Understanding Adoption Barriers to Dwell-Free Eye-Typing: Design Implications from a Qualitative Deployment Study and Computational Simulations

Per Ola Kristensson
pok21@cam.ac.uk
University of Cambridge
United Kingdom

Morten Mjelde
Morten.Mjelde@tobiidynavox.com
Tobii-Dynavox
Norway

Keith Vertanen
vertanen@mtu.edu
Michigan Technological University
USA

ABSTRACT

Eye-typing is a slow and cumbersome text entry method typically used by individuals with no other practical means of communication. As an alternative, prior HCI research has proposed dwell-free eye-typing as a potential improvement that eliminates time-consuming and distracting dwell-timeouts. However, it is rare that such research ideas are translated into working products. This paper reports on a qualitative deployment study of a product that was developed to allow users access to a dwell-free eye-typing research solution. This allowed us to understand how such a research solution would work in practice, as part of users' current communication solutions in their own homes. Based on interviews and observations, we discuss a number of design issues that currently act as barriers preventing widespread adoption of dwell-free eye-typing. The study findings are complemented with computational simulations in a range of conditions that were inspired by the findings in the deployment study. These simulations serve to both contextualize the qualitative findings and to explore quantitative implications of possible interface redesigns. The combined analysis gives rise to a set of design implications for enabling wider adoption of dwell-free eye-typing in practice.

CCS CONCEPTS

• **Human-centered computing** → **Empirical studies in accessibility**; **Accessibility theory, concepts and paradigms**.

KEYWORDS

eye-typing, dwell-free eye-typing, gaze communication

ACM Reference Format:
Per Ola Kristensson, Morten Mjelde, and Keith Vertanen. 2023. Understanding Adoption Barriers to Dwell-Free Eye-Typing: Design Implications from a Qualitative Deployment Study and Computational Simulations. In *28th International Conference on Intelligent User Interfaces (IUI '23), March 27–31, 2023, Sydney, NSW, Australia.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3581641.3584093

IUI '23, March 27–31, 2023, Sydney, NSW, Australia

ACM ISBN 979-8-4007-0106-1/23/03.
https://doi.org/10.1145/3581641.3584093

1 INTRODUCTION

Eye-typing is a text entry method that allows users to communicate using an eye-tracker. The user writes by gazing at individual letter keys on the keyboard. The written text can be used in a number ways, for example, it can be used to write messages on social media, emails and essays, or it can be sent to a speech synthesizer to assist a user in face-to-face to communication. The ability to write with their eyes is indispensable for nonspeaking individuals with motor disabilities that make other means of communication impractical or impossible. Given its importance it has been extensively studied (e.g. [23]) and numerous techniques and alternatives have been proposed in the literature (e.g. [14, 19, 22, 23, 27, 31, 35, 47, 48]).

Traditional eye-typing works by presenting the user with an onscreen keyboard on a screen mounted in front of the user. The user's gaze on the screen is estimated by an eye-tracker, which in an assistive product that is typically integrated into the screen. The user writes text by fixating at each desired letter key in turn. A fundamental problem of this approach is that the system must be able to interpret whether a user intends to *type* the key at the user's estimated gaze point, or whether they just wish to *look* at the key. This is sometimes referred to as the Midas touch problem [6]. Traditional eye-typing tackles this problem using a dwell timeout. At the moment a fixation is detected on a letter key, the system triggers a visual timeout indication. If the user maintains a fixation on the key for a set threshold (typically between 800 and 1,600 ms) the system will interpret the user's action as a key press.

Traditional eye-typing is relatively easy to design, implement, and deploy to users. However, it has three well-known deficiencies. First, performance is bounded by the dwell timeouts. For example, a performance model of eye-typing [14] shows that for any reasonable operating point, performance is limited to about 20 words-per-minute[1] (wpm) at most. Second, it is straining to use eye-typing as it is unnatural to force the eyes to fixate on a series of targets on a screen. The eyes are sensory organs foremost and control organs secondarily [49]. Third, due to the need to manage dwell timeouts, the act of typing a single letter becomes a high-level task for the user and this breaks the flow of writing. It is not desirable to devote substantial attention to individual letter input when writing since the act of writing involves transmitting thoughts into a computer and such thoughts consists of words, phrases, sentences, and paragraphs.

Dwell-free eye-typing [14] proposes to eliminate dwell timeouts via statistical decoding. To use dwell-free eye-typing, the user writes

[1]In this paper *entry rate* is measured in words per minute (wpm), with a word being defined as five characters (including space).

by quickly fixating at each desired letter in sequence. When the user fixates at some ending location, the user's gaze fixations are converted to text. Dwell-free eye-typing allows the user considerable freedom of operation: a user can choose to type a word, a phrase, or an entire sentence at a time. The paper that proposed dwell-free eye-typing [14] set out a compelling vision for a dwell-free eye-typing system: a system that allows users to write characters, words, and sentences uninterrupted by dwell timeouts. That paper also demonstrated the human performance potential of such a technique by studying potential entry rates. Later, this vision was realized as a working product and deployed as a free software update for all users of the commercial assistive gaze product Tobii-Dynavox Communicator.

The central contribution of this paper is a qualitative *deployment study* [1] that demonstrates how walking the last mile and transforming a research concept into a working product made available to end-users in their own homes and familiar working setups can yield design insights and know-how for improving gaze-based assistive communication. There is very little research work in the text entry space that (1) is translated into actual products; and (2) is studied in terms of practical adoption barriers facing end-users. The most similar work is the path towards commercialization of gesture typing, originally called "SHARK^2" and "ShapeWriter" [51, 52].

We report on a qualitative *in situ* deployment study of this dwell-free eye-typing product made available to users reliant on gaze-based text entry. We visited six nonspeaking eye-typing users highly familiar with, and reliant on, the conventional eye-typing interface option in Tobii-Dynavox Communicator. Through observations and interviews, we analyze positive and negative qualities induced by the current iteration of dwell-free eye-typing and identify barriers for adoption that need to be overcome to ensure widespread adoption of the technique.

To complement this qualitative deployment study, we carry out computational simulations to understand the implications of certain design parameter choices that were either suggested by users or could potentially alleviate some of the adoption barriers we observed in the user study.

In summary, this paper makes the following contributions:

- We report on a qualitative *in-situ* deployment study of a dwell-free eye-typing interface made available to users reliant on a commercial eye-typing product in their daily lives.
- Using computational simulations directly inspired by findings in the deployment study, we explore a range of design parameters that could mitigate issues identified in the user study.
- We identify five current adoption barriers of dwell-free eye-typing and distill design implications to assist future work in this space.

2 RELATED WORK

Conventional dwell-based eye-typing has been extensively studied for over 40 years [23]. Majaranta et al. [22] studied the effect of allowing users to modify the dwell-timeouts and found that under highly controlled conditions, it was possible for users to achieve an entry rate close to 20 wpm. As later demonstrated by a human performance model [14], this operating point is probably the upper-bound for dwell-based eye-typing assuming no use of word prediction. Alternatives to dwell-based eye-typing have also been investigated, for example *Dasher* [46, 47], which results in entry rates similar to that of dwell-based eye-typing with adaptive dwell-timeouts [30, 31, 35].

Dwell-free eye-typing [14] eliminates dwell-timeouts entirely. The original vision was an eye-typing interface that allows users to write text of variable length unconstrained by dwell-timeouts whatsoever [14]. As noted in this prior work [14], such a system should in theory be possible to design since languages are highly redundant [34] and therefore a statistical decoder should be able to search for plausible letter key combinations guided by a language model. This is reminiscent to how, for example, continuous speech recognition [25] decodes a user's acoustic signal into text.

A more closely related example is early work on fixation tracing that used hidden Markov models to perform isolated word recognition among a set of 1,000 words [32, 33]. Finally, another related example is gesture keyboard technology [8, 17, 50, 51]. The central idea of the gesture keyboard is to allow the user to articulate gestures for words by initially tracing the words on an onscreen keyboard (typically using a finger). After prolonged use, users learn to recall such gestures directly from motor memory and thereby transition from slow closed-loop tracing to fast open-loop gesturing [8, 17, 50, 51]. Gesture keyboards can achieve this using a decoder that infers the user's intended word given the gesture the user has articulated over the keyboard layout [8, 17, 50].

Dwell-free eye-typing [14], as originally envisioned, is different in that it allows the user to continuously gaze at a series of characters that may comprise a few letters, a word, a phrase, or a sentence. This minimizes the use of dwell-timeouts and can therefore, in theory, maximize performance. Prior work [14] found that using a simulated decoder, the empirical human performance potential of such dwell-free eye-typing was on average 46 wpm with able-bodied users. However, no actual decoder was used in the original paper. The same paper also demonstrated through a human performance model that for every conceivable operating point, dwell-free eye-typing will be faster than dwell-based eye-typing. At the optimal operating point of conventional eye-typing, dwell-free eye-typing is potentially more than twice as fast [14]. However, actual practical entry rates depend on the performance of the statistical decoder, the individual user, and the user's context, including their eye-tracking setup.

Several other works have later used the term 'dwell-free' (e.g. [19, 26, 27]), however, these techniques use relatively simple models for decoding and rely on explicit word separation, which enforces a hard upper bound on achievable performance. However, notably Pedrosa et al. [27] studied achievable entry and error rates with their system with six users with disabilities in a lab study.

The existing literature has focused heavily on entry and error rates with controlled stimuli (e.g. [21, 41]), presentation styles (e.g. [13]) and text entry tasks (e.g. [4, 43]). In this work we instead focus on barriers for adoption by studying how users engage with a dwell-free eye-typing feature in their ordinary assistive gaze communication setup, which they rely on in their daily lives. Such a *deployment study* [1] of HCI research translated into a product opens up the potential for rich and unique feedback to the research community that is unavailable unless a potential research solution is made to walk the last mile. While user adoption of HCI products, including text entry methods, have been studied before (e.g. [2, 9, 11]), we believe such *in situ* studies are particularly important for dwell-free eye-typing as such a setup relies on a complicated joint human-machine system consisting of, among other things, a decoder, an eye-tracker, a speech synthesizer, an elaborate user interface and, as we shall see, the expectations and needs of the individual user reliant on such technology. Many critical design parameters for user adoption are therefore likely to only be observable by carrying out a deployment study of an actual product made available to end-users reliant on such a technique for everyday communication.

3 DWELL-FREE EYE-TYPING PRODUCT INTERFACE

The dwell-free eye-typing approach deployed to end-users closely follows the approach proposed in prior work [14]. As part of a commercial gaze assistive communication product, Tobii-Dynavox Communicator, a dwell-free eye-typing feature is available as a free software update. Once installed, it allows access to a different keyboard than the ordinary eye-typing keyboard. This new keyboard implements dwell-free eye-typing by allowing users to write characters, words, or sentences by sequentially gazing at the intended onscreen keyboard keys that comprise the intended text (Figure 1). For example, a user can write "the cat sat" by sequentially gazing at the keys T-H-E-[space]-C-A-T-[space]-S-A-T. When the user gazes at an individual key the letter lights up as a visual indication to the user that the key has been registered as an observation by the system. Importantly, however, users do not have to fixate on a key for a set duration for the key to be registered. Once the user fixates on the output area the system infers the user's intended text from the sequence of registered observations and presents the resulting text to the user in the output area. The keyboard uses the QWERTY layout.

The interface allows users to perform error correction using several methods. First, once the text has been inferred it is placed in the output area (Figure 2). Above the output area the system displays the next best hypotheses to the user. The user can choose among these hypotheses by fixating on the corresponding *Select* key. To assist the user, the part of the alternative text hypotheses that are different to the text in the output area is rendered in a distinct text format. Second, the user can enter a fine-grained error correction interface by fixating on the *Correct* key to the right of the output area. This opens up a detailed correction interface (Figure 3). Here the user can select the individual words in the recognized phrase by fixating on them. Once a word has been fixated on, the user can either choose an alternative word candidate or enter a dwell-based interface that allows the user to manually enter the desired word (via the *ABC* key).

Dwell-free recognition is implemented as a statistical decoder that uses Bayesian inference to search a very large hypothesis space of possible letter sequences. The statistical decoder receives an observation sequence of gaze points from the eye-tracker and searches for the most likely hypotheses of the user's intended text.

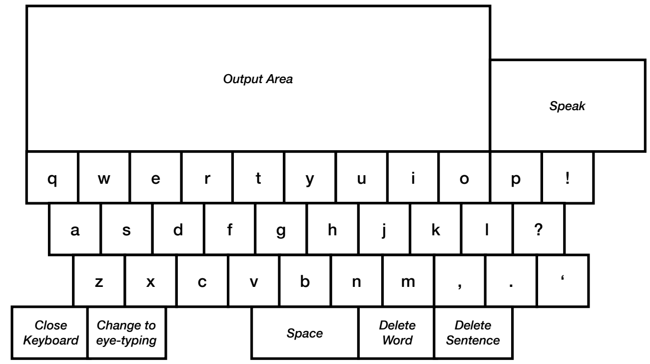

Figure 1: A schematic illustration of the dwell-free eye-typing keyboard interface in Tobii-Dynavox Communicator. A user's recognized text is shown in the *Output Area*. The *Close Keyboard* key closes the dwell-free eye-typing keyboard application. The *Change to eye-typing* key changes the mode from dwell-free eye-typing to regular (dwell-based) eye-typing. The *Delete Word* and *Delete Sentence* keys remove the last word and last sentence respectively from the output area. The *Speak* key speaks the text in the output area via text-to-speech.

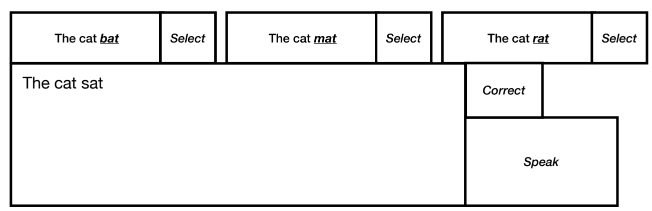

Figure 2: A schematic illustration of the dwell-free keyboard interface after the user has written "The cat sat". The user can quickly correct the output by choosing among three alternative text hypotheses. To assist the user, the differences in these alternatives compared to the text in the output area is made distinct via text formatting. The user can choose one of the alternatives by fixating on the *Select* key next to it. If none of the options are correct, the user can fixate on the *Correct* key to open a dedicated error correction interface.

This search is guided by a likelihood model of a gaze point corresponding to an individual key on the keyboard and a language model providing prior beliefs of the user's intended text, reminiscent to how a speech recognition decoder is implemented. To make the search tractable, we employ beam pruning to filter out partial hypotheses that have a low probability. The search process is guided by a character-level language model and each generated word is reassessed by a word-level language model. The complete technical details are published in the documentation for two granted U.S. patents [15, 16].

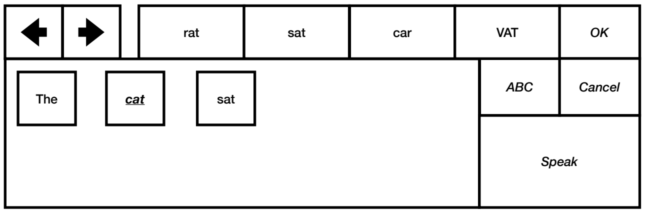

Figure 3: A schematic illustration of the dedicated error correction interface. The user can either fixate on an individual word in the output area or fixate on the left/right arrow keys to select an individual word that needs modification. Once an individual word is selected the interface will show alternative word hypothesis for this word. In addition, a pop-up menu appears (not shown in the figure) that allows the user to delete the selected word. The *ABC* key opens up a regular (dwell-based) eye-typing interface to input the desired word.

4 QUALITATIVE STUDY

We recruited six participants from among the company's customers. All participants were literate but could not speak. Three participants had amyotrophic lateral sclerosis (ALS) and three had cerebral palsy (CP). All used conventional eye-typing as their primary communication method. Four participants identified as men, two as women.

We informed participants about the purpose and procedure of the study during recruitment. We obtained informed consent before our visit. At the visit, the purpose of the study was reiterated by a familiar support team member from the company and informed consent was reconfirmed. This support team regularly visited the customers to help with technical problems or modifications of the product. Each visit lasted several hours and begun with a discussion of on-going issues and concerns with their set-up unrelated to dwell-free eye-typing. The participant then calibrated the eye-tracker and demonstrated their currently preferred setup for gaze writing.

We then introduced dwell-free eye-typing and explained how it works. We explained to the participants that the key they are currently looking at would light up. Participants were instructed that they should aim to light up each key in sequence when entering their text but they did not have to dwell on the keys. Participants were also told that they should not worry if unwanted keys lit up. Participants then wrote a few words and an example sentence using dwell-free eye-typing. This was followed by the participant writing a set number of test sentences. After each sentence, the participant was given entry and error rate feedback. Next, we showed the participant the error correction interface and asked them to carry out a few corrections or edits of previously input text.

After writing using the dwell-free feature, we discussed design features with the participants. Topics included accuracy, error correction, positive features, negative features, additional features, and differences in features compared to their existing set-up (which were customized to the individual). Finally we collectively reflected on the experience.

4.1 Individual Observations

We briefly summarize individual observations here to provide a better context of the study environment and study activities. Key findings across all participants are reviewed in the next subsection.

4.1.1 Participant A. Participant A identified as male, had ALS and was in the 40–50 age range. He used a QWERTY keyboard with traditional dwell-based eye-typing but experienced considerable difficulty in accurately controlling the gaze cursor for traditional eye-typing. We observed frequent sporadic and unintended rapid ballistic saccades. He had difficulty in reaching intended fixation points as the saccades tended to be too imprecise, requiring frequent closed-loop re-fixation attempts. Further, he had difficulty fixating at the center of a key (within a region of error) for short durations (approximately 800–1200 milliseconds).

The eye control difficulties resulted in severe difficulties in using the traditional dwell-based eye-typing interface. Eye-typing require the user to fixate at an intended location for a preset time duration. As Participant A experienced great difficulties in maintaining a fixation for the required duration, this often resulted in the dwell timeout for his intended key being reset. This was sometimes exacerbated by an inadvertent fixation on a neighboring key, again due to limited eye control.

As a result of limited eye control, overall entry rate was very low, around 2 or 3 words per minute. Due to the low entry rate, it was unsurprising that there was a heavy reliance on word predictions and a strong utilization of word predictions, when they were suitable. The overall typing pattern was to input a single letter and immediately scan all word predictions, only attempting to input another letter if all word predictions were unsuitable. This strategy was, however, unreliable due to limited eye control, which relatively frequently resulted in an inadvertent selection of an unintended word prediction. Participant A was very adept at selecting successive word predictions. However, due to poor eye control this also frequently resulted in selecting unwanted word predictions.

Dwell-free eye-typing was possible but with very poor recognition results. Due to limited eye control there were two behavioral gaze control issues which the statistical dwell-free eye-typing decoder could not handle: (1) difficulty in fixating on the intended key, which resulted in him eventually giving up and proceeding to the next key; and (2) a difficulty on retaining a fixation on the intended key, which frequently resulted in an oscillating pattern where two or three neighboring keys were activated repeatedly in close succession, for example AZAZAZAZAZAXAAZA.

Dwell-free eye-typing, as it is currently designed, does not work well for this participant. When asked why he did not use dwell-free eye-typing, the response was that the letters were too small (and thus difficult to fixate on) and "it feels like guessing", a reference to the poor recognition rate.

4.1.2 Participant B. Participant B identified as male, had CP and was in the 30–40 age group. He used a QWERTY keyboard with traditional dwell-based eye-typing. He understood the overall eye-typing and communication product very well and had reconfigured it extensively to fit his needs. He achieved a high quality calibration with the eye-tracker and was able to eye-type at a relatively high speed. He relied heavily on word predictions but did not use sentence predictions, which were also presented by the traditional eye-typing system. He had difficulty fixating for a long time on individual letters, which frequently resulted in early unwanted terminations of the dwell-timeout. He made extensive use of word predictions and adapted a strategy of typing a letter and then scanning the word suggestions.

Participant B was able to use dwell-free eye-typing at a relatively high speed (in relation to standard eye-typing). However, he did not use dwell-free eye-typing normally. He was able to write the three test sentences with few errors. In response to a question on why he did not use dwell-free eye-typing he responded that "[he] didn't get on with it."

By probing deeper the following issues and areas of improvement were unraveled. First, the system integration of dwell-free eye-typing was poor. Participant B showed us an example of typing a Facebook message using his own setup, which had superior integration. He suggested including a function to copy text to the clipboard, making the keyboard as similar as possible to the existing eye-typing keyboard, and to visualize the user's eye gaze using a big round cursor.

The preferred setup for dwell-free eye-typing according to Participant B would encompass: (1) full system integration (in particular, allowing sending text to arbitrary applications); (2) highlighting the entire letter key and letter when a user is gazing at it; and (3) allowing a tolerance such that a dwell-free eye-typing user would not need to look at the exact letter and only need to select a letter in the vicinity of the intended key.

4.1.3 *Participant C.*

Participant C identified as male, had CP and was in the 40–50 age group. He used an ABC (alphabetical) keyboard with traditional dwell-based eye-typing. He communicated using a screen with a built-in eye-tracker connected to his wheelchair via a stand. Due to involuntary movements, the chair and the stand frequently moved around, which was exacerbated by poor mechanical damping of the eye-tracking camera mount. His head frequently moved in and out of the tracking box. Despite this, his eye-tracker calibration was of a high quality.

Participant C was reliant on eye-typing and had difficulty hitting the precise letters. However, he did not oscillate between two neighboring letters. His eye gaze moved with very rapid saccades across the keys on the keyboard. As a result of the difficulty in selecting precise letters, dwell-free eye-typing performance was very poor.

He stated he believed dwell-free eye-typing would be more useful if it had a tolerance such that a dwell-free eye-typing user would not need to look at the exact letter and only need to select a letter in the vicinity of the intended key. He would also prefer if it provided integrated word predictions.

4.1.4 *Participant D.*

Participant D identified as male, had ALS and was in the 40–50 age group. He used a QWERTY keyboard with traditional dwell-based eye-typing. He was able to accurately fixate at visual targets and his eye gaze moved quickly and precisely between the keys. However, he experienced difficulties in managing a fixation at a letter key for a fixed duration. He mentioned he experienced this difficulty in all his dwell-based interaction and had found that increasing the dwell-timeout from 800 ms to above 900 ms mitigated some of the difficulty. He had a preference for a slower dwell-timeout. When using traditional eye-typing he relied heavily on word predictions.

Participant D was aware of the dwell-free eye-typing feature and was very fast and accurate when using dwell-free eye-typing. However, he did not use it as his main keyboard due to a number of issues:

(1) He would like it to look and behave as the regular traditional eye-typing keyboard and he would want to be able to fluidly switch between dwell-based and dwell-free keyboard entry, which currently was not possible.
(2) He would like to view the current system hypothesis of his intended text as he is writing. He would also like to view word predictions for dwell-free eye-typing.
(3) He would prefer to relax the requirement of selecting each letter in the intended word. He felt that if it would suffice to fixate briefly in the vicinity of the intended letter. This would provide a more relaxing experience when writing using dwell-free eye-typing.
(4) He was concerned about the treatment of punctuation and felt that there should either be a way to indicate to the system what the intended punctuation should be (before recognition) or there should be editing functions that make it easy to ensure accurate punctuation following recognition.
(5) At its current state and given his current experience, he felt dwell-free eye-typing was more suitable for speaking (via text-to-speech) than writing text intended to be read by someone.

4.1.5 *Participant E.*

Participant E identified as female, had ALS and was in the 60–70 age group. She used a QWERTY keyboard with traditional dwell-based eye-typing. She easily obtained a high quality calibration and exhibited excellent eye control. She could dwell-free eye-type fluently, and became quite attached to it during the testing session. However, interestingly, she had not seriously tried it before due to "lack of confidence".

She preferred flexible letter selection, if possible, where there would be no need to precisely select the intended letter. She liked an idea of splitting the message window into multiple end-of-utterance sections with specific punctuation symbols. For example, fixating on a comma key would trigger a decode of the previous observation sequence and append a comma at the end.

We asked her to consider whether it would be useful to integrate word prediction into the dwell-free eye-typing process or, alternatively, to have the system reveal the current active phrase or sentence (the system's best hypothesis of the user's intended text given the current observation sequence of gaze input). She was unsure whether this would be useful or not.

Overall, she was happy with dwell-free eye-typing and could use it with very few mistakes. The major flaw she observed was the lack of integration, which made it impractical for her daily communication tasks.

4.1.6 *Participant F.*

Participant F identified as female, had ALS and was in the 40–50 age group. She used a QWERTY keyboard with traditional dwell-based eye-typing. She easily achieved a high quality calibration and exhibited excellent eye control. She mastered dwell-free eye-typing easily and could write a wide range of phrases without any errors.

She would prefer more error tolerant recognition and was in particular in favor of language model adaptation so that the system would be more apt to recognize her phrases and style of writing. Further discussion about this revealed that by error-tolerance she meant the decoder being resilient to input errors in the current style of interaction (in which users were instructed to obtain at least one gaze point on each intended key). Additional error-tolerance in a form that removed the need to gaze at a precise key would be "amazing" but she did not deem this a critical feature.

She was unaware there even existed an error correction interface for the dwell-free keyboard. Upon trying it she requested a more efficient error correction interface that allowed her to directly delete and insert words.

4.2 Key Findings

We will now review the findings based on the interviews and observations. We analyzed the results based on a model consisting of four themes. First, the *target audience*: there is a wide range of conventional eye-typing users and we wanted to understand if all segments of conventional eye-typing users were able to, or could be made able to, access dwell-free eye-typing. Second, their *current strategies* in communicating and overcoming existing barriers in conventional eye-typing. Third, *efficacy of dwell-free eye-typing* as an interaction technique. Fourth, *barriers to adopting dwell-free eye-typing*, including barriers that were solely due to the specific product realization. We encoded our notes and recordings and used affinity diagramming guided by the above four themes to arrive at clustered issues. For clarity, we structure these results as key findings.

4.2.1 User Groups. The participants can be subdivided into three user groups. The first group consists of expert users. They are well-versed with the dwell-free eye-typing technique, yet for various reasons they do not utilize it to its full potential. The second user group captures intermediate users with potential to become expert users. They can potentially use dwell-free eye-typing to a level which would allow them faster communication rates overall. However, they are currently not at this level. The third user group consists of users that exhibit highly noisy gaze control. For various reasons, including potentially rectifiable issues such as tracking error or lack of sleep (Participant A), users in this group struggle with gaze control in one or more of the following aspects: (1) difficulty in precisely fixating at an intended gaze location; (2) difficulty in maintaining a fixation at an intended gaze location for a set duration; and (3) difficulty in maintaining a fixation for a very short duration in the order of tens of milliseconds.

4.2.2 Word Prediction Reliance. There was a high reliance on word predictions. This behavior cuts across all three user groups. This is unsurprising as good word predictions overall increase entry rate when users are rate-limited, which is the case when using traditional dwell-typing. It is evident all users have had extensive practice with word predictions and are often able to anticipate a correct prediction. This learned behavior acts as a user investment of effort in mastering the existing traditional dwell-dependent eye-typing interface and may increase friction when transitioning to the dwell-free keyboard as currently designed (i.e. without word predictions).

4.2.3 Dwell-Free Eye-Typing is Effective as a Text Entry Method. Among expert and intermediate users, the ability to fluidly use dwell-free eye-typing with minimum effort was high. These users were able to use the existing interface to quickly and accurately articulate their intended text with low effort. It was noticeably easier to communicate with these users when they used dwell-free eye-typing. However, many other issues beyond text entry performance precluded adoption of dwell-free eye-typing.

4.2.4 No Graceful Degradation. When the dwell-free eye-typing system output an incorrect inference of the user's intended text, the resulting text was often nonsensical and impossible to fix without deleting all the text and starting over. This frustrated users and increased their hesitation to use dwell-free eye-typing.

4.2.5 Ineffective Error Correction Interface. The error correction interface was not effective. There were often very few alternative words and they were often nonsensical or irrelevant. In addition, users had to take a leap of faith and invest considerable effort in changing to a special error correction mode and thereafter dwell on individual words before they were able to ascertain if there even were any suitable alternative word candidates available.

4.2.6 Lack of Transparency. Users perceived dwell-free eye-typing as "magic" and in the word of one user, "it feels like guessing". Occasional nonsensical results exacerbated this perception and the nearly always uninformative error correction interface unfortunately further reinforced this perception.

4.2.7 Lack of Confidence. In general, the participants lacked confidence in using dwell-free eye-typing. This prevents adoption of dwell-free eye-typing due to at least two observable factors. First, both expert and intermediate users underestimated their own performance. This is somewhat expected as users are generally poor at estimating their own objective performance. Second, there was resistance to change, which resulted in participants being surprised by their own performance when pushed to try dwell-free eye-typing. The resistance to change is rational as there is in general a trade-off between perceived, or real, effort required to learn a new technique versus a (positive) perceivable net gain in performance. This trade-off decision is further muddled by additional confounding variables that exacerbate the learning process for dwell-free eye-typing: a less fluid experience, poor integration, and poor support for interleaving existing typing practices, such as reliance on word predictions.

4.2.8 Poor Integration. The keyboard implementation of dwell-free eye-typing within the user interface suffers from poor integration. For example, it is not possible for users to seamlessly send text to external applications such as Facebook. It is also not possible to seamlessly switch between eye-typing and dwell-free eye-typing when, for example, there is a need to type a proper name or password. In addition, there is no easy copy-paste functionality within the dwell-free eye-typing keyboard.

5 COMPUTATIONAL SIMULATIONS OF DESIGN OPTIONS

Taken together, the findings from the qualitative deployment study suggest that the first product realizing dwell-free eye-typing may have been offering users too big a step change in requiring dwell-free input of entire sentences. This was simply too far from the deterministic and word-at-a-time input process users were already highly accustomed to. Further, recognizing entire sentences, including guessing unspecified upper/lower case and difficult to predict punctuation symbols, led to unacceptably high recognition error rates.

The deployment study generated many ideas on how one might redesign the user interaction and recognition system to address problems identified by users. Developing and testing a large number of interface features with users, especially with actual gaze users, would be an expensive undertaking. Further, not only are there a number of possible new features, but also some features (e.g. word suggestions) have multiple variants (e.g. number of suggested words). In such cases, offering more interface options is in direct competition with offering users larger onscreen targets.

We therefore conducted computational simulations to better understand the potential performance of various redesign options. We investigated four design changes: (1) length of observation sequence (one word, two words, etc.); (2) letter locking—dwelling on individual keys while using dwell-free eye-typing; (3) offering alternative word recognition hypotheses; and (4) offering word predictions prior to dwell-free input. We also investigate the performance of combining the most promising features in tandem.

5.1 Approach

To investigate these design parameters, we first collected examples of dwell-free eye-typing. Our data collection interface first had users calibrate the eye-tracker. The interface then displayed a random sentence from the Enron mobile dataset [42]. The user wrote the sentence by looking at each letter in the sentence, including spaces. After dwelling on a stop button, the user could either retry the same sentence (discarding the data), or move to a new sentence. The data collector did not perform recognition; it only recorded eye-tracking data. In total, we collected 582 sentences from 24 users who were not motor-impaired.

Using our dwell-free decoder, the Character Error Rate (CER)[2] on this data, ignoring case but including punctuation, was 6.8%. We converted this data into a format compatible with the VelociTap decoder [44]. This research decoder was originally designed for touchscreen typing data. The decoder uses a two-dimensional Gaussian centered at each key to model the distribution of possible keys for each touch observation. Configurable penalties allow the decoder to delete or insert observations. The decoder performs a beam search for the most likely recognition hypotheses guided by a character and word n-gram language model. We used the same character and word language models reported in prior work [44]. For full details on the decoder's operation see prior work [36].

[2]Character error rate is here defined as the minimum number of character insertions, deletions and substitutions necessary to transform the decoded text into the source text, divided by the number of characters in the source text and multiplied by 100.

We adapted the VelociTap decoder to recognize our eye tracker data. We used the research decoder because it has features not present in our dwell-free decoder including the ability to simulate the performance of a user that makes perfect use of interface features such as recognition alternatives or next word suggestions. Our deployed dwell-free decoder has features to help delete short fixations and to insert multiple letters based on a single fixation (e.g. the letter "o" in "food"). These two features substantially improve accuracy on dwell-free input (without these features CER doubles). After adding these features to the research decoder, we obtained a similar CER of 7.4% compared to the dwell-free decoder.

The collected data was a continuous trace through all the letters in a sentence including spaces. However, for the purposes of the remaining experiments, we required sentence traces where each trace was segmented according to the corresponding word in its reference text. We used the research decoder to force align the traces to the reference transcripts. 491 of the 582 eye traces force aligned successfully. The traces that failed to align may have had input errors causing a mismatch in the number of words compared to the reference. Note that this means the force aligned dataset is somewhat easier on average than the complete dataset.

In our experiments, we played back the eye trace of each sentence to the research decoder one segment at-a-time. In most experiments (aside from Experiment 1), a segment corresponded to a single word. After every segment, we queried the recognizer for the most likely recognition results given the noisy eye data and any previous recognized text. We assume a simulated user that selects the correct option if available, otherwise the user selects the most likely (incorrect) option and carried on (i.e. we did not simulate having an error correction feature such as backspace). Note that this means that recognition errors earlier in a sentence may negatively impact subsequent predictions due to corrupting the decoder's language model context. We report the following metrics:

- **Final sentence CER** — The character error rate of the final text for a sentence. This will be greater than zero when the recognition results presented to the user did not allow exact writing of the target reference text.
- **Recognition accuracy** — How often the top recognition result was the target word. If the interface provided multiple recognition hypotheses, this was how often any of the hypotheses was the target word.
- **Recognition accuracy ignoring punctuation** — Similar to the previous metric but ignoring any end of word punctuation (comma, period, exclamation point, and question mark).

5.2 Experiment 1: Input Chunk Size

The current dwell-free keyboard uses sentence-at-a-time input. The statistical decoding process can instead operate on smaller chunks of input such as one or two words. This may provide a more familiar text entry method and could make correcting errors easier since corrections could be performed in a smaller amount of text. However, shorter chunks may be fundamentally harder to recognize due to a recognition decision being based on less information.

We simulated entry of various input chunk sizes. A chunk size of one corresponds to entering a single word prior to requesting

	Input chunk size (words)						entire
	1	2	3	4	5	6	sentence
Final sentence CER (%)	7.46	6.98	6.85	6.69	6.58	6.63	6.66
Recognition accuracy (%)	76.52	61.46	48.77	36.82	26.73	21.96	10.59
Recognition accuracy ignoring punctuation (%)	91.62	87.69	84.22	80.75	77.99	77.19	73.12

Table 1: Final sentence character error rate and how often recognition events were correct as the number of words in the input was increased.

	Locked letters			
	None	First	Last	First and last
Final sentence CER (%)	7.46	7.12	4.88	4.41
Recognition accuracy (%)	76.52	77.22	86.92	87.29
Recognition accuracy ignoring punctuation (%)	91.62	92.41	89.76	90.22

Table 2: Final sentence character error rate and how often recognition events were correct depending on which letters were locked based on the known reference word.

recognition, a size of two corresponds to entering two words prior to recognition, and so on. Note that for this experiment, we assumed that the location of the spaces between words was known. In the auspices of a user interface (UI), this would require a UI action, such as dwell-clicking on a space key. To support chunk sizes of two or more, the UI would need not only a space action, but also a separate action to signal recognition. Alternatively, recognition could be streaming with a result shown somewhere in the UI (e.g. as a prediction above the keyboard). This prediction would likely still require explicit confirmation (e.g. by dwell-clicking it).

As shown in Table 1, providing more words at one time improved recognition accuracy. This may be because it allows the recognizer to make a better global decision taking into account the overall sequence of words. It may also be due to the known (and correct) spaces reducing the final sentence error rate. While bigger chunks were better, this presumes we know with certainty how to segment observations into words.

We also measured how often recognition was completely correct. This measures how often users would experience a dwell-free input experience requiring no error correction. We measured this including and ignoring punctuation. The latter estimates the utility for someone using dwell-free for person-to-person communication rather than written communication. As expected, the larger the input chunk size, the less likely the decoder was able to infer the intended text completely correct. Notably, even with the most conservative input size of a single word and ignoring punctuation, only 9 out of 10 dwell-free inputs resulted in the correct word. This indicates that simply moving to word-at-time dwell-free input would not provide as accurate an experience as the deployment study participants desired.

Ignoring end of word punctuation increased accuracy markedly (e.g. from 77% to 92% for a chunk size of one). These characters were particularly hard to recognize as they are not very predictable under the current n-gram language model. A deficiency of such n-gram models is that contextual clues, such as the first word of a sentence (e.g. a sentence starting with "why") may be out of the model's context by the end of the sentence. Using a neural network language model may help as these models are better at modeling long-range dependencies and other latent factors. A further problem may be that punctuation keys were near each other on the keyboard. This leads to confusability in the keyboard's probability model. Thus it may be desirable to separate punctuation keys on the onscreen keyboard.

5.3 Experiment 2: Locked Letters

A possible hybrid input method might have a user dwell-click on the first letter of each word before performing a dwell-free trace over the remaining letters. Alternatively, the user could perform a dwell-free trace of a word and then dwell-click the final letter. This would allow the final dwell-click to serve not only to help the decoder, but also to request recognition. We simulated these options by replacing the first and/or last observation of every word with an observation that was locked to the known letter in the reference word (i.e. the decoder could not misrecognize locked letters). This experiment used an input chunk size of one word.

As shown in Table 2, locking letters improved accuracy in all cases. Locking the first letter only improved accuracy slightly. The improvement for locking the last letter was striking. Part of this gain was driven by the fact that many errors were from incorrect recognition of punctuation after the word (comma, period, question mark, and exclamation point). However, even ignoring punctuation we still found accuracy improved by 10% by locking just the last letter.

A possible redesign would be to use the dwell on the last letter to trigger recognition for the previous dwell-free input event. This would offer the benefit of the added accuracy, but it does assume the user can accurately perform a conventional dwell-click on an individual key. It also means the user will finish by looking at a different location from where recognition results normally appear (i.e. above the keyboard).

	Recognition slots offered				
	1	2	3	4	5
Final sentence CER (%)	7.57	6.39	5.56	5.19	4.97
Recognition accuracy (%)	76.23	80.31	83.41	85.06	85.89
Recognition accuracy ignoring punctuation (%)	91.58	94.06	94.68	94.92	95.01

Table 3: Final sentence character error rate and how often the correct word was in the set of recognition results for increasing number of slots populated with the most likely recognition hypotheses.

	Slots offered (next word / recognition n-best list)				
	1	2	3	4	5
Final sentence CER (%)	7.36	6.22	5.39	5.00	4.78
Recognition accuracy (%)	76.89	80.69	83.95	85.72	86.55
Recognition accuracy ignoring punctuation (%)	92.28	94.47	95.30	95.71	95.79
Next word prediction used (%)	25.28	31.56	36.09	38.57	41.44

Table 4: Final sentence character error rate and how often the correct word was in the set of recognition results for increasing numbers of slots. The same number of slots were used for the next word predictions prior to input and for recognition alternatives after input.

5.4 Experiment 3: Recognition Alternatives

We simulated word-at-a-time dwell-free input with a variable number of recognition alternatives after each word recognition. As shown in Table 3, providing not only the best recognition hypothesis, but also several of the most probable competing alternatives, improved accuracy. The majority of the gains were seen by allowing the user to choose between the top-3 recognition results. By offering the top-3 results, 95% of words could be written successfully (ignoring punctuation). However, this would still leaves 1 in 20 words needing some other error correction strategy, such as reverting to dwell-based input.

A possible redesign might append the best recognition result to the keyboard text result area. The appended text would be visually annotated to allow easy review by the user in a "safe" area that does not trigger further actions. Below the keyboard, we could present dwell buttons for the other top recognition results. We could also include a button to delete the previous recognition to allow the user to reattempt dwell-free input (or fallback to dwell-based typing).

5.5 Experiment 4: Next Word Prediction + Recognition Alternatives

Prior to dwell-free input of a word, prediction slots above the on-screen keyboard could be populated with words that might follow the current text. This allows the user to simply dwell on the desired word rather than risk a mistake during dwell-free input. In the deployment study, we found this was a popular feature used by participants in their conventional dwell keyboard.

We simulated this feature by first predicting the most likely words prior to each word in the force-aligned data. Note that the simulation only proposed word predictions without punctuation, thus it was inherently unable to predict words with commas, periods, question marks, or exclamation points. If the desired word was present, we assumed the user selected it. If the desired word was not present, we assumed the user provided the dwell-free input. This input was recognized and the user could then select from the n-best recognition results (similar to Experiment 3).

As shown in Table 4, the next word prediction feature was frequently able to predict the word before input. If three predictions were offered, 36% of all words could be entered using this feature. Comparing with Table 3, we also observe a small reduction in final sentence CER as well. This resulted from the next word predictions sometimes allowing the user to avoid an uncorrectable error that would have occurred if they had used dwell-free input for a word.

5.6 Combining Letter Locking, Word Predictions, and Recognition Alternatives

One possible interface design would be to offer next word predictions prior to the start of input of the user's next desired word. If dwell-free input was required, the user would generate a continuous eye trace through all the letters of a word with a final dwell-click that locked the final letter of the word. This event would signal to the decoder that it should populate a set of n-best recognition hypotheses that the user can select from. These recognition hypotheses can be located at the top of the keyboard area, or they can appear next to the final dwell-clicked character. The latter approach might afford a speed advantage as the recognition hypotheses can be displayed while the dwell-time was still being completed for the final character. If the desired word is shown, the user can then relocate their gaze to the desired word to start the dwell-click confirmation period for that word. The disadvantage of this approach is that it would be a departure from where users are currently accustomed to seeing predictions (i.e. above the keyboard).

As shown in Table 5, the combined design had the lowest sentence CER of any method tested. If three slots were available for

	Slots offered (next word / recognition n-best list)				
	1	2	3	4	5
Final sentence CER (%)	4.63	3.12	2.79	2.55	2.44
Recognition accuracy (%)	87.91	92.86	93.89	94.59	94.92
Recognition accuracy ignoring punctuation (%)	91.13	94.39	95.21	95.42	95.63
Next word prediction used (%)	22.03	27.73	32.62	35.17	38.04

Table 5: Final sentence character error rate and how often the correct word was in the set of recognition results for increasing numbers of recognition alternative hypotheses. The same number of slots were used for the next word prediction prior to input and for recognition alternatives after input. We assume the user dwelled on the last character of a word to lock it.

predictions and recognition hypotheses, the final CER was 2.8%. Recall that the original sentence recognition error rate provided by the research decoder was 7.4%. This represents a substantial improvement in accuracy. However, these results assume a hypothetical user who makes perfect use of the provided predictions and recognition alternatives. Further, this word-at-a-time approach requires explicit dwell-clicks to select items and this will inevitably slow the input process.

6 DISCUSSION

Dwell-free eye-typing is in theory faster than dwell-based eye-typing [14]. However, user adoption requires more than studying entry and error rates among able-bodied users in a typical text entry transcription task. Many subtleties play a large role in practical performance. The computational simulations help illuminate how some fairly fundamental design parameter choices have deep ramifications in a commercial-grade statistical decoder for dwell-free eye-typing.

The deployment study identified five user adoption barriers for dwell-free eye-typing: (1) poor integration with external applications; (2) poor integration of error correction interface; (3) an inability to fluidly combine dwell-based and dwell-free text entry; (4) lack of support for editing the final text; and (5) accuracy problems in the face of noisy eye gaze control.

6.1 Design Implications and Research Opportunities

Having identified barriers to adoption we now distill design implications based on the interviews and observations with participants, and informed by the design performance simulations where relevant. Figure 4 summarizes and links these design implications to IUI research challenges identified in a recent review [45].

6.1.1 Remove Restriction of Gazing at the Intended Key. In the current product, users were instructed to dwell-free eye-type by gazing at each intended key in sequence and that at least a single gaze point had to register on the user's intended key. We observed two issues with this restriction. First, some participants had difficulty gazing at the intended key but could quickly gaze at a nearby key. Their performance could likely be improved by instructing them to merely gaze in the vicinity of their intended key. Second, even participants with highly accurate eye control felt the current apparent restriction induced unnecessary stress, in particular since the consequence of an incorrect recognition result was very cumbersome. Relaxing the instruction to gazing in the vicinity of a nearby key should ideally be coupled with a visualization that makes users less likely to be concerned with gazing at a specific key, such as, for example, a circular semi-transparent area cursor. While this feature is supported by our statistical decoder, giving users confidence to gaze less precisely would require appropriate interface feedback and user training. Further research on different decoder parameters and different decoder architectures, such a deep neural network architectures, may result in more robust solutions that are less reliant on precise fixation locations.

6.1.2 Allow Fluid Dwell-Based and Dwell-Free Eye-Typing. Many participants expressed a desire to switch between dwell-free eye-typing and traditional dwell-based eye-typing. This is feasible to implement in the decoder as demonstrated by the computational simulations with locked letters. This letter locking approach has been shown to be effective on a smartwatch keyboard [5, 39]. We recommend this feature be included in a dwell-free keyboard. We also suggest future work investigate the many user interface design parameters that arise in such a hybrid design as their consequences are difficult to anticipate. For example, many participants were uncomfortable with the opaque and mysterious nature of dwell-free eye-typing. A hybrid design that seamlessly blends dwell-based and dwell-free eye-typing needs to be co-designed with users to ensure the final design is not only theoretically efficient but also easy to understand and behaves predictably.

6.1.3 Provide Word Predictions. All our participants were expert users of word predictions. Every single participant engaged with traditional eye-typing by typing a single letter and then scanning for a suitable word suggestion. This is logical as a rate-limited user can benefit substantially from word predictions, assuming the precision of the word prediction algorithm is high [12]. This familiar and efficient feature should be included in a dwell-free eye-typing interface. Multiple word predictions or even sentence predictions could also be added. However, care must be taken to present them in a way that is not distracting to the user and careful planning is required to ensure the screen real-estate permits the display of a suitable number of predictions. Prior work has examined such design decisions using computational experiments (e.g. [39]) and we believe a similar approach could be fruitful for this application.

Design Implication	
Remove Restriction of Gazing at the Intended Key	1 2 4 5
Allow Fluid Dwell-Based and Dwell-Free Eye-Typing	1 2 4 5 6
Provide Word Predictions	1 4
Provide Alternative Words without Requiring a Mode Switch	1 2 4
Allow Easy Fine-Tuning of Punctuation and Capitalization	1 4 5
Support Multiple Recognition Triggers	3 4 5
Provide a Tutorial and Performance Indicator	7 8
Ensure Complete Integration	6 8

IUI Research Challenge	
1	Presentation in the Interface
2	Negative Effects of Intelligence
3	Explanation
4	User Interaction
5	Technical Challenges
6	When to Integrate Intelligence
7	Trust, Confidence, Reliance and Privacy
8	Usability and Usefulness
9	Evaluation and Standards

Figure 4: A map of the design implications we identify in this work to research challenges identified in a recent review of IUI research [45].

6.1.4 Provide Alternative Words without Requiring a Mode Switch. Participants engaging well with dwell-free eye-typing often wanted to fine-tune the final text and, in the event of an error, correct it. However, no participant realized there even was a correction interface, and, when shown the correction interface, did not like that it was on a separate screen. As recognition alternatives are often the user’s intended words, we suggest displaying them in a highly visible position on the same screen as the main keyboard interface. This may be difficult to fully accommodate due to limited screen sizes on current assistive eye-tracking devices. However, given the considerable user benefit a unified text entry and correction interface may offer, it may be worth investing in upgraded hardware with a larger screen and a higher resolution. In addition, prior research has proposed word confusion network displays for speech recognition [24], handwriting recognition [20], and for mobile speech recognition interfaces on small displays [40]. We believe such approaches are promising avenues to explore to avoid a mode switch between entry and correction.

6.1.5 Allow Easy Fine-Tuning of Punctuation and Capitalization. One participant wanted to fine-tune punctuation and capitalization. Another participant felt dwell-free eye-typing would be more suitable for conversations rather than text intended to be read. Our computational experiments demonstrated repeatedly that punctuation was particularly hard to recognize correctly. Our decoder currently uses an n-gram language model that can see only a limited window of previous characters. Neural language models (e.g. GPT-2 [29] and more recent models) are capable of conditioning on a much larger window of text [7] and might better predict case due to their encoding of text into continuous features. However, even with improved language modeling, errors may still occur. These should be addressed by offering a *post-hoc* fine-tuning interface allowing words to be modified after recognition rather than specifying case and punctuation directly in their dwell-free input. Care must be taken to design such an interface given the limited screen real-estate.

6.1.6 Support Multiple Recognition Triggers. Participants in general felt dwell-free eye-typing appeared magical and unpredictable. We believe this is primarily due to the lack of agency in a recognition-based interface. When the decoder works, the system is predictable. However, when the decoder does not work the system is unpredictable and there is very little the user can do to preempt such an event. The causes of recognition errors are numerous and only some relate to the user’s behavior. Factors that are completely outside a user’s control include, for example, the quality of the language model and decoder search errors. Further, when recognition errors occur they can confuse users as the output is often nonsensical. It may be better in the case of low confidence recognition events to instruct the user to repeat their input rather than displaying likely incorrect results.

One idea proposed by several participants was to provide intermediate decoding results. While a decoder can do this (since the decoder can generate hypotheses as input is streamed into the decoder) it may be inadvisable for three reasons. First, these incomplete hypotheses may appear confusing or nonsensical as they are incomplete and only reflect the best partial result of an ongoing search process. Second, displaying intermediate hypotheses may divert a user’s visual attention, affecting their eye gaze precision. Third, it would be difficult to devise a robust selection mechanism for accepting an intermediate result that does not risk prematurely terminating input.

A more robust solution with fewer risks is to introduce flexible triggering of decoding at natural endpoints, such as when a user inputs a phrase or sentence delimiter. This may demand a careful redesign of trigger keys to avoid false activations. We studied such designs in the computational simulations in this paper and the results are promising in terms of reducing errors. Prior work [38] has studied the performance impact of allowing able-bodied users to modulate the amount of input they provide to a virtual keyboard decoder and demonstrated that such solutions are viable, although the highest performance was observed when users entered entire sentences. In another study with able-bodied users [53], multiple word input allowed faster typing but the interface had to be carefully designed to avoid additional cognitive burden. Future work is required to explore such solutions for dwell-free eye-typing and investigate their impact on performance in user studies with participants from the target audience.

6.1.7 Provide a Tutorial and Performance Indicator. The participants engaging well with dwell-free eye-typing did not fully understand how they were supposed to use it or how to use additional features, in particular error correction. In addition, the value of dwell-free eye-typing in terms of its immediate value proposition to the participants was unclear. These issues can be mitigated by introducing a playful tutorial demonstrating the principles of dwell-free eye-typing and introducing the user to all features. To further entice and engage users, performance indications such as entry rate can be shown to the user to make the value proposition clear. Prior work in the AI in education literature has considered the design of intelligent text entry tutorials [10]. An early example is the built-in playful text entry tutorial *Giraffe* in Palm Pilots [28]. Later, Kristensson and Zhai [18] introduced a text entry tutorial in the form of a game that used an expanding rehearsal interval algorithm to improve learning rates. Further work has explored games for learning sentence-decoding based text entry and gesture based text entry [3, 37]. We believe techniques from the intelligent tutoring system community could be adapted to realize effective and efficient interactive tutorials for dwell-free eye-typing.

6.1.8 Ensure Complete Integration. All participants engaging well with dwell-free eye-typing raised concerns about the current dwell-free eye-typing product having poor integration with the rest of the system. We caution against naïve integration work, such as merely sending text to an active application window, as it is important that any integration with a user's application allow fluent and easy to understand error correction and text editing of previously written text.

6.2 Limitations and Future Work

Dwell-free eye-typing is only suitable for literate users and, among these, probably only suitable in the foreseeable future to users with precise eye control or eye control that is only mildly perturbed by noise. Therefore, it cannot be considered a complete eye-tracking solution and must be complemented with other technologies, such as traditional eye-typing.

The deployment study draws its conclusions from six participants sampled from the customer database of Tobii-Dynavox. We engaged with each participant for several hours as part of an occasional support visit carried out by the company. While we would argue that such sampling and user study contexts allowed us to better understand barriers to adoption, we nevertheless acknowledge that the sample is small and cannot be considered completely representative. We hope further work can help to corroborate the findings in this paper.

The researcher involved in the study and the interpretation of the results was also deeply involved in developing the dwell-free eye-typing product. There is therefore an unavoidable bias in both the qualitative study itself and the interpretation of the results. While substantial care has been invested in carrying out the user study and interpreting the results in an as unbiased manner as possible, we nevertheless caution the reader that unavoidable bias is most likely intrinsic in the qualitative results.

For future work we see two fruitful avenues for research. The first is to carry out further *in situ* user studies to study refined iterations of a dwell-free eye-typing interface informed by the design implications in this paper. The second is to investigate technical improvements in the decoder architecture, such as choice of decoding algorithm, language modeling approach, and the possibility to adapt and personalize to a user's behavior and writing style. We hope the key findings in the deployment study and the identified design implications can guide such research.

7 CONCLUSIONS

This paper has investigated barriers to adoption of a dwell-free eye-typing product that realizes a system previously envisioned in the literature [14]. Rather than having to dwell on each letter, users can instead quickly glance through all the letters in their desired sentence. The system was deployed to users reliant on eye-typing for their everyday communication. We sampled six active users of a commercial gaze assisted communication product and identified five adoption barriers: (1) poor integration with external applications; (2) poor integration of error correction interface; (3) an inability to fluidly combine dwell-based and dwell-free text entry; (4) lack of support for editing the final text; and (5) accuracy problems in the face of noisy eye gaze control. Inspired by the results of the deployment study we carried out computational simulations to better understand the potential quantitative effects of different user interface features. By combining the best performing features we found dwell-free eye-typing may offer a character error rate below 3%. This would be accurate enough for many purposes, such as interpersonal communication. Based on these investigations, we distilled a set of design implications that can help guide future improvements to dwell-free eye-typing.

User performance of both dwell-based and dwell-free eye-typing is highly reliant on the individual. While one could focus on experiments investigating quantitative entry and error rates, both are highly variable among individuals and avoid many issues that must be solved for dwell-free eye-typing to be useful in practice. Rather, we suggest a more fruitful endeavour would be understanding and tackling adoption barriers facing eye-typing users. We hope our user study, design performance simulations, and distilled design implications will help progress dwell-free eye-typing, unlocking the potential this new interaction method may offer eye gaze communicators.

ACKNOWLEDGMENTS

This work was carried out when the authors were either employees or consultants for Tobii-Dynavox.

REFERENCES

[1] Antonia Bertolino, Pietro Braione, Guglielmo De Angelis, Luca Gazzola, Fitsum Kifetew, Leonardo Mariani, Matteo Orrù, Mauro Pezze, Roberto Pietrantuono, Stefano Russo, et al. 2021. A Survey of Field-based Testing Techniques. *ACM Computing Surveys (CSUR)* 54, 5 (2021), 1–39. https://doi.org/10.1145/3447240

[2] Parmit K Chilana, Amy J Ko, and Jacob Wobbrock. 2015. From user-centered to adoption-centered design: a case study of an HCI research innovation becoming a product. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. 1749–1758. https://doi.org/10.1145/2702123.2702412

[3] Gennaro Costagliola, Mattia De Rosa, Vittorio Fuccella, and Fabrizio Torre. 2012. TypeJump: A typing game for KeyScretch. In *2012 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*. IEEE, 241–242. https://doi.org/10.1109/VLHCC.2012.6344534

[4] Dylan Gaines, Per Ola Kristensson, and Keith Vertanen. 2021. Enhancing the composition task in text entry studies: Eliciting difficult text and improving error

rate calculation. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. 1–8. https://doi.org/10.1145/3411764.3445199

[5] Dylan Gaines, Per Ola Kristensson, and Keith Vertanen. 2021. Enhancing the Composition Task in Text Entry Studies: Eliciting Difficult Text and Improving Error Rate Calculation. 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 725, 8 pages. https://doi.org/10.1145/3411764.3445199

[6] Robert JK Jacob and Keith S Karn. 2003. Eye tracking in human-computer interaction and usability research: Ready to deliver the promises. In *The mind's eye*. Elsevier, 573–605.

[7] Urvashi Khandelwal, He He, Peng Qi, and Dan Jurafsky. 2018. Sharp Nearby, Fuzzy Far Away: How Neural Language Models Use Context. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Melbourne, Australia, 284–294. https://doi.org/10.18653/v1/P18-1027

[8] Per Ola Kristensson. 2007. *Discrete and continuous shape writing for text entry and control*. Ph. D. Dissertation. Institutionen för datavetenskap.

[9] Per Ola Kristensson. 2009. Five challenges for intelligent text entry methods. *AI Magazine* 30, 4 (2009), 85–85. https://doi.org/10.1609/aimag.v30i4.2269

[10] Per Ola Kristensson. 2011. Design dimensions of intelligent text entry tutors. In *Artificial Intelligence in Education: 15th International Conference, AIED 2011, Auckland, New Zealand, June 28–July 2011 15*. Springer, 494–496. https://doi.org/10.1007/978-3-642-21869-9_80

[11] Per Ola Kristensson. 2015. Next-generation text entry. *IEEE Annals of the History of Computing* 48, 07 (2015), 84–87. https://doi.org/10.1109/MC.2015.185

[12] 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*. 1–12. https://doi.org/10.1145/3411764.3445566

[13] Per Ola Kristensson and Keith Vertanen. 2012. Performance comparisons of phrase sets and presentation styles for text entry evaluations. In *Proceedings of the 2012 ACM international conference on Intelligent User Interfaces*. 29–32. https://doi.org/10.1145/2166966.2166972

[14] Per Ola Kristensson and Keith Vertanen. 2012. The potential of dwell-free eye-typing for fast assistive gaze communication. In *Proceedings of the symposium on eye tracking research and applications*. 241–244. https://doi.org/10.1145/2168556.2168605

[15] Per Ola Kristensson, Keith Vertanen, and Morten Mjelde. 2018. Gaze based text input systems and methods. US Patent 10,082,864.

[16] Per Ola Kristensson, Keith Vertanen, and Morten Mjelde. 2020. Gaze based text input systems and methods. US Patent 10,551,915.

[17] Per-Ola Kristensson and Shumin Zhai. 2004. SHARK2: a large vocabulary shorthand writing system for pen-based computers. In *Proceedings of the 17th annual ACM symposium on User interface software and technology*. 43–52. https://doi.org/10.1145/1029632.1029640

[18] Per Ola Kristensson and Shumin Zhai. 2007. Learning shape writing by game playing. In *CHI'07 Extended Abstracts on Human Factors in Computing Systems*. 1971–1976. https://doi.org/10.1145/1240866.1240934

[19] Andrew Kurauchi, Wenxin Feng, Ajjen Joshi, Carlos Morimoto, and Margrit Betke. 2016. EyeSwipe: Dwell-free text entry using gaze paths. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. 1952–1956. https://doi.org/10.1145/2858036.2858335

[20] Kazutaka Kurihara, Masataka Goto, Jun Ogata, and Takeo Igarashi. 2006. Speech Pen: Predictive Handwriting Based on Ambient Multimodal Recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Montréal, Québec, Canada) *(CHI '06)*. Association for Computing Machinery, New York, NY, USA, 851–860. https://doi.org/10.1145/1124772.1124897

[21] 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*. 754–755. https://doi.org/10.1145/765891.765971

[22] Päivi Majaranta, Ulla-Kaija Ahola, and Oleg Špakov. 2009. Fast gaze typing with an adjustable dwell time. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. 357–360. https://doi.org/10.1145/1518701.1518758

[23] Päivi Majaranta and Kari-Jouko Räihä. 2002. Twenty years of eye typing: systems and design issues. In *Proceedings of the 2002 symposium on Eye tracking research & applications*. 15–22. https://doi.org/10.1145/507072.507076

[24] J. Ogata and M. Goto. 2005. Speech Repair: Quick Error Correction Just by Using Selection Operation for Speech Input Interfaces. In *Proceedings of the International Conference on Spoken Language Processing* (Lisbon, Portugal). 133–136.

[25] Mukund Padmanabhan and Michael Picheny. 2002. Large-vocabulary speech recognition algorithms. *IEEE Annals of the History of Computing* 35, 04 (2002), 42–50. https://doi.org/10.1109/MC.2002.993770

[26] Diogo Pedrosa, Maria da Graça Pimentel, and Khai N Truong. 2015. Filteryedping: A dwell-free eye typing technique. In *Proceedings of the 33rd annual acm conference extended abstracts on human factors in computing systems*. 303–306. https://doi.org/10.1145/2702613.2725458

[27] Diogo Pedrosa, Maria Da Graça Pimentel, Amy Wright, and Khai N Truong. 2015. Filteryedping: Design challenges and user performance of dwell-free eye typing. *ACM Transactions on Accessible Computing (TACCESS)* 6, 1 (2015), 1–37. https://doi.org/10.1145/2724728

[28] David Pogue. 1999. *PalmPilot: the ultimate guide*. O'Reilly & Associates, Inc.

[29] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners. (2018). https://d4mucfpksywv.cloudfront.net/better-language-models/language-models.pdf

[30] Kari-Jouko Räihä. 2015. Life in the fast lane: Effect of language and calibration accuracy on the speed of text entry by gaze. In *IFIP Conference on Human-Computer Interaction*. Springer, 402–417. https://doi.org/10.1007/978-3-319-22701-6_30

[31] Daniel Rough, Keith Vertanen, and Per Ola Kristensson. 2014. An evaluation of Dasher with a high-performance language model as a gaze communication method. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*. 169–176. https://doi.org/10.1145/2598153.2598157

[32] Dario D Salvucci. 1999. Inferring intent in eye-based interfaces: tracing eye movements with process models. In *Proceedings of the SIGCHI conference on Human Factors in Computing Systems*. 254–261. https://doi.org/10.1145/302979.303055

[33] Dario D Salvucci. 2000. An interactive model-based environment for eye-movement protocol analysis and visualization. In *Proceedings of the 2000 symposium on eye tracking research & applications*. 57–63. https://doi.org/10.1145/355017.355026

[34] Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal* 27, 3 (1948), 379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x

[35] Outi Tuisku, Päivi Majaranta, Poika Isokoski, and Kari-Jouko Räihä. 2008. Now Dasher! Dash away! Longitudinal study of fast text entry by eye gaze. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. 19–26. https://doi.org/10.1145/1344471.1344476

[36] Keith Vertanen. 2021. *Probabilistic Text Entry-Case Study 3* (1 ed.). Association for Computing Machinery, New York, NY, USA, 277–320. https://doi.org/10.1145/3447404.3447420

[37] Keith Vertanen, Justin Emge, Haythem Memmi, and Per Ola Kristensson. 2014. Text blaster: a multi-player touchscreen typing game. In *CHI'14 Extended Abstracts on Human Factors in Computing Systems*. 379–382. https://doi.org/10.1145/2559206.2574802

[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*. 1–12. https://doi.org/10.1145/3173574.3174200

[39] Keith Vertanen, Dylan Gaines, Crystal Fletcher, Alex M Stanage, Robbie Watling, and Per Ola Kristensson. 2019. VelociWatch: Designing and evaluating a virtual keyboard for the input of challenging text. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–14. https://doi.org/10.1145/3290605.3300821

[40] Keith Vertanen and Per Ola Kristensson. 2009. Parakeet: A continuous speech recognition system for mobile touch-screen devices. In *Proceedings of the 14th international conference on Intelligent user interfaces*. 237–246. https://doi.org/10.1145/1502650.1502685

[41] 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*. 295–298. https://doi.org/10.1145/2037373.2037418

[42] 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 & Services* (Stockholm, Sweden) *(MobileHCI '11)*. ACM, New York, NY, USA, 295–298. https://doi.org/10.1145/2037373.2037418

[43] Keith Vertanen and Per Ola Kristensson. 2014. Complementing text entry evaluations with a composition task. *ACM Transactions on Computer-Human Interaction (TOCHI)* 21, 2 (2014), 1–33. https://doi.org/10.1145/2555691

[44] 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. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (Seoul, Republic of Korea) *(CHI '15)*. ACM, New York, NY, USA, 659–668. https://doi.org/10.1145/2702123.2702135

[45] Sarah Theres Völkel, Christina Schneegass, Malin Eiband, and Daniel Buschek. 2020. What is "intelligent" in intelligent user interfaces? a meta-analysis of 25 years of IUI. In *Proceedings of the 25th international conference on intelligent user interfaces*. 477–487. https://doi.org/10.1145/3377325.3377500

[46] David J Ward, Alan F Blackwell, and David JC MacKay. 2000. Dasher—a data entry interface using continuous gestures and language models. In *Proceedings of the 13th annual ACM symposium on User interface software and technology*. 129–137. https://doi.org/10.1145/354401.354427

[47] David J Ward and David JC MacKay. 2002. Fast hands-free writing by gaze direction. *Nature* 418, 6900 (2002), 838–838. https://doi.org/10.1038/418838a

[48] Jacob O Wobbrock, James Rubinstein, Michael W Sawyer, and Andrew T Duchowski. 2008. Longitudinal evaluation of discrete consecutive gaze gestures for text entry. In *Proceedings of the 2008 symposium on Eye tracking research & applications*. 11–18. https://doi.org/10.1145/1344471.1344475
[49] Shumin Zhai. 2003. What's in the Eyes for Attentive Input. *Commun. ACM* 46, 3 (2003), 34–39. https://doi.org/10.1145/636772.636795
[50] Shumin Zhai and Per-Ola Kristensson. 2003. Shorthand writing on stylus keyboard. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. 97–104. https://doi.org/10.1145/642611.642630
[51] Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: reimagining keyboard interaction. *Commun. ACM* 55, 9 (2012), 91–101. https://doi.org/10.1145/2330667.2330689
[52] Shumin Zhai, Per Ola Kristensson, Pengjun Gong, Michael Greiner, Shilei Allen Peng, Liang Mico Liu, and Anthony Dunnigan. 2009. Shapewriter on the iPhone: from the laboratory to the real world. In *CHI'09 Extended Abstracts on Human Factors in Computing Systems*. 2667–2670. https://doi.org/10.1145/1520340.1520380
[53] Mingrui Ray Zhang and Shumin Zhai. 2021. PhraseFlow: Designs and Empirical Studies of Phrase-Level Input. 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 197, 13 pages. https://doi.org/10.1145/3411764.3445166