Typing Behavior is About More than Speed: Users' Strategies for Choosing Word Suggestions Despite Slower Typing Rates

FLORIAN LEHMANN, Department of Computer Science, University of Bayreuth, Germany
ITTO KORNECKI, ETH Zurich, Switzerland
DANIEL BUSCHEK, Department of Computer Science, University of Bayreuth, Germany
ANNA MARIA FEIT, Saarland University, Saarland Informatics Campus, Germany

Mobile word suggestions can slow down typing, yet are still widely used. To investigate the apparent benefits beyond speed, we analyzed typing behavior of 15,162 users of mobile devices. Controlling for natural typing speed (a confounding factor not considered by prior work), we statistically show that slower typists use suggestions more often but are slowed down by doing so. To better understand how these typists leverage suggestions – if not to improve their speed – we extract eight usage strategies, including completion, correction, and next-word prediction. We find that word characteristics, such as length or frequency, along with the strategy, are predictive of whether a user will select a suggestion. We show how to operationalize our findings by building and evaluating a predictive model of suggestion selection. Such a model could be used to augment existing suggestion algorithms to consider people's strategic use of word predictions beyond speed and keystroke savings.

CCS Concepts: • **Human-centered computing** → **Empirical studies in HCI**; **Text input**; **Touch screens**.

Additional Key Words and Phrases: Text entry, word prediction, word suggestion, intelligent text entry methods, mobile text entry, typing

ACM Reference Format:
Florian Lehmann, Itto Kornecki, Daniel Buschek, and Anna Maria Feit. 2023. Typing Behavior is About More than Speed: Users' Strategies for Choosing Word Suggestions Despite Slower Typing Rates. *Proc. ACM Hum.-Comput. Interact.* 7, MHCI, Article 229 (September 2023), 26 pages. https://doi.org/10.1145/3604276

1 INTRODUCTION

Word suggestions are a widely used feature when typing on mobile devices such as smartphones. As the user types letters, the virtual keyboard suggests words that are predicted to match the intended input. Likely candidates are presented above the keyboard where the user can select them, as shown in Figure 1. This intelligent text entry (ITE) feature originates from augmentative and alternative communication (AAC) methods that support people with cognitive or physical impairments. It was developed to reduce the effort of communicating via text input. Instead of entering each letter of a word manually – which can take many seconds for AAC users – the complete word can be selected from the suggestion list, saving 30% to 50% of input actions [12, 19] and resulting in faster communication rates [13, 37].

However, for mobile users without any impairment, the use of word suggestions has been related to slower typing performance, despite keystroke savings. It comes at a cognitive cost for shifting

Authors' addresses: Florian Lehmann, florian.lehmann@uni-bayreuth.de, Department of Computer Science, University of Bayreuth, Bayreuth, Germany; Itto Kornecki, ittokor@gmail.com, ETH Zurich, Zurich, Switzerland; Daniel Buschek, daniel.buschek@uni-bayreuth.de, Department of Computer Science, University of Bayreuth, Bayreuth, Germany; Anna Maria Feit, feit@cs.uni-saarland.de, Saarland University, Saarland Informatics Campus, Saarbrücken, Germany.

2573-0142/2023/9-ART229
https://doi.org/10.1145/3604276

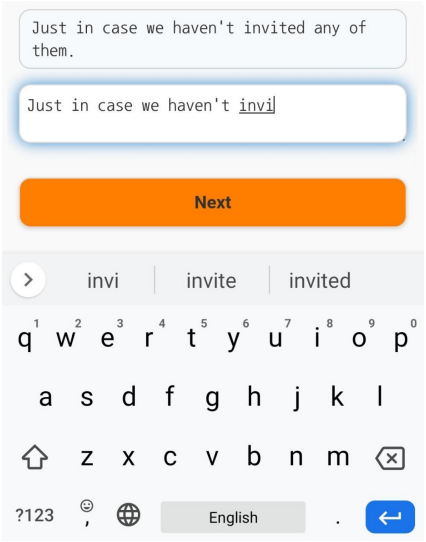


Fig. 1. Typical word suggestion interface on mobile devices. Screenshot from the typing test that provided the data for this work. The suggestion bar offers the currently typed string, and two word completions.

attention from pressing keys to reviewing and choosing words from the suggested list. This cost may outweigh the performance benefit [22, 25, 33]. Particularly on mobile devices, where users take only few hundred milliseconds to press keys [32], recent studies have observed that users of word suggestions were generally slower despite significant keystroke savings [3, 32, 33].

Nevertheless, word suggestions are a popular feature of mobile keyboards and studies reported that users perceive them to require less effort and are more satisfied with their typing experience [21, 33] despite the slower performance. Thus, we observe a discrepancy between text entry research and word suggestion usage in practice. While research on word suggestions is motivated by increasing typing speed, end-users utilize suggestions to improve their overall typing experience. How they integrate word suggestions into their typing process and the factors contributing to their typing experience are not well understood.

To address this, the goal of this paper is to better understand the way in which people make use of word suggestions during regular typing on mobile devices (not considering gesture typing), including – but going beyond – their effect on typing performance. Therefore, we analyze an existing dataset published by Palin et al. [32] which contains typing data from several thousand volunteers who participated in an online typing test. In contrast to prior work [3, 33], people were typing on their own devices and thus interacting with their familiar word suggestion algorithms. They were not paid to participate in the study but were intrinsically motivated coming to the webpage to learn about their mobile typing speed. This ensures a high external validity of the dataset and our findings here.

For the first time, we statistically show that typing speed and word suggestion usage mutually affect each other: Word suggestions are more commonly used by slower typists and at the same time suggestions slow down the typing process. To this end, we compute four different speed metrics that allow us to separate the motor processes and cognitive processes involved in typing with word suggestions. This enables us to assess the cognitive costs of suggestions and their effect on users' typing performance, disentangling the ambiguous correlations and average trends found by prior work [3, 32].

But how do people make use of word suggestions if not to improve their speed? We show evidence that users employ specific strategies when using suggestions, and that the strategy used depends heavily on the word which the user is typing. We identify eight strategies, including word completion and correction as the most prevalent ones. These significantly differ in how users type before making a selection and are particularly used for longer and infrequent words. In other cases, suggestions are strategically used to capitalize a word or to add an apostrophe for contraction words (e.g. *can't, don't*). In several cases, people also select incorrect predictions which are similar to their intended word, and then manually correct them to enter their intended word.

Our findings challenge the assumption often made by prior work, which is that users will select a word as long as it matches the one they are currently typing [12, 23, 25]. Instead, we show that typists strategically utilize suggestions for specific types of words. This indicates that suggestions are an integral part of a user's typing strategy and not simply a reaction to the keyboard displaying them. Crucially, this also implies that users plan ahead whether and how to use suggestions *before* these are displayed by the keyboard. This holds important implications for the design of ITE methods. To better support users, word suggestions should not only match the word a user is currently typing, but also consider how typists will want to make use of them. For example, instead of favoring frequent words which are more likely to be typed by the user, a suggestion algorithm could prioritize displaying less frequent words for which users are more likely to need a suggestion for (e.g. because they are unsure about spelling). We show how this could be achieved by deriving a selection model from our empirical findings that predicts whether a user will select a displayed suggestion based on the user's typing behavior and the characteristics of the word. We discuss how such a model could be used to augment existing algorithms to provide suggestions that are not just accurate, but also likely to be used by the typist. We make the code for our analysis and the selection model publicly available.

2 RELATED WORK

In the following, we briefly describe how word suggestions are computed on mobile keyboards. Although the underlying algorithms are not the focus of our work, they inform our analysis and findings. We then discuss prior work on the benefits and drawbacks of using word suggestions and show how such empirical findings have been used by researchers to inform the design of ITE systems.

2.1 Word suggestion algorithms

The foundation of most intelligent text entry methods is a statistical language model which predicts the probability of the next word or phrase given a series of words or characters the user has previously entered (see e.g. [30] for an overview). A typical approach uses n-gram dictionaries that contain the frequency with which various combinations of words occur in a text corpus. A word completion algorithm determines the most frequent words that start with the characters entered by the user so far. Modern keyboards on mobile devices combine such an algorithm with an error correction mechanism, a so-called decoder that infers a user's intended word from their noisy input actions (e.g. pressing small keys with a large thumb, resulting in entering the wrong characters). Therefore, a completion algorithm combines a language model with a touch model that estimates the probability that a series of observed touch points corresponds to a specific word [1, 16, 30, 38]. In recent years, much research has been dedicated to improving these language and touch models to accurately predict the word that the user intends to type next (e.g. [6, 15, 34, 40, 41]). In particular, recent advances in machine learning have also led to neural network-based language models for mobile word suggestion [18, 42]. While advanced models are able to capture the statistical relationships between a larger number of words to make more reliable predictions about the next

word a user wants to type, they all share the same assumption: if the correct word is suggested to the user, the user will choose it rather than typing the word manually. Little research has explored the tendency of typists to use these suggestions and the factors that influence these choices beyond the accuracy of the algorithm.

We show that even if a word is correctly suggested to the user, a user may still choose to type it manually. We explore how typists choose word suggestions as part of their typing process. We show that users strategically decide to use suggestions for specific types of words and, at times, they even choose wrong suggestions that are similar to their intended word. We propose a simple selection model which augments existing suggestion algorithms and takes into account user preferences.

2.2 Keystroke savings versus performance improvements

Today's word suggestions on mobile devices are largely inspired by early work in the 1980s and 1990s (e.g. [19, 22, 23, 29, 35]). These works aimed to improve augmented and alternative communication (AAC) devices which support people with cognitive or motor impairments to communicate via generated speech or written text.

Several studies have empirically and theoretically shown that word suggestion algorithms can yield large keystroke savings of at least 30 −50% [12, 14, 19, 33]. Given the amount of time that AAC users often require to type individual keys, it is reasonable to assume that such keystroke savings translate to faster typing speeds. Several studies have empirically confirmed this assumption [4, 29, 36]. At the same time, researchers recognized that the cognitive cost of using word suggestions might counteract the performance benefit from keystroke savings [22]. Choosing a word suggestion requires interrupting an often automated motor task, shifting attention to the suggested words, and deciding whether to choose one of them. There is a risk that the desired word might not be suggested, and switching attention from manual typing to the suggestion list might not pay off.

2.2.1 Keystroke savings do not translate to increased typing speed on mobile devices With the advent of mobile phones and touch interfaces, word suggestions became an integral part of standard virtual keyboards. However, some recent studies have observed that users type significantly slower when using word suggestions. Quinn and Zhai [33] found that using word suggestions reduced the typing speed from an average of 3.09 characters per second to 2.66 characters per second in a lab study and attributed this to the cognitive cost of evaluating and selecting the suggestions. The same was found by Arnold et al. [5] in the context of phrase suggestions, noting that the cost of accepting suggestions counteracts any speed benefit it may provide [5]. A large-scale online study of mobile typing by Palin et al. analyzed the effect of various ITE methods on typing speed. They found a negative correlation between users' typing speed and their number of chosen word suggestions. Similarly, in a crowd-sourcing study, Alharbi et al. [3] found that words chosen from the suggestion list were typed on average 2.09s slower than manually typed words. However, these studies did not control for the manual typing speed of participants, leaving open the question as to whether typing with word suggestions is indeed slower or whether the effect is caused by slower typists using word suggestions more often.

Overall, we can say that the performance benefits of word suggestion highly depend on the characteristics of the text entry method, the accuracy of the prediction algorithm, and the user's strategy for distributing their attention. We perform a detailed statistical analysis to disentangle the motor processes of typing from the cognitive processes involved in choosing suggestions and quantify the negative effect each chosen suggestion has on a user's typing performance, independent of their motor speed.

2.2.2 Typists objectives for using word suggestions Despite the apparent reduction in speed, word suggestions still benefit mobile users' by saving keystrokes [32, 33] and improving their subjective

typing experience. For example, Quinn and Zhai found that typists using word prediction perceived the typing experience to require less physical effort, and less effort overall, compared to typing normally [33]. Prior to this work, Kamvar and Baluja found a similar reduction in perceived workload when surveying typists using word prediction systems on 9-key mobile phones [21].

Beyond these few observations, prior studies have not considered the potential objectives of people to choose word suggestions despite their negative impact on performance. This paper takes a first step to explore users' strategic use of word suggestions that are motivated by other aspects than speed. We demonstrate that they are more likely to choose suggestions for specific types of words, such as capitalized words or contractions, purposefully omitting for example an apostrophe and relying on the algorithm to offer this suggestion.

2.3 Informing the design of ITE methods with user simulations

Simulation models have long been used in HCI to inform the design of interactive systems, a famous example being Card et al.'s Model Human Processor proposed in the 1980s [10]. Although their use is not as widespread as in other disciplines, the rise of advanced machine learning models and large-scale data also yields new computational approaches for HCI research [28]. In particular for text entry systems, researchers have long tried to model and simulate user performance for example to optimize keyboards (see e.g. [11] for an overview) but also to inform the design of ITE methods [12], in particular word suggestion algorithms.

For typing with word suggestions, Koester and Levine [23] and more recently Kristensson and Müller [25] developed simulation models that predict a user's typing performance as a result of their cognitive and motor characteristics, their interaction strategy, and the accuracy of the word suggestion algorithm. Such models can be used to assess the impact of the accuracy of the algorithm on overall performance or could evaluate design choices, such as varying the number of displayed suggestions as studied by [2, 7]) without having to run a user study.

These simulation models are based on empirical observations. Most studies were interested in understanding the effect the language model has on user performance. As such, they controlled and systematically varied the accuracy of the suggestion algorithm and observed user's performance [3, 33]. However, empirical studies that explore the interaction strategy of users when they select words from the suggestion list are missing. Thus, most simulation models assumed that users would choose a suggestion as soon as it is displayed [12] or formulated simple strategies where users check the suggestion list after a fixed number of characters [23, 25]. This reflects the general assumption that users will choose any accurate word suggestion. Recently, Oulasvirta, Jokinen, and Howes have proposed computational rationality as a theory for predicting interaction strategies as a result of the user adapting their behavior to an interface, their own cognitive and motor constraints, and their objectives and preferences [31]. While advances have been made to model the motor and cognitive characteristics of users typing on mobile keyboards [20], they do not include the use of word suggestions for which the objectives that guide a user's actions are unclear yet, as discussed above.

Thus, our research analyzes empirical data of word suggestion usage to identify common strategies which users employ when using word suggestions while typing. This will enable us to better understand people's objectives when typing, inform simulation models, and ultimately allow for better design of ITE methods.

3 DATASET

We build on an existing dataset by Palin et al. [32]. We decided on this dataset because of its large scale and its high external validity. We next describe the original data and our pre-processing steps to obtain a large-scale accurate dataset of word suggestion usage on mobile devices.

Actual keystrokes	Registered keyboard event
	'T', 'Th', 'The'
'T', 'h', 'e'	'T', 'h', 'e', 'The'
	'undefined', 'undefined', 'undefined'

Table 1. Types of faulty backend behavior which we filtered out to obtain a valid dataset of word predictions.

3.1 Original data: the 37K dataset

We obtained the dataset by Palin et al. [32] from their project page[1]. The authors collected mobile typing data from 37,370 volunteers who participated in an online typing test. People came to a commercial website to learn about their typing speed and voluntarily chose the typing experiment offered by the researchers among a list of standard tests provided by the webpage. In the typing test, participants transcribed 15 sentences randomly selected from the Enron mobile email corpus [39] and the English Gigaword Newswire corpus. They used their own keyboard and keystroke data was logged via the browser. The online typing test offers more control than an in-the-wild study and yields a larger sample of participants and mobile devices than a lab-based study. The intrinsic motivation of people to learn about their speed and the use of their own devices without any restrictions of modifications as done in other studies (e.g. [3] ensures realistic typing behavior and thus a high external validity of the collected data. The large scale of the dataset allows us to perform powerful statistical analyses. On the other side, people were self-selected, resulting in a dataset of rather young participants with a larger percentage of women. See the original paper for more details on the data collection [32].

3.2 Filtering

The browser-based logging comes with several limitations, such as restricted access to keypress information on some devices. Given the large size of the original dataset (over 37,000 users) we decided to rigorously exclude parts of the data to obtain accurate logs of word suggestion usage with a homogeneous quality.

We took three filtering steps:

(1) *Exclude users with invalid event data:* in particular for some Android devices, keystrokes were logged by the browser in unexpected ways. Table 1 shows several examples. We exclude those users to ensure consistent data logs, needed to reliably detect the selection of a suggested word (see below).
(2) *Exclude non-native speakers:* since the transcription task was in English, we exclude self-reported non-native speakers to increase the homogeneity and realism of the typing behavior.
(3) *Exclude swipe users:* gesture typing (or swiping) is a common technique where the user continuously draws from one letter to another to enter a word. Word suggestions are an inherent feature of this input method needed to disambiguate between similar gestures and to fix recognition errors. In contrast, manually entered text does not rely on the availability of suggestions. Thus, we expect swipe users to use suggestions differently than manual typists and remove self-reported swipe users and those for which we detected the use of gestures for more than 10% of words.

[1] https://userinterfaces.aalto.fi/typing37k/

3.3 After filtering: input data from 15,162 participants

After filtering, we arrived at data from 15,162 participants These participants reported their gender as female (10,565; 69.68%), male (3,762; 24.81%) or did not disclose (835; 5.51%). Their age ranged from 6 to 61 years, with a median of 21 years. This is similar to the original dataset [32]. In terms of operating systems, a large part of Android users was excluded due to invalid logging behavior yielding in 91.67% of participants using iOS.

3.4 ITE event detection

The web-based data collection is limited in what information it receives about each keystroke, thus the dataset does not include information about whether the user pressed a single key, chose a word suggestion or was automatically corrected by an algorithm. The original work solved this by classifying events based on the observable changes in the text input field. We decided to derive our own recognition scheme since our conservative filtering approach described above ensures that keystroke events have consistent metadata.

We label each keystroke as one of three events: user types normally (*none*), autocorrection is triggered (*autocorrect*), or a word is selected from the suggestion bar (*suggestion*). We do not consider the gesture class recognized by Palin et al. [32] since we previously filtered out all self-reported swipe users. We distinguish these events in three steps:

(1) *Based on input length:* Manually entered characters are recorded in the backend as a single-character string. Thus, all events with an input length of one are classified as *none*. All others are ITE events.
(2) *Based on edit distance:* We compute the Levenshtein distance between the current and previous state of the input field. Autocorrections have an edit distance of at least one character. Thus, all ITE events with an edit distance of 0 are labeled as *suggestion*, which might occur if a user selected a suggestion for a word they already typed out.
(3) *Based on inter-key interval (IKI)*: the remaining ITE events are distinguished based on the time between the current and previous input event. In the case of *autocorrection*, the event happens automatically after pressing the space key, thus yielding a small IKI value. Suggestion selection requires more time since it is considered to be cognitively more demanding [23]. After assessing the IKI distributions of the ITE events in our dataset, we define a double threshold to distinguish ITE events while minimizing false positives: ITE events with an IKI below 400ms, are classified as *autocorrection*, and those above 500ms are labeled as *suggestion*.

4 TYPING PERFORMANCE WITH WORD PREDICTIONS

In this part, we want to first explore how typing speed and word suggestion usage impact each other. In their original analysis of the dataset, Palin et al. observed a negative correlation between the number of suggestions used and participants' typing performance. However, this might depend on several interacting factors, including the accuracy of the algorithm, the users' cognitive abilities, and their motor skills [25]. To disentangle this correlation, we define four speed metrics that separate the motor and cognitive processes involved in typing. We then correlate these with the suggestion usage of typists. While it might not be sufficient to establish a causal relationship between typing speed and suggestion usage from our observational data, it gives us a more differentiated view on people's typing behavior on their own devices without restricting or affecting their normal interaction.

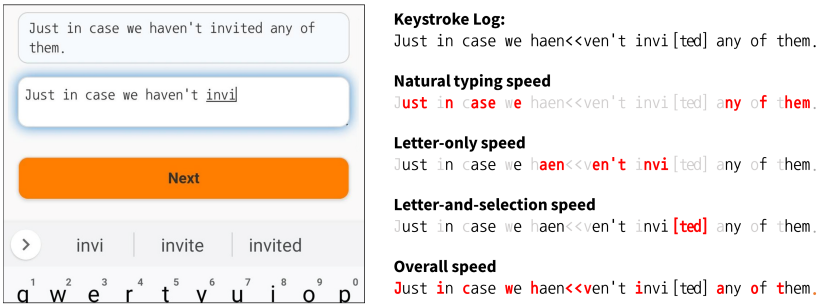


Fig. 2. We compute four different speed metrics to differentiate between motor expertise and the cognitive processes involved in typing, such as checking for and selecting suggestions. The first line shows an example keystroke log where a user corrected two mistakes (indicated by <) and entered the letters *ted* with a single keystroke (indicated by [ ]) by choosing a suggestion. The different metrics build on each other and are computed based on the highlighted keystrokes (in red and black). We use natural typing speed as a control variable when analyzing the impact of word suggestions on people's typing performance.

4.1 Performance metrics

We measured typing speed in four different ways, allowing us to assess the effect of choosing word suggestions on a user's overall typing performance. The intuition behind them is described in the following. Their computation is specified in Table 4 in the Appendix A.1 and exemplified in Figure 2. Each of them is measured in characters per second by averaging across the inter-key intervals of all keystrokes included in the computation. In addition, we measured keystroke savings as a standard performance metric used in the literature.

4.1.1 Natural speed We define the natural speed as a performance measure of the motor processes involved in pressing the keys on the keyboard. We thus *exclude* the use of word suggestions (or other ITE features) and only analyze keystrokes of lowercase letters that are preceded by lowercase letters, in common words shorter than five characters. In this way, we ensure to capture typing behavior that reflects a participant's motor skill, and minimize delays due to cognitive events such as checking the source text [24] or checking for suitable suggestions.

4.1.2 Letter-only speed The letter-only speed is measured similarly to the natural typing speed, except no restrictions are made regarding the typed word (e.g. including longer words). Thus, in addition to the motor speed, this measure includes the cognitive process of considering the use of suggestions and reviewing the word suggestion list, but it does not include the keyboard event where a suggestion is selected.

4.1.3 Letter-and-selection speed In addition to the previous keystrokes, this measure includes keyboard events that represent the selection of a suggestion. Thus, this speed involves the process of both reviewing and then selecting a word from the suggestion list.

4.1.4 Overall speed The overall speed involves all keystrokes, including non-letter ones, such as spaces, backspaces, corrections, capitalization, etc. It is equivalent to character-per-second speed evaluations made in the literature [27].

4.1.5 Keystroke savings In addition to typing speeds, we computed *keystroke savings* (KS) achieved by using word suggestions: We computed the keystrokes per character (KPC) by dividing the total

	Natural Speed	Letter-only Speed	Letter-and-selection Speed	Overall Speed
Non-suggestion users	6.13 (1.81)	5.6 (1.69)	5.6 (1.68)	4.19 (1.49)
Suggestion users	5.22 (1.43)	4.67 (1.28)	4.08 (1.21)	3.19 (0.86)
Difference[1]	–	-0.12*** (0.01)	-0.74*** (0.01)	-0.43*** (0.01)

[1] controlled for natural speed *** p<0.001

Table 2. Mean and standard deviation for speeds of suggestion and non-suggestion users. Difference was controlled for natural speed.

number of keystrokes by the total number of characters in the final text. We used the formula described by Garay-Vitoria and Abascal [13] to convert KPC to keystroke savings: $KS = 1 - \frac{1}{KPC}$

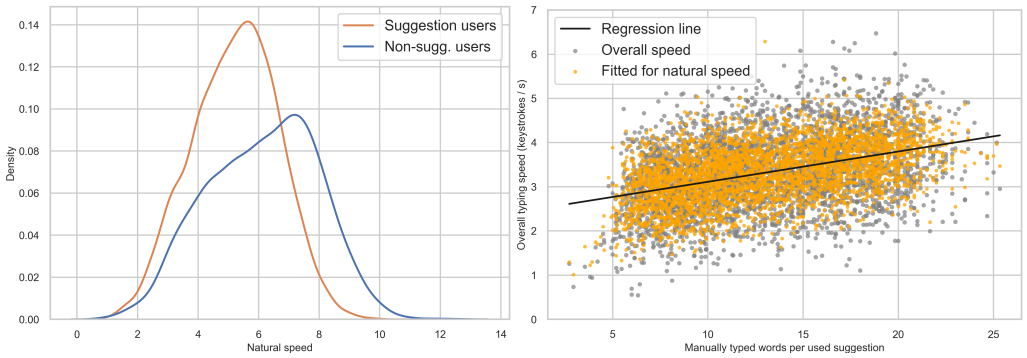
4.2 Suggestion- versus non-suggestion users

For analyzing the impact of suggestion usage on typing speed, we further filtered our dataset to only include people who never or frequently used word suggestions, yielding two user groups: (1) 3,926 *non-suggestion users* – who entered all characters manually and never chose a suggestion; and (2) 4,396 *suggestion users* – who chose suggestions more than five times during the 15 transcribed sentences. We first compared participants' natural typing speed, thus analyzing the differences in motor skill between *suggestion users* and *non-suggestion users*. We then compared the letter-only speed, letter-and-selection speed, and overall speed between the two user groups. All results are given in Table 2. The bottom row shows the performance differences between these two groups in keystrokes per second when controlling for their natural typing speed. Therefore, we performed ordinary least squares (OLS) analyses on each speed metric where the independent variables were users' natural speed and whether they used suggestions (a binary label). The results thus indicate how the usage of predictions affect typing speed at different levels of interaction, as described in the following.

Natural speed: The distributions of natural typing speeds for *non-suggestion users* and *suggestion users* as shown in Figure 3a descriptively show that there are more slower typists who make use of word suggestions. *Non-suggestion users'* natural speed was on average 6.10 keystrokes per second (SD = 1.81) versus 5.24 (SD = 1.43) for *suggestion users*.

Letter-only speed: Even when excluding the time taken to select a word from the suggestion list, *suggestion users* still type slower: They have a letter-only speed of 4.67 keystrokes per second (SD = 1.28), whereas *non-suggestion users* have 5.6 (SD = 1.69). Controlled for the difference in natural speed, *suggestion users* type 0.12 keystrokes per second slower (SD = 0.01). This could be attributed to the cognitive processes of checking the word suggestion list in cases where no suggestion is selected.

Letter-and-selection speed: Including the time taken to select words from the suggestion list, the difference in speed between *suggestion users* and *non-suggestion users* increases: The average letter-and-selection speed of *suggestion users* is 4.08 keystrokes per second (SD = 1.2) and 5.6 (SD = 1.68) of *non-suggestion users*. Controlled for difference in natural speed, *suggestion users* type 0.74 keystrokes per second slower (SD = 0.01). A large part of this difference could thus be attributed to the time it takes to decide for and select a suggestion from the word prediction list.



(a) (b)

Fig. 3. The effect of using word suggestion on typing speed. (a) Distributions of natural speed for *suggestion users* and *non-suggestion users*. (b) Effect of manually typed words per selected suggestion on typing speed: the more words are typed without suggestions, the faster the overall speed. Grey dots visualize the overall speed. Orange dots visualize the fitted values for overall speed, after controlling for natural speed.

Overall speed: Analyzing all the keystrokes, including also spaces, error corrections, and first letters of a word, show a smaller difference in typing speed: *Suggestion users* type 3.19 keystrokes per second (SD = 0.86) and *non-suggestion users* 4.19 (SD = 1.49). Controlled for natural speed, the difference is 0.43 keystrokes per second (SD = 0.01). Although there is a difference in typing speed that cannot be explained by the difference in natural speed, it is not as large as suggested by the difference in letter-and-selection speed. This could indicate word suggestions might be faster in certain cases, for example when correcting errors.

4.3 The impact of suggestion usage on typing performance

We further analyzed the effect of the rate at which suggestions were used. For this, we performed a multivariate regression using OLS on the data of *suggestion users*. We correlate the overall speed of users with their use of suggestions while controlling for their natural speed (i.e. natural speed and suggestion rate as independent variables, and overall speed as dependent variable). To ensure normality of the data, we compute suggestion rate as the number of *manually* typed words per used suggestion, e.g. a user might choose a suggestion every 10 words. The final analysis does not exclude outliers since it did not change the results. We refer to our analysis scripts for further assessments on the linearity and homoscedasticity of the data.

The results are visualized in Figure 3b and show that the more words are typed without a suggestion, the faster the user types: For every additional manually typed word per suggested word, the overall speed increased by 0.029 keystrokes per second (SD = 0.002, $p < 0.001$). This analysis takes into account the natural typing speed of users, that means, for two participants with the same natural speed, a user selecting a suggestion once in ten words will type 0.42 keystrokes per second slower than a user selecting a suggestion only once in 20 words. Comparing the gray and orange dots in Figure 3b, we see that controlling for participants' natural typing speed reduces parts of the variation in the data and shows a clear linear relationship between suggestion usage and overall speed after controlling for users' motor skills. However, we also see note that there is considerable variance left which cannot be explained by natural typing speed or suggestion usage.

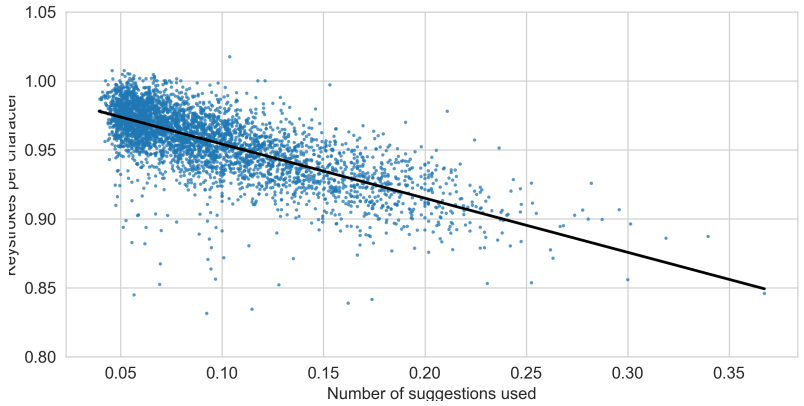


Fig. 4. Incremental keystrokes savings of suggestion usage.

4.4 Keystroke savings

We examined the effect of using suggestions on KPC and keystroke savings using an OLS analysis where the rate of suggestion usage (in terms of percentage of words for which suggestions were selected) was the independent variable, and the keystrokes per character were the dependent variable. Similar to above, we did not exclude outliers.

The result is visualized in Figure 4 (r^2 = 0.484). The results show that increased usage of suggestions leads to a decrease in the number of keystrokes per character (KPC). Our analysis shows for every 1% increase in the usage rate of suggestions, the KPC decreases by 0.004 (SD = 6E-5, p < 0.001). Practically speaking, a user selecting a suggestion once every 10 words will have keystroke savings of 4%, while a user selecting a suggestion once every five words will have keystroke savings of 8%.

4.5 Summary

By controlling for natural typing speed, we have found that there is a relationship between typing speed and the use of word suggestions: Typists who type more slowly tend to rely more often on suggestions. Conversely, the use of suggestions also slows down the typing speed. We see a difference between *suggestion users* and *non-suggestions users* even if we exclude the time it takes to choose a suggestion from the suggestion list. Thus, the keystroke savings achieved through the use of word suggestions do not translate to performance improvements in mobile typing. We further discuss these results in the Discussion section.

5 USERS' STRATEGIES FOR SELECTING WORD SUGGESTIONS

To better understand how and why people use suggestions, if not for speed, we looked for trends in which words users tend to select a suggestion vs typing them manually. We refer to these trends as *strategies* throughout this paper.

Without prior ground truth and a vast feature space associated with the words and typing behavior, it is extremely difficult to identify strategies using unsupervised methods (e.g. clustering). Therefore, we employed a semi-qualitative approach: In the first step, we combined insights from manual inspection of the typing data with our experience and domain knowledge about word suggestion use. Based on these insights, we formed hypotheses about potential strategies. In the second step, we implemented a rule-based classifier to identify all occurrences of these strategies in the dataset. We iterated both steps to narrow down a list of frequent strategies.

Next, we first describe the identified strategies, before taking a closer look at the two most common ones.

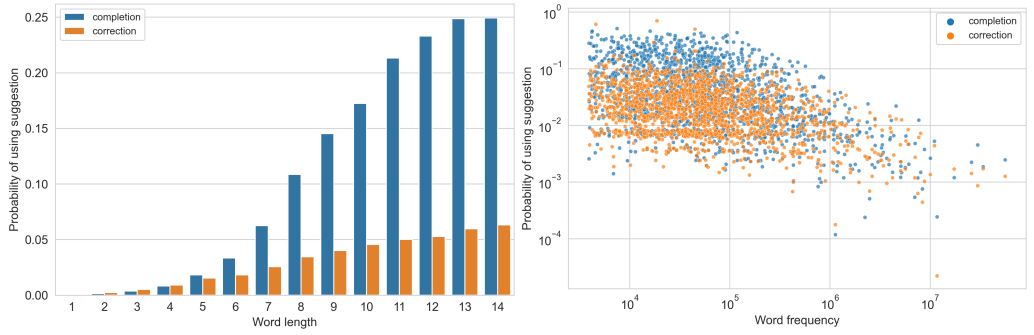
5.1 Selection strategies

Table 3 provides an overview of the eight strategies we found. Their associated rules for labeling them in the data can be found in Table 5 in the Appendix. In total, the eight strategies account for 98.5% of all suggestion usage. The strategy *completion* is used most often with 60.6%, followed by *correction* with 28.3%. Together, these two account for almost 90% of all suggestion usage. All other strategies occur with less than 5% individually.

Note that the frequency of a strategy depends heavily on the words it is applied on. For example, the *contraction* strategy is only used for 2.6% of all suggestion usage but accounts for 48.9% of suggestion usage of contraction words. Similarly, *capitalization* accounts for 23.6% for suggestions of capitalized words (vs. 2.5% overall).

Strategy	Description	Frequency
Completion	The user types the beginning of the word and then selects a word from the suggestion list to complete it. E.g. user types "wat" and selects "watch".	60.60%
Correction	The user types part or all of a word and then selects a suggestion to correct a mistake that was made. E.g. user types "abliity" and selects "ability".	28.30%
Prediction	The user selects a word from the suggestion list before typing anything.	3.70%
Contraction	The user types a contraction word fully without the apostrophe and then selects a word from the suggestion list to add it. E.g. user types "cant" and selects "can't".	2.60%
Capitalization	The user types a capitalized word fully in lower case, and then selects a word from the suggestion list to add capitalization. E.g. user types "bob" and selects "Bob".	2.50%
Insert space	The user fully types two or more words but forgets to add a space between them. The user then uses the suggestions to add a space between the words. E.g. User types "thereis" and selects "there is".	0.40%
Fixup	The user makes a mistake while typing a word. To correct the mistake, the user partially erases the word and selects a new word from the suggestion list. E.g. user types "press", then erases to get "pres", and then selects "president".	0.40%
Select-and-modify	A "meta-strategy": The user selects a suggestion (potentially with any other strategy) and modifies it. We found these types: Adding or removing characters at the end (e.g. select "apples", modify to "apple") and replacements at the end (e.g. "responsible" to "responsibility").	5.0%

Table 3. Description and frequency of the eight strategies for using word suggestions we found in our analysis.



(a) (b)

Fig. 5. Effect of word characteristics on the probability of using word suggestion. (a) Word length. (b) Word frequency.

5.2 A closer look at completion and correction strategies

As *completion* and *correction* are the most used strategies, we take a closer look at them here: In particular, we analyzed the effects of word length and frequency, and examine the related typing behavior.

5.2.1 Effect of word length Figure 5a descriptively shows the effect of word length (in characters) on these two strategies. Longer words have a higher probability of being used as part of the two strategies. We verify this by examining length vs frequency: For words with four characters, *correction* is only used for 0.90% of all words and *completion* for 0.82%. In contrast, for words with 14 characters, *correction* is used for 6.3% and completion for 25%. These results indicate that for both strategies, longer words are more frequently entered with the use of suggestions.

5.2.2 Effect of word frequency Figure 5b shows the effect of word frequency (as per [17]) on suggestion usage. The probability of using suggestions in the *completion* or *correction* strategy decreases for more frequently used words. For further comparison, we categorized words into common words (frequency > 100,000 in [17]) and uncommon ones (<100,000): For suggestions of common words, *completion* was used 1.8% and *correction* 0.42%. For uncommon words, *completion* and *correction* were used 5.5% and 2.7% of the time, respectively. These results indicate that getting support for entering uncommon words is a motivator for using word suggestions.

5.2.3 Typing speed before selecting suggestions We examined the *letters-only typing speed* (defined in Section 4.1) for the characters of a word that were typed before selecting a suggestion for that word. Figure 6 visualizes this *lead-up speed*. In summary, the lead-up speed for words without suggestion usage was 5.8 keystrokes per second (SD = 2.13). If the *completion* strategy was used for a word, that speed was 3.46 (SD = 1.70), and for the *correction* strategy, it was 4.63 (SD = 2.04). An ANOVA revealed a significant difference between the speeds of these strategies ($p < 0.001$). This indicates that the negative performance impact of using suggestions varies between strategies. This could be explained by the fact that the completion strategy might involve checking the word prediction list more frequently, whereas for correction it might only be checked at the end of a word, once a typing mistake was detected.

5.2.4 Edit distance We analyzed the edit distance between the manually typed word and selected suggested word. For *completion* (Figure 7a), this edit distance increases with word length. For *correction* (Figure 7b), the distance stays close to 1 regardless of word length. For a word of four

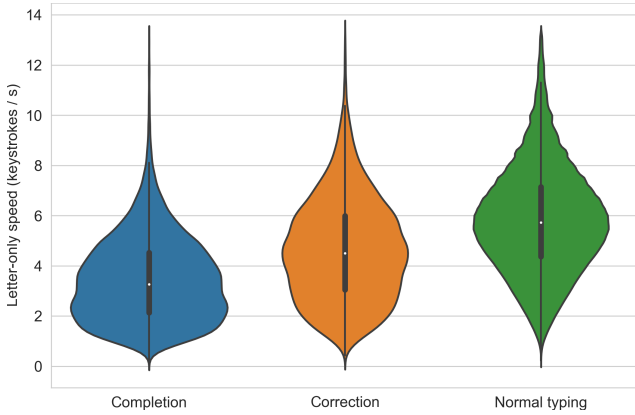
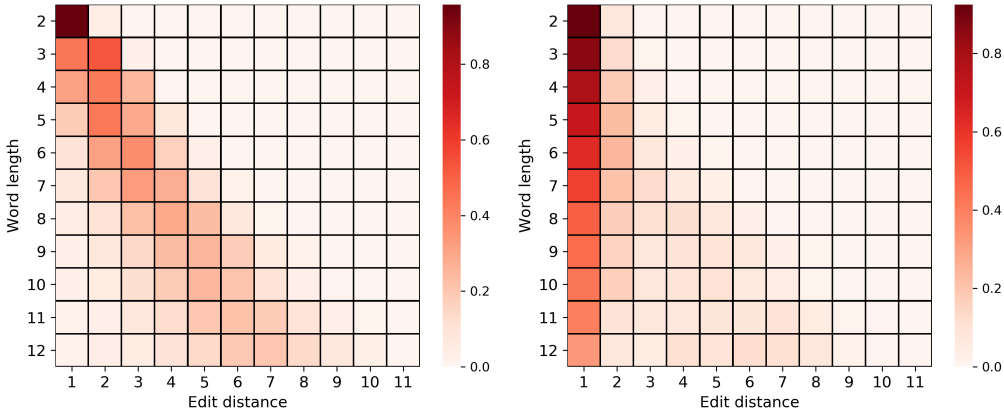


Fig. 6. Typing speed leading up to a selection for the strategies of *completion* and *correction*. For comparison, we show typing speed for words where suggestion is not used.



(a) Completion (b) Correction

Fig. 7. Edit distance of completion and correction for increasing word lengths, per strategy. Color indicates frequency.

characters, for example, *completion* and *correction* result in a median edit distance of two and one character(s), respectively. For a word of 10 characters, the median edit distance for *completion* is 5 characters and for *correction* 2 characters. These results further support the bigger picture: The strategies differ in various aspects. Concretely, in addition to word length, frequency, and speed, this also reflects in how much the suggestion changes/extends the typed (parts of a) word.

5.3 Summary

In summary, we identified eight strategies for how typists use word suggestions. The strategies *completion* and *correction* are used most often and users particularly prefer to use them for long and uncommon words.

Furthermore, they used suggestions strategically to modify words, such as capitalization or adding apostrophes for contractions (e.g. *can't* or *don't*), which was used for almost 50% of contraction

words. Another approach of strategically using suggestions happened in parallel to the main strategies: Users select a prediction close to the intended word to modify it afterwards.

6 APPLICATION: IMPROVING SUGGESTION ALGORITHMS

Our analysis shows that the characteristics of a word influence whether a user will select a corresponding word suggestion. Moreover, we saw that their typing behavior changes depending on the strategy the suggestion corresponds to (e.g. completion versus correction). This has important implications for the design of word suggestion algorithms, which are typically optimized to show accurate suggestions, not considering whether users will choose them.

In this section, we show how our empirical findings could be operationalized to improve existing word suggestion algorithms. For demonstrational purposes, we developed a logistic regression model that predicts the probability that a displayed (correct) suggestion would be selected by a user. In the following, we give a brief intuition on how this selection model works and how it could be combined with a standard suggestion algorithm. In the Appendix, we provide detailed information on the logistic regression model we built from our dataset, including the features' coefficients, their statistical significance, and an assessment of its performance.

For computing the probability that the user will select a suggested word, we introduce the selection model P_S. This model predicts a probability based on the input behavior of the user, the characteristics of the word, and the corresponding strategy. Mathematically, this can be formulated as:

$$P_S = P(selection|i, w, strategy) \tag{1}$$

Where i is the set of features representing the current input of the user, w are the characteristics of the suggested word (e.g. the length) and $strategy$ is the strategy that would be invoked by selecting the word from the suggestion list. Typing "wat" and selecting "watch" from the suggestion list, for example, implies a completion strategy.

6.1

The predictions of the selection model could be directly combined with the output of a suggestion algorithm to make accurate predictions that will also be chosen by users. This can be formulated as follows:

$$P = P_A \times P_S \tag{2}$$

Where P_A is the word probability estimated by an existing keyboard prediction algorithm (e.g. with language and touch models) and P_S is the probability estimated by the selection model, as described above.

With a practical example, we describe the process of how the model can be used to estimate the probability of selections and rerank candidates in the suggestion list. We suppose for this example, the user has the intent to write the phrase *I love musicals*, and already typed the beginning *I love m*.

(1) The user types the letter *u*. The current string is *I love mu*.
(2) The language model and the touch model suggest several words that could likely extend the sentence. We give three examples and their hypothetical probability score P_A: *my*=0.5, *music*=0.3, and *musicals*=0.08.
(3) For each suggestion, the strategy is inferred (see Appendix): *my* is a correction, *music* a completion, *musicals* a completion.

(4) For each candidate, the selection model computes P_S, which is the likelihood of the word being selected if suggested to the user. For example, *my*=0.01 (e.g. since the word is very short), *music*=0.04, and *musicals*=0.2 (e.g. since it is longer and infrequent).
(5) Probabilities of the language model are combined with the probability of the selection model by multiplication, yielding: *my*=0.005, *music*=0.012, *musicals*=0.016.
(6) In a typical interface (see Figure 1), the top two suggestions are then presented to the user ordered by descending score, together with their current input.
(7) Repeat steps 2 to 7, every time a user inserts a new character.

This process illustrates how the selection model could be used to influence the order of suggestions to not only focus on the accuracy of the suggestion but also consider whether the suggestion will be useful to the user.

7 DISCUSSION

Our results show that there are more slower typists who make use of word suggestions on mobile keyboards. At the same time, we find suggestion users typed slower than people that do not use suggestions. This difference was found to be significant even when controlling for participants' natural typing speed. These typists make use of suggestions willing to suffer potential speed penalties through their use of word suggestions. With our in-depth analysis of their suggestion usage, we identify specific strategies that describe how typists make use of word suggestions and integrate them into their typing process. Our empirical findings could inform a predictive model of suggestion usage and we showed how such a model could be used to refine existing suggestion algorithms. In the following, we discuss our findings on decreased typing performance, the identified strategies, and our selection model.

7.1 There are more slower typists who make use of word suggestions, but by doing so they type slower

We found that word suggestions decrease typing performance on mobile devices. This confirms observations from prior studies [3, 32, 33]. However, our speed analysis controls for a confounding factor not considered by prior work: natural typing speed. This allowed us to quantify the effect the suggestion usage rate has on a person's typing performance. As a result, our work contributes to the literature with the insight that word suggestions and typing speed are related in *two* directions: 1) there are more slower typists that use word suggestions in the first place, but 2) by doing so they type even slower.

We discuss two possible explanations: Either, slow typists are aware of their slow typing and therefore use suggestions in an effort to increase their typing speed (and fail to do so). Or, they are less concerned about speed and use suggestions to reduce typing effort or with some other objective in mind. The former motivates a discussion of the costs of using suggestions, the latter a closer look at the identified strategies. We discuss both aspects in the following sections (Section 7.2, Section 7.3)

7.2 Cognitive costs outweigh speed benefits

Why do suggestions decrease speed? Our results indicate two reasons: First, suggestions introduce cognitive costs, as users have to read the shown suggestions and check for a word that fits their intention. This is in line with related work, which reports increased costs for each additional word in a suggestion list [8, 13, 23].

Second – and beyond this literature – we found that suggestions reduced the speed of *manual keystrokes*: Suggestion users typed more slowly overall, even if we excluded the selection times and

controlled for natural speed. A possible explanation is that the mere presence of suggestions adds cognitive costs. This might be caused by users thinking about how and when to use suggestions while typing, or by users switching attention and gaze between the suggestion list and the keys of the keyboard.

This finding connects to theoretical observations by Kristensson and Müllners [25] on the goodness of a typing strategy using word predictions. They characterize the trade-off between the physical cost of typing and the cognitive cost of using word predictions, as well as the prediction algorithm. Here, we add two empirical insights: First, we empirically show how this trade-off plays out in practice across many people and their keyboards and suggestion algorithms. In particular, suggestions decreased performance but are still used. This leads to the second insight, namely that a closer look is needed at the underlying behaviour: If not for speed, which objectives motivate users to use suggestions? Our analyses concretely identified eight strategies, which extend the literature on typing with suggestions, as discussed in detail next.

7.3 Typists strategically use word suggestions as a "toolbar"

Our results challenge a dominant assumption in prior work on word predictions: Users do not simply select a word if it is close to their intended word. Instead, they use the list of suggestions as a kind of "shortcut toolbar" strategically checking it for specific words and ignoring it otherwise. For example, when applying the strategies of *contraction* and *capitalization*, users employ suggestions as shortcuts to avoid manually typing apostrophes and capitalization.

These and other strategies also highlight the importance of considering a much broader range of motivations when studying word suggestions and designing user interfaces that offer them: Users strategically choose suggestions depending on (perceived) properties of the word they type, not only to avoid typing it at all in the first place. Concretely, these are properties that make words cumbersome to type, for instance, because of their length, familiarity, or required operations on mobile keyboards (e.g. switching mode to CAPS or special characters). Such observations can be interpreted in terms of the theory of computational rationality: users adapt their typing behaviour to internal (motivation, skills) and external bounds (word and keyboard characteristics) [20, 31].

The most frequently occurring strategies in our analysis are *completion* and *correction*. Note that this contradicts the core premise of today's large language models: Such models generally assign higher scores to more *common* words (in a given context), as learned from a training corpus. However, as we show, *completion* and *correction* are mainly used as shortcuts for long and *uncommon* words. Therefore, our results suggest that current language models could be improved for the specific use case of word prediction in keyboards, for example, by adding a new cost term in model training that reflects the behaviour strategies and features identified here.

7.4 Limitations and future work

Since we relied on existing data from a transcription task [32] our list of strategies might exclude some strategies or bias their frequencies in our analysis. The self-selected sample of participants includes younger and more female users who took the test to learn about their typing speed. People with more diverse cognitive or motor abilities, in particular older people are not well represented in the dataset and might differ in their strategic use of predictions. The transcription task might have influenced users to check the presented text to resolve spelling uncertainties instead of using suggestions. Another strategy that is excluded because of the transcription task is the use of suggestions for inspiration. For other tasks, such as writing emails, short messages, or notes, the list of strategies and their overall usage might differ. To explore more strategies, we suggest to investigate these use cases in future work. Lab studies building on our findings here could also involve eye tracking, as gaze behavior can be expected to vary between strategies as well.

We focus on the analysis of word suggestion usage with default mobile keyboards of Android and iOS. However, many keyboards offer further writing support features, such as auto-completion and auto-correction or suggest whole phrases. Such features might affect the strategies or afford new ones. For example, we expect decreased use of *completion* and *correction* when typing with features that change/extend words automatically. Future work could study the potentially more complex strategies introduced by such features, for example, the likely pattern of correcting faulty or unwanted auto-completions and auto-corrections [3].

We have presented a simple yet interpretable logistic regression model to explore which features contribute to the selection of a word, and as a demonstration towards turning our analytical findings into actionable improvements for keyboards. This opens a space for future work to improve on our baseline model for applications in keyboards, for example, by investigating more features, training on different datasets, and exploring other models and training methods (e.g. Deep Learning). With our model and its evaluation metrics reported here, we provide a useful baseline for comparison.

7.5 Designing intelligent text entry systems for more than performance

In line with the literature, our analysis focused on input behaviour, yet what we found encourages a broader scope: Our findings imply that text suggestion systems could be improved on both the UI and algorithm level, considering the identified user strategies and underlying user motivations beyond speed. In this light, we argue that the perspective on suggestions in the text entry research community could be extended even further: Since we now know that speed is not the users' only motivation for using intelligent keyboard features, we argue that neither should it have to remain such a dominant goal for designers and developers.

What else might be considered? To give one example, future text suggestion systems could be mindful of factors of *agency*. Recent work has found effects of the UI design of suggestions on agency-related factors, such as perceived control and authorship when writing with phrase suggestions on a smartphone [26].

This broader perspective opens up fresh design directions, informed by our analyses here: For instance, mobile keyboard UIs could be designed to support the strategies more directly, adapt to them, or be open to adaptation by users. As a simple, concrete example, instead of a binary on/off switch for suggestions, keyboard apps could offer toggles for different strategies (e.g. suggest capitalization? suggest contractions? etc.). Furthermore, algorithms could be designed to show text that is informed not only by word frequency, but also by strategic value, or even agency-related values. For instance, a keyboard might suggest typical phrases in a functional business email but only suggest single words in a chat with a close friend or partner where "personal" authorship might matter more to the writer and receiver.

8 CONCLUSION

We have analyzed a large-scale realistic dataset of typing with and without word suggestions, leading to two key insights:

First, we have confirmed and enriched the emerging picture in the literature about the negative impact of word suggestions on typing speed: Concretely, controlling for natural typing speed enabled us to show that there are more people who type slowly to begin with who make use of word suggestions. They are then further slowed down by doing so.

This raises the question of why and how people make use of word suggestions, if not to improve their speed, leading to our second insight: Users employ specific strategies when using suggestions. Concretely, we identified eight such strategies here, including using suggestions for completion, correction, capitalization, and contraction.

In conclusion, a main takeaway from our analysis for the community is that the suggestion list in the UI is used more like a "toolbar" – a perspective which we argue may open up powerful new design directions. Furthermore, this also has implications for algorithms and models: For example, current language models generally assign higher scores to more common words but users employ strategies that desire suggestions as shortcuts for uncommon or cumbersome to type words.

Overall, our findings thus encourage further research with a much broader perspective on user motivations and needs for mobile text suggestions. We hope our empirical analysis and its discussion encourage future research and design around intelligent text entry systems to look beyond speed.

We make our processed data and analysis scripts available, https://osf.io/u9aej/.

Acknowledgments

This project is funded by the Bavarian State Ministry of Science and the Arts and coordinated by the Bavarian Research Institute for Digital Transformation (bidt). Part of this work was done while the last author was a researcher at ETH Zurich.

References

[1] 2022. *Statistical Keyboard Decoding.* Cambridge University Press, 188–211. https://doi.org/10.1017/9781108874830.009

[2] Jiban Adhikary, Max Isom, and Keith Vertanen. 2022. The Impact of Number of Predictions on User Performance in a Dwell Keyboard. In *MobileHCI 2022 Workshop on Shaping Text Entry Research for 2030.* event-place: Vancouver, BC.

[3] Ohoud Alharbi, Wolfgang Stuerzlinger, and Felix Putze. 2020. The Effects of Predictive Features of Mobile Keyboards on Text Entry Speed and Errors. *Proceedings of the ACM on Human-Computer Interaction* 4, ISS (Nov. 2020), 1–16. https://doi.org/10.1145/3427311

[4] 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 (Sept. 2006), 146–154. https://doi.org/10.1080/10400435.2006.10131913

[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).* Association for Computing Machinery, Tokyo, Japan, 603–608. https://doi.org/10.1145/2984511.2984584

[6] Daniel Buschek and Florian Alt. 2015. TouchML: A Machine Learning Toolkit for Modelling Spatial Touch Targeting Behaviour. In *Proceedings of the 20th International Conference on Intelligent User Interfaces* (Atlanta, Georgia, USA) *(IUI '15).* Association for Computing Machinery, New York, NY, USA, 110–114. https://doi.org/10.1145/2678025.2701381

[7] 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.* ACM, Montreal QC Canada, 1–14. https://doi.org/10.1145/3173574.3173829

[8] Daniel Buschek, Martin Zürn, and Malin Eiband. 2021. The Impact of Multiple Parallel Phrase Suggestions on Email Input and Composition Behaviour of Native and Non-Native English Writers. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21).* Association for Computing Machinery, New York, NY, USA, Article 732, 13 pages. https://doi.org/10.1145/3411764.3445372

[9] Leonard S. Cahen, Marlys J. Craun, and Susan K. Johnson. 1971. Spelling Difficulty. A Survey of the Research. *Review of Educational Research* 41, 4 (Oct. 1971), 281. https://doi.org/10.2307/1169440

[10] Stuart K. Card, Thomas P. Moran, and Allen Newell. 1983. *The Psychology of Human-Computer Interaction.* Lawrence Erlbaum. https://books.google.de/books?hl=de&lr=&id=2EoPEAAAQBAJ&oi=fnd&pg=PP1&ots=DSSP2kUOhL&sig=lQMAhA40COrxLlnTllkKlcuY8rE&redir_esc=y#v=onepage&q&f=false

[11] Anna Maria Feit. 2018. Assignment Problems for Optimizing Text Input. , 182 + app. 56 pages. http://urn.fi/URN:ISBN:978-952-60-8016-1

[12] 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* (Seoul, Republic of Korea) *(CHI '15).* Association for Computing Machinery, New York, NY, USA, 649–658. https://doi.org/10.1145/2702123.2702503

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

[14] Nestor Garay-Vitoria and Julio González-Abascal. 1997. Intelligent word-prediction to enhance text input rate (a syntactic analysis-based word-prediction aid for people with severe motor and speech disability). In *Proceedings of the*

2nd international conference on Intelligent user interfaces - IUI '97. ACM Press, Orlando, Florida, United States, 241–244. https://doi.org/10.1145/238218.238333

[15] Mayank Goel, Leah Findlater, and Jacob Wobbrock. 2012. WalkType: using accelerometer data to accomodate situational impairments in mobile touch screen text entry. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, Austin Texas USA, 2687–2696. https://doi.org/10.1145/2207676.2208662

[16] Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th international conference on Intelligent user interfaces - IUI '02*. ACM Press, San Francisco, California, USA, 194. https://doi.org/10.1145/502716.502753

[17] Project Gutenberg. 2006. Wiktionary:Frequency lists/PG/2006/04/1-10000. https://en.wiktionary.org/wiki/Wiktionary:Frequency_lists/PG/2006/04/1-10000 last accessed: 2023-01-25.

[18] Andrew Hard, Kanishka Rao, Rajiv Mathews, Swaroop Ramaswamy, Françoise Beaufays, Sean Augenstein, Hubert Eichner, Chloé Kiddon, and Daniel Ramage. 2018. Federated Learning for Mobile Keyboard Prediction. https://doi.org/10.48550/ARXIV.1811.03604

[19] D. Jeffery Higginbotham. 1992. Evaluation of keystroke savings across five assistive communication technologies. *Augmentative and Alternative Communication* 8, 4 (Jan. 1992), 258–272. https://doi.org/10.1080/07434619212331276303

[20] Jussi Jokinen, Aditya Acharya, Mohammad Uzair, Xinhui Jiang, and Antti Oulasvirta. 2021. Touchscreen Typing As Optimal Supervisory Control. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 720, 14 pages. https://doi.org/10.1145/3411764.3445483

[21] Maryam Kamvar and Shumeet Baluja. 2008. Query suggestions for mobile search: understanding usage patterns. In *Proceeding of the twenty-sixth annual CHI conference on Human factors in computing systems - CHI '08*. ACM Press, Florence, Italy, 1013. https://doi.org/10.1145/1357054.1357210

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

[23] Heidi Horstmann Koester and Simon Levine. 1998. Model simulations of user performance with word prediction. *Augmentative and Alternative Communication* 14, 1 (Jan. 1998), 25–36. https://doi.org/10.1080/07434619812331278176

[24] Andreas Komninos, Angeliki Tsiouma, Georgia Gogoulou, and John Garofalakis. 2022. Don't Look Up: The Cost of Attention to Stimulus Phrases in Mobile Text Entry Evaluations. *Pan-Hellenic Conference on Informatics*. https://doi.org/10.1145/3575879.3576015

[25] Per Ola Kristensson and Thomas Müllners. 2021. Design and Analysis of Intelligent Text Entry Systems with Function Structure Models and Envelope Analysis. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '21)*. Association for Computing Machinery, New York, NY, USA, Article 584, 12 pages. https://doi.org/10.1145/3411764.3445566

[26] Florian Lehmann, Niklas Markert, Hai Dang, and Daniel Buschek. 2022. Suggestion Lists vs. Continuous Generation: Interaction Design for Writing with Generative Models on Mobile Devices Affect Text Length, Wording and Perceived Authorship. In *Proceedings of Mensch Und Computer 2022* (Darmstadt, Germany) *(MuC '22)*. Association for Computing Machinery, New York, NY, USA, 192–208. https://doi.org/10.1145/3543758.3543947

[27] I. Scott MacKenzie and R. William Soukoreff. 2002. Text Entry for Mobile Computing: Models and Methods,Theory and Practice. *Human–Computer Interaction* 17, 2-3 (2002), 147–198. https://doi.org/10.1080/07370024.2002.9667313

[28] Roderick Murray-Smith, Antti Oulasvirta, Andrew Howes, Jörg Müller, Aleksi Ikkala, Miroslav Bachinski, Arthur Fleig, Florian Fischer, and Markus Klar. 2022. What Simulation Can Do for HCI Research. *Interactions* 29, 6 (nov 2022), 48–53. https://doi.org/10.1145/3564038

[29] Alan F. Newell, Lynda Booth, John Arnott, and William Beattie. 1992. Increasing literacy levels by the use of linguistic prediction. *Child Language Teaching and Therapy* 8, 2 (June 1992), 138–187. https://doi.org/10.1177/026565909200800203

[30] Per Ola Kristensson. 2018. *Statistical Language Processing for Text Entry*. Vol. 1. Oxford University Press. https://doi.org/10.1093/oso/9780198799603.003.0003

[31] Antti Oulasvirta, Jussi P. P. Jokinen, and Andrew Howes. 2022. Computational Rationality as a Theory of Interaction. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems* (New Orleans, LA, USA) *(CHI '22)*. Association for Computing Machinery, New York, NY, USA, Article 359, 14 pages. https://doi.org/10.1145/3491102.3517739

[32] 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*. ACM, Taipei Taiwan, 1–12. https://doi.org/10.1145/3338286.3340120

[33] 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*. ACM Press, New York, NY, USA, 83–88. https://doi.org/10.1145/2858036.2858305

[34] Dmitry Rudchenko, Tim Paek, and Eric Badger. 2011. Text Text Revolution: A Game That Improves Text Entry on Mobile Touchscreen Keyboards. In *Pervasive Computing*, Kent Lyons, Jeffrey Hightower, and Elaine M. Huang (Eds.). Vol. 6696. Springer Berlin Heidelberg, Berlin, Heidelberg, 206–213. https://doi.org/10.1007/978-3-642-21726-5_13 Series Title: Lecture Notes in Computer Science.

[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 (Jan. 1987), 181–191. https://doi.org/10.1080/07434618712331274499

[36] 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, Article 17 (feb 2009), 34 pages. https://doi.org/10.1145/1497302.1497307

[37] Keith Trnka, Debra Yarrington, John McCaw, Kathleen F. McCoy, and Christopher Pennington. 2007. The Effects of Word Prediction on Communication Rate for AAC. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Companion Volume, Short Papers (NAACL-Short '07)*. Association for Computational Linguistics, USA, 173–176. event-place: Rochester, New York.

[38] Keith Vertanen, Crystal Fletcher, Dylan Gaines, Jacob Gould, and Per Ola Kristensson. 2018. The Impact of Word, Multiple Word, and Sentence Input on Virtual Keyboard Decoding Performance. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, Montreal QC Canada, 1–12. https://doi.org/10.1145/3173574.3174200

[39] Keith Vertanen and Per Ola Kristensson. 2011. A versatile dataset for text entry evaluations based on genuine mobile emails. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services - MobileHCI '11*. ACM Press, Stockholm, Sweden, 295. https://doi.org/10.1145/2037373.2037418

[40] Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating Fast Mobile Text Entry using Sentence-Based Decoding of Touchscreen Keyboard Input. (2015), 10.

[41] 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* (Toronto, Ontario, Canada) *(CHI '14)*. Association for Computing Machinery, New York, NY, USA, 2307–2316. https://doi.org/10.1145/2556288.2557412

[42] Seunghak Yu, Nilesh Kulkarni, Haejun Lee, and Jihie Kim. 2018. On-Device Neural Language Model Based Word Prediction. In *Proceedings of the 27th International Conference on Computational Linguistics: System Demonstrations*. Association for Computational Linguistics, Santa Fe, New Mexico, 128–131. https://aclanthology.org/C18-2028

APPENDIX

A.1 Computation of speed metrics

We define different speed measures (all in keystrokes per second) to be able to disentangle cognitive and motoric processes in our performance analysis. Table 4 specifies the keystrokes whose inter-key intervals are included in the computation of each metric.

Measure	Keystrokes
Natural speed	• Must belong to words which were typed without the use of ITE's (i.e. no autocorrection or suggestion was used). • Must be a lowercase letter. • Must be preceded by a lowercase letter. • Must belong to words that are shorter than 5 letters. • Must belong to words that have a frequency higher than 100,000 [17].
Letter-only speed	• Must be a lowercase letter. • Must be preceded by a lowercase letter.
Letter-and-selection speed	• Must be a lowercase letter. • Must be preceded by a lowercase letter. OR • Is the keystroke representing the selection of a word from the suggestion list.
Overall speed	• All keystrokes are included.

Table 4. Description of keystrokes that are included in the computation of the different speed metrics.

A.2 Detection of suggestion usage strategies

Table 5 describes the criteria for detecting a strategy from the typing log.

A.3 Evaluation of the selection model

For estimating P_S, we use a logistic regression model. We chose logistic regression because it is probabilistic in nature. instead of making making a binary prediction whether a user would choose an offered suggestion or not, it outputs a continuous probability, which lends itself well to be used alongside the language model. To train the model, for each strategy, we perform a One vs. Rest analysis. *Fixup* and *space insertion* strategies were excluded since they occurred too infrequently. The features we include in the model are summarized in Table 6. They are based on our empirical findings and our domain knowledge. Some features were left out for certain strategies if they were not relevant. In particular, *leadup_speed* and *leadup_length* are not relevant for *prediction*, in which case no characters are typed yet for the currently suggested word. Similarly, for *contraction* and *capitalization*, we exclude *leadup_length*, since these strategies require the user to type the entire word before selecting the suggestion.

Strategy	Detection criteria	Comment
Completion	edit_distance is greater than 0 AND edit_distance equals len_diff AND The previous keystroke belongs to the same word as the current keystroke	A completion where even just one letter was corrected is classified as a correction. E.g. completing "reps" into "responsibility" is considered a correction due to the initial one-letter spelling mistake.
Correction	Case 1: edit_distance does not equal len_diff AND The previous keystroke belongs to the same word as the current keystroke. Case 2: Was classified as completion AND len_diff == 1 AND Last character of the previous text field is equal to the last character of the current text field AND The last two letters of the current text field are not equal	In the case where correction causes a len_diff equal to edit_distance (i.e. letters are added), we misclassify it as a completion. Case 2 partially corrects this by checking that a completion caused the last character of the text field to change. But this only corrects the cases where len_diff == 1. That being said, a correction causing an edit_distance and len_diff greater than 1 is rare.
Prediction	Only one keystroke in the word AND Last character of the previous text field is a space	
Contraction	edit_distance == 1 AND len_diff == 1 AND The selected word from the suggestion list contains an apostrophe AND The previous word string does not contain an apostrophe	
Capitalization	edit_distance == 0 AND len_diff == 0 AND The letters typed prior to the selection are all lower case AND The word selected from the suggestion list contains a capital	
Fixup	Preceded by a backspace AND The previous keystroke belongs to the same word as the current keystroke AND The last character of the previous text field is not a space	This does not currently catch words that were fully erased. Moreover, due to limitations in the web backend, we can only detect instances where the user made changes to the end of the word. Therefore, changes made after placing the cursor in the middle of the word are not detected.

Table 5. Rules for classifying strategies.

To determine the weights of these features, we trained the model on 90% of the dataset, excluding a random set of 10% of the 84,156 words for testing. In addition, we excluded words with only one character (e.g. *I*) and words with more than 14 characters which are too rare to be used for training. We also excluded words that were selected accidentally, i.e. where the selected suggestion differs from the final word that was typed.

A.3.1 Feature importance The coefficients of the logistic regression for each strategy are shown in Table 7. The significance and magnitude of the coefficients provide insights into the features and

their goodness for predicting whether a suggestion will be used. Thus, they allow us to gain further insights into which types of words each strategy is used and how user behavior differs between them.

The coefficients for word frequency, word length, and lead-up speed confirm our previous empirical findings that *completion* and *correction* are used on infrequent words. Only *prediction* is used for more frequent words. Similarly, the coefficients confirm that suggestions are used for longer words, where *word_length* in particular explains the use of *completion*. The *lead_up* speed indicates that suggestions are typed slower also before the suggestion is chosen, except for *contractions* which are typed somewhat faster before using a suggestion, likely because the user does not need to perform additional keyboard events to enter the apostrophe but can use the suggestion algorithm to insert it.

The *leadup_length* shows that users type fewer letters leading up to a *completion* than they do leading up to a *correction*. This confirms our earlier analysis of the edit distance associated with each strategy (Figure 5).

Some features we assumed to be relevant turned out to be less important. We still included them here to inform future work. We assumed, for example, touch offset might be positively correlated with using a suggestion since it represents touch keystrokes in areas of the screen that are inconvenient to reach. However, the influence of the offset on suggestion usage is very small in this model. Thus, discomfort might not motivate users to select suggested words or another metric might be more representative of discomfort during typing. Likewise, we find only a small effect of the number of backspaces. This might indicate that users are not motivated by the error-proneness of a word when using suggestions or they are unable to anticipate how error-prone a word is.

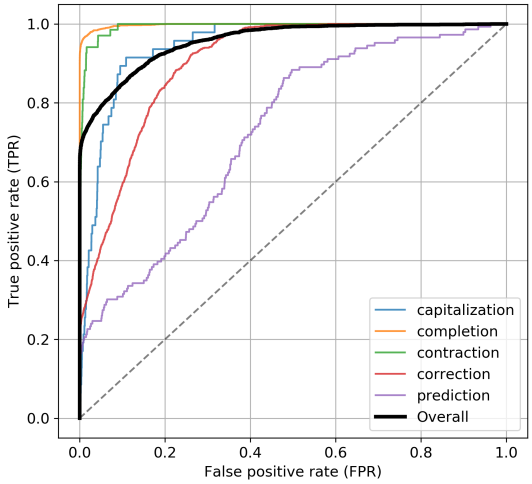


Fig. 8. Performance of the selection model.

A.3.2 Model performance for predicting suggestion use We excluded 10% of the data from training to use for testing the performance of our model. The logistic regression model outputs a probability that a word was typed by selecting a suggestion. This probability can be converted into a binary prediction by setting a threshold to distinguish between a manually typed word and a selected suggestion. Based on such binary predictions we evaluate the performance of the model. Note that the binary prediction is only used here to get an idea about the performance of the model. The

Feature name	Description	Relevance
word_length	The length, in number of characters, of the word.	Longer words may be more complex or take longer to type and thus suggestions are more likely to be used.
frequency	The log frequency of the word. This represents how common the word is in the English language [17].	Like word length, frequency is a proxy for the complexity of a word [9].
base_speed	Average characters per second across all instances of the word being typed manually (i.e. without ITEs). Includes only words typed more than 20 times.	Words that are complex or difficult to type would take a long time to type (on a character basis).
touch_offset	The average offset between the touchpoint and the center of the key. Data provided by Buschek et al. [7].	Words that involve pressing keys with larger offsets may be associated with more errors or more discomfort while typing.
backspaces	The average number of backspaces used when the word was typed normally in the dataset. Only words that were typed more than 20 times are included.	Words that on average incur more backspacing might indicate that the word is difficult to type or spell.
is_contraction	Whether the word is a contraction or not. Determined based on whether it contains an apostrophe.	Typing an apostrophe often requires additional keystrokes and therefore users may prefer to use suggestions to automatically insert apostrophes.
is_capitalized	Whether the word contains capital letters.	Capitalizing letters requires at least one additional keystroke (e.g. the "shift" key), and therefore users may prefer to use suggestions to automate capitalization.
leadup_speed	The typing speed, in characters per second, for the characters, typed so far.	A slower typing speed may reflect a user inspecting the suggestion list while typing, and therefore may indicate that the user intends on selecting a suggestion.
leadup_length	The number of characters typed before a suggestion is chosen.	Strategies vary in the number of letters being typed relative to the overall word length.

Table 6. List of word features used in the selection model.

application suggested in section 6.1 uses the continuous probability and does not define any fixed threshold.

Figure 8 visualizes the performance of our selection model as receiver-operating characteristic (ROC) curves. We use this ROC curves for a visual evaluation of the overall performance of the selection model and each strategy separately. These ROC curves show the performance of the model at all thresholds. Put briefly, the curves can be interpreted by visually estimating the area under the curve (AUC); the bigger the AUC, the better the performance of the model.

	Capitalization	Completion	Contraction	Correction	Prediction
const	-11.14*** (0.17)	-8.3*** (0.05)	-12.96*** (0.58)	-5.87*** (0.02)	-6.42*** (0.03)
word_length	0.37*** (0.04)	**5.58*** (0.03)**	-0.09 (0.1)	0.77*** (0.01)	0.53*** (0.04)
frequency	-0.61*** (0.07)	-0.41*** (0.02)	0.2 (0.11)	-0.39*** (0.01)	0.51*** (0.04)
is_contraction	-0.7 (1.13)	0.53*** (0.09)	**12.14*** (0.61)**	-1.05*** (0.1)	0.01 (0.9)
is_capitalized	**5.4*** (0.17)**	-0.24*** (0.07)	-1.44 (10.84)	-0.6*** (0.05)	-0.08 (0.25)
base_speed	0.53*** (0.05)	0.06*** (0.02)	-0.13** (0.06)	0.12*** (0.01)	0.14*** (0.03)
touch_offset	0.08*** (0.04)	0.04*** (0.01)	0.05 (0.06)	-0.04*** (0.01)	-0.11*** (0.02)
backspaces	-0.0 (0.03)	-0.25*** (0.02)	-0.2*** (0.03)	-0.7*** (0.01)	-0.57*** (0.05)
leadup_speed	**-1.76*** (0.09)**	**-3.37*** (0.04)**	0.26*** (0.06)	**-1.97*** (0.02)**	N/A
leadup_length	N/A	**-5.47*** (0.03)**	N/A	-0.3*** (0.01)	N/A

Table 7. Coefficients of the selection model for each strategy. *** = p < 0.01, ** = p < 0.05; non-sig. in gray, $|c|$ > 1 in bold.

Across strategies, the model performs better to predict *completion* and *contraction* with a rather large AUC, while the predictive power for the strategy *prediction* is rather limited showing a smaller AUC, corresponding to the low coefficients of the features for this strategy. Since distributions of strategy usages varied in the dataset some curves show steps, for example *contraction* and *capitalization*. With more data, these curves would become more smooth.

To provide a more specific example, we set the threshold to 0.05 to investigate the performance at a point that is close to the "elbow" of the ROC curve. At this threshold, the inherent probability that a user selected a suggestion is quite low. As described in Section 5.2.1, for example, a very long word only has a 25% chance of completion. At a threshold of 0.05, the accuracy of predicting a selection is 92.6%, and the precision and recall are 32.9% and 81.0%, respectively. With this threshold, our model is thus reasonably good at predicting the use of a suggestion with a low rate of false negatives, but with a high rate of false positives.

This evaluation shows that our model performs reasonably well and could be used in combination with existing suggestion algorithms. In our envisioned application, the purpose of the selection model is not to make a binary prediction but to estimate a continuous probability to rank words based on their likelihood of being suggested.

Received January 2023; revised May 2023; accepted June 2023