leah Findlater, Darren Gergle, and James J. Higgins. 2011. The Aligned Rank Transform for Nonparametric Factorial Analyses Using Only Anova Procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*. ACM, New York, NY, USA, 143–146. https://doi.org/10.1145/1978942.1978963

[42] Jacob O Wobbrock and Brad A Myers. 2006. From Letters to Words: Efficient Stroke-based Word Completion for Trackball Text Entry. In *Proceedings of the 8th International ACM SIGACCESS Conference on Computers and Accessibility (Assets '06)*. ACM, New York, NY, USA, 2–9. https://doi.org/10.1145/1168987.1168990

[43] Hisao Yamada. 1980. *A historical study of typewriters and typing methods, from the position of planning Japanese parallels*. Journal of Information Processing.

[44] Ming Yin, Jennifer Wortman Vaughan, and Hanna Wallach. 2019. Understanding the Effect of Accuracy on Trust in Machine Learning Models. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19)*. ACM, New York, NY, USA, 279:1–279:12. https://doi.org/10.1145/3290605.3300509