EDITalk: Towards Designing Eyes-free Interactions for Mobile Word Processing

Debjyoti Ghosh[1,2], Pin Sym Foong[1], Shengdong Zhao[1], Di Chen[1], Morten Fjeld[3]

[1]NUS-HCI Lab, School of Computing,
[2]NUS Graduate School for Integrative Sciences and Engineering,
National University of Singapore, Singapore
debjyoti, pinsym@u.nus.edu; sundychen1018@gmail.com;
zhaosd@comp.nus.edu.sg

[3]t2i Lab, Chalmers University of Technology,
Gothenburg, Sweden
fjeld@chalmers.se

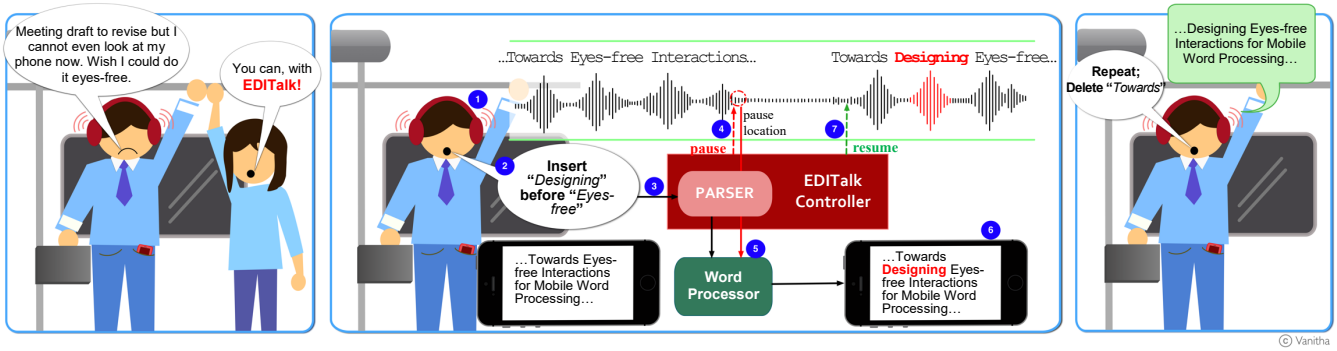

Figure 1. EDITalk allows the user to barge in (or interrupt) real time while listening to a text to facilitate eyes-free word processing. On user utterance, the system pauses real time and executes the desired user operation; system components are shown in the 2nd pane (numbers 1-7 show information flow); sample utterances and system output are shown in the 3rd pane.

ABSTRACT

We present EDITalk, a novel voice-based, eyes-free word processing interface. We used a Wizard-of-Oz elicitation study to investigate the viability of eyes-free word processing in the mobile context and to elicit user requirements for such scenarios. Results showed that meta-level operations like *highlight* and *comment*, and core operations like *insert*, *delete* and *replace* are desired by users. However, users were challenged by the lack of visual feedback and the cognitive load of remembering text while editing it. We then studied a commercial-grade dictation application and discovered serious limitations that preclude comfortable speak-to-edit interactions. We address these limitations through EDITalk's closed-loop interaction design, enabling eyes-free operation of both meta-level and core word processing operations in the mobile context. Finally, we discuss implications for the design of future mobile, voice-based, eyes-free word processing interface.

CHI 2018, April 21–26, 2018, Montreal, QC, Canada

ACM 978-1-4503-5620-6/18/04...$15.00
https://doi.org/10.1145/3173574.3173977

Author Keywords

Eyes-free interfaces; Voice-based word processing; Barge-in; Eyes-free interaction design; Conversational UI

ACM Classification Keywords

H.5.m. Information interfaces and presentation (e.g., HCI): Miscellaneous;

INTRODUCTION

Despite the limited real estate of smartphone screens, word processing[1] is a commonly performed task on smartphones. In the mobile context, eyes-free input can be desirable in many situations [19]; for example, a researcher may want to revise his paper draft while commuting when hand-holding the phone is inconvenient, thus making visual engagement and manual editing difficult (Fig. 1). However, eyes-free word processing is not well supported on mobile phones. Although speech-input is well-supported by applications like Dragon Anywhere or Google Docs, our preliminary explorations showed that most of the editing operations, such as text deletion or replacement, text formatting and annotations require visual feedback, making them unsuitable for eyes-free use. Simultaneous use of an eyes-free input and output modality, such as speech, to design a fully integrated and interactive eyes-free word processing system remains unexplored.

[1] We base our definition of "word processing" on Roberts and Moran's taxonomy of editing tasks [10] and generalize it to include any scenario where text editing tasks are performed as in social media or form interaction tasks.

Speech *output*, rather than speech *input*, has been the focus of most previous research on speech-based systems [9, 13]. As an input mechanism, speech is natural, fast [17] and can be performed eyes-free. Speech recognition is already well integrated across mobile platforms and is leveraged by popular voice-based virtual assistants (e.g., Siri, Amazon's Alexa, etc.). In their work on blind users, Azenkot *et al.* [1] explored speech as an eyes-free input modality and showed that though speech input was an efficient alternative to the on-screen keyboard with Apple's VoiceOver, users faced difficulty in reviewing and editing the speech recognizer's output and eventually fell back on using the on-screen keyboard. Conceivably, with speech input studies, the focus has been on hands-free rather than eyes-free use [4, 5, 12]. Several touch or stylus-based input interactions [6, 8, 14] have been developed in an attempt to improve error correction, but their high visual demand renders them unsuitable for eyes-free use.

Vertanen and Kristensson's work on fluid text interaction [16] aims to address error correction and revision through a one-step, voice-based technique using automatic alignment models which do not need visual feedback. Hence, this is potentially an important step towards eyes-free word processing. However, the motivation was hands-free use of the technique rather than eyes-free use. So, their work does not explore eyes-free user behavior and challenges specific to eyes-free word processing. Also, their error-correction and revision is based on user's normative turn-taking behavior and does not allow the user to barge in with the intended correction. Yet, we argue, that the ability to barge-in is crucial to eyes-free word processing to prevent sensory overload to user's short term memory owing to the linear [2, 11, 15] and temporal [2, 18] nature of audio. Further, the context of use for the fluid speech model is currently limited to single sentence utterances. The model's capabilities can be extended to a larger body of text by integrating it to a continuous-time interactive system. These factors motivated us to look at designing interactions around establishing a closed-loop, voice-based eyes-free word processing system with barge-in capabilities.

The paper details an exploratory, first-time study of speech as a simultaneous input/output modality to facilitate eyes-free word processing. In that, our focus was on establishing the feasibility and desirability of such a system [7] through a general use mobility scenario (see accompanying video). Possible specialized contexts of use such as for the visually impaired and in noisy crowded environments were considered but addressing issues specific to these scenarios needs an understanding of how users might want to use a new method of interaction. We developed EDITalk as a proof of concept to facilitate this understanding.

To explore relevant and desirable functions for eyes-free word processing, we first carried out an initial interview and a Wizard-of-Oz elicitation study [3]. Then, with the elicited user requirements, we conducted an observational study to further investigate how users attempted to complete these desired operations using a commercial-grade dictation application. Results showed that there are severe limitations of using such applications to achieve eyes-free word processing. In response, we propose EDITalk, a novel voice-based interface designed in-house to facilitate mobile eyes-free word processing (Fig. 1). Results of our usability study showed that EDITalk enables the user to achieve eyes-free word processing by (1) reducing the users' need to specifically and accurately reference text locations; (2) offering an operation method that effectively separates spoken editing commands from spoken text input; and (3) providing adequate system feedback for operations performed by the user. Our main contribution for this paper include EDITalk's novel interface, based upon our findings from the elicitation study and the observational study to understand the limitations of eyes-free interaction with text using existing dictation applications. We also contribute implications for the design of future mobile, voice-based, eyes-free word processing interfaces.

ELICITATION STUDY: PATTERNS OF SPEECH-INPUT FOR EYES-FREE WORD PROCESSING

We conducted an interview study to explore user scenarios for eyes-free word processing on mobile devices. This was followed by an elicitation study to elicit and understand user requirements in such scenarios.

Methods

Participants

We recruited 12 participants (5 females, 7 males, mean age=26.75 years) for the study. All the participants self-reported that they had prior experience of using their phone's respective virtual assistants (Siri or Google Assistant). They were also very familiar with word processing on their mobile devices. The participants were mostly from academia with the exception of 1 working professional. 8 of these interviews were conducted face-to-face and 4 over a video conference.

Procedure and Apparatus

For the interview component, each participant was asked to think of the most common scenarios in which they find themselves performing text manipulation operations on their mobile devices. After the brief interview, we asked the participants to review a piece of their self-written text[2] with the objective of trying to improve the writing, using a prototype that we had conceived for the purpose of this

[2] In our pilot studies, users found it significantly more difficult to process text about which they had no prior knowledge, while modifying text written by oneself placed lesser cognitive load and hence was more feasible to process, even without visual feedback. Hence, to maximize user elicitation, we had asked participants to send us a piece of their own academic writing (200-250 words) prior to the elicitation study.

study. The prototype constituted a notepad application (Apple *Notes*) with the participant's text pasted inside and *VoiceOver for Mac* (screen reader) for machine voice to read the text out to the participant over a pair of headphones. Also, it included a human facilitator who simulated the experience of an eyes-free word processor. The human facilitator would stop the VoiceOver once the participant started issuing any command, perform the desired operation on the text, provide necessary confirmation and resume the VoiceOver from where it was stopped. The range of operations that the participants could perform on the text included meta-level operations like markups (highlight, etc.), annotations (inline comments, etc.), contextual navigations (go to beginning of last paragraph, etc.) and any core editing operations [10]. The objects for these operations could be either sentence-level or a portion of the text (word-level / phrase-level) [10].

On completion of the study, participants were asked to fill up a post-survey questionnaire where they needed to identify the operations they found most relevant to revise their own writing. Finally, we encouraged our participants to share with us their thoughts and feedback on what components they would include in a "perfect" system, as designers or developers of such an eyes-free, voice-based word processing system.

Results and Discussion

Interview responses revealed a set of distinct user scenarios from the 12 participants (Table 1).

Word processing scenarios on smartphones	# occurrences
Revise self-written text (academic writing, meeting drafts, etc.)	11
Review text written by others (student submissions, peer reviews, etc.)	9
Active engagement with text with the purpose of information extraction (lengthy formal emails, software requirement specification documents, etc.) and decision making (schedule events, take notes, etc.)	6
Casual writing (random thoughts, ideas etc.) and revise while composing if text is longer than a few sentences	5
Revise lists (to-do, reminders, grocery list, etc.)	4
Others	3

Table 1. Scenarios obtained from participants' responses

Active engagement with text (revision, review or for information extraction) was the most frequent scenario (> 68%) as reported by our participants. When asked about the word processing operations that they usually perform in these scenarios, the answers mostly focused on *highlighting*, recording *comments* on portions of the text and performing core operations like *insert*, *delete* and *replace*. 8 of our participants mentioned that they frequently find themselves performing word processing operations for these scenarios, in situations where it is either difficult or undesirable to hand hold the phones and use it visually. The recurrent situations in their responses were: on the go scenarios like utilizing time spent on commutes or walking, and desirable eyes-free scenarios like when the eyes are tired but the brain can still handle cognitive load. Interestingly, 9 of 12 participants (75%) expressed willingness to be able to use their device hands- and eyes-free in these situations.

For each participant, we counted the number of times they performed each operation while revising their respective piece of self-written text. These operations were by and large in agreement with the same set of operations, identified by our participants, to be relevant to the eyes-free scenario. We grouped the operations into 5 categories - Markup (*highlight*, etc.), Comment, Core (*Insert*, *Delete*, *Replace*, etc.), Navigation (*Repeat Line*/*Para*, *Restart* from beginning etc.) and Others (*Pause*, *Resume*, etc.). The percentage distribution for operations in each category, recorded over all 12 participants was – Markup (18.2%), Comment (31.8%), Core (24.5%), Navigation (14.5%) and Others (11%). It was interesting to note that *meta*-level operations (Markup and Comment) were performed twice as frequently (50%) as the core processing operations (24.5%).

From the elicited user requirements, we extracted and grouped the set of pivotal operations that we found were relevant to the eyes-free interaction with text for mobile word processing (Table 2).

Group	Operation	Object
NAVIGATION	Repeat	Current Sentence Previous Sentence
	Restart	n/a
META	Highlight Comment On	Sentence, Phrase
CORE	Insert Delete Replace	Phrase

Table 2. Relevant word processing operations for eyes-free mobile scenarios

Participants were not comfortable with committing to permanent changes in the text without having visual confirmation for the changes they have caused. They would rather record the changes as comments and later apply those changes on regaining visual access to the text. This finding underscored the importance of having necessary system feedback upon completion of an operation. Also, this led us to the decision of implementing EDITalk as a system with review mode functionalities (changes made to original text can be tracked). Another recurrent user behavior was to interrupt the prototype's VoiceOver and barge in with the

intended revision operation. This was in stark contrast to waiting for specific delimiters in the text (end of sentence, end of paragraph, etc.) before suggesting an operation. We attributed this behavior to the fact that waiting would overload the user's short-term memory owing to the linear and temporal nature of audio. Hence, we decided to integrate barge-in capabilities into EDITalk's interaction design.

OBSERVATIONAL STUDY: LIMITATIONS OF USING EXISTING DICTATION APPLICATIONS FOR EYES-FREE SCENARIOS

After eliciting user requirements, we investigated the effectiveness of existing dictation applications in meeting these requirements.

Methods

Participants

We recruited 8 participants (4 females, 4 males, mean age=28.25 years) for the purpose of this study. All of them self-reported to be frequent users of word processing applications. 5 participants (62.5%) mentioned that they had prior experience in using speech recognition applications.

Procedure and Apparatus

The dictation application used for the study was a paid version of *Dragon Professional Individual for Mac 6* (henceforth, *Dragon).* Microsoft Word was used as the text editor. We chose *Dragon* for our study, as it is representative of the state-of-the-art in speech-input based interaction with the computer [20, 21]. The audio equipment used for the study was a Bose QC35 Headphone with Microphone.

For the study, we shortlisted 8 operations[3] from Table 2 to test with our participants. Each participant had to perform 8 tasks (each task testing multiple trials of an operation) (8 participants x 8 tasks). For task completion, a participant was required to listen to a distinct short paragraph (mean word count was 46.1 words) and perform all trials of the operation associated with the task, on hearing a pre-informed trigger. A sample instruction to participants was: *"You need to ask the computer to highlight the current line whenever you encounter the name of an animal, like a cat or a dog (the trigger)".* For each task, the paragraphs were excerpts from children's stories written in simple English to rule out any bias due to language complexity. The order of the tasks remained the same for all participants.

After setting up their personal voice profile, the participants were trained on the set of commands that would be required to accomplish the tasks. They were allowed to look at the screen through this training so that they could learn how the software reacts to their voice commands and build up confidence in the system. We allowed the participants to try out the voice commands for a few minutes until they could confidently perform each of the basic operations. This training session was followed by an eyes-free practice session, where we blindfolded the participants and asked them to perform the tasks. After each task, we allowed them to take a look at the screen and see the effects of their instructed operations on the text. The actual study was conducted with blindfolded participants at the end of the practice session. We blindfolded the participants to limit the difficulties of using Dragon eyes-free. This ensured that participants carried through the study to completion. Also, since Dragon is not designed for eyes-free use, we wanted to elicit the best possible results from the application for comparison with EDITalk. For each task, we calculated the *Accuracy* as, *Accuracy* = Number of correctly performed operations / Number of operations needed to complete the task.

Results and Discussion

Accuracy

Eyes-free performance using *Dragon* was understandably poor. The reported accuracy does not do justice to the system's capability as a dictation application, as it is not designed for eyes-free use. Nonetheless, we present the results to contextualize the limitations of using an existing dictation application without standard visual feedback. The accuracy (in %) for all 8 operations (min=0, max=50) was approximately normally distributed (mean=20.57, SD= 15.3). The actual values are plotted in Fig. 3.

Usability Challenges

Sentence level operations had the lowest accuracy (8.33%, 0% and 12.5% for *repeat sentence*, *repeat last sentence* and *highlight sentence* operations respectively). Word processing operations need an explicitly and accurately defined selection to work upon. Visually, users can easily identify sentence boundaries by using visual feedback to guide the selection. However, without visual feedback identifying sentence boundaries is highly challenging and requires precise recall of the words at the start and end of the sentence. This explains the low accuracy of our participants in performing sentence level operations. Phrase selection, however, was easier to perform as defining a selection in this case required users to recall only the target phrase, so accuracy for these operations were higher compared to sentence-level operations.

The low levels of accuracy alone do not adequately reflect the confusion and frustration of the participants when trying to use the application to accomplish the tasks without visual feedback. There were multiple instances of participants not being able to complete all the trials for a task as the original text had been unintentionally replaced by the participant's utterances. On analysis, the limitations (L1-L3) of using existing dictation applications were:

[3] The two *comment* operations were not chosen as *Dragon* does not feature voice commands for recording comments on text.

L1. *Lack of Location Context*: Text-to-speech (TTS) does not offer its current text location context. So, for multiple occurrences of the target phrase for a phrase-level operation, the phrase can sometimes be wrongly matched to another valid location within the text[4].

L2. *Difficulty Separating Text-Entry Utterances from Command Utterances*: The default behavior of the system upon detecting a user utterance is text entry, and only when the utterance follows a predefined syntax does the system perceive it as a command. In instances with speech recognition errors, this led the system to interpret some user utterances (originally intended as a command) as text entry, thus altering the original text. For example, either the speech recognizer failed to pick up a valid keyword for a command (henceforth, *system limitation*) or the user failed to dictate a matching phrase as the object for a desired operation (henceforth, *human-factor limitation*). Speech being temporal and linear in nature, recent information from the TTS overloads user's short-term memory and makes recall of old information difficult. As the system could not differentiate between the user's attempting to issue a command versus wanting to enter text, it made eyes-free use of the system impractical for word processing.

L3. *Lack of System Feedback*: With lack of any kind of feedback from the system for performed operations, participants were often doubtful if their commands had resulted in the correct operation. To counter this, the participants often tried to rewind the TTS to an earlier location to listen to the altered text. Coupled with the limitation of separating text entry utterance from command utterance, this led to an unintended modification in the text, resulting in confusion and frustration for the participants.

IMPLEMENTATION

Based on our insights from the elicitation and the observational study, we designed EDITalk (Figs. 1-2), a voice-based web application to facilitate mobile eyes-free word processing.

System Design

The goals of our implementation were to: (1) Effectively integrate the speech-to-text (dictation) engine and the text-to-speech (TTS) engine, so as to facilitate a complete eyes and hands-free interaction with the text; (2) Enable users to easily perform sentence level operations including meta-operations like *highlight* and *comment*; (3) Design a command structure to accommodate both *system* and *human-factor limitations;* (4) Enable adequate system feedback to confirm users on the outcome of the word processing operations performed on the text; and (5) Design a system with review mode functionalities to enable the user to track changes in the text visually, at a later point in time, once h/she regains access to a screen. The design choices we adopted to achieve these goals are as follows:

(1) Speech-to-text and TTS Integration. EDITalk's design comprises four main components: a TTS engine, an asynchronous Automatic Speech Recognition (ASR) engine, a word processor module (WP) and a controller (CTRL) module. The CTRL integrates all the components together to form a closed-loop design. The TTS facilitates an eyes-free listening to the text, while the ASR engine constantly listens in for any incoming user utterances. WP does the actual meta (*highlight*, *comment*) and core (*insert*, *delete*, *replace*) operations on the text, with instructions from the CTRL. The CTRL also handles the navigation operations (*repeat*, *restart*).

Fig. 1 (2nd pane) shows information flow amongst the different components of EDITalk upon a user utterance. The ASR listens for a user utterance and sends the recognized voice context from the utterance to the CTRL. The CTRL responds by sending a pause signal to the TTS. On receiving the pause signal, the TTS responds with the pause location and saves the location so that it can retrieve it once it resumes (to address L1). CTRL parses the previously received voice context from the ASR to determine the operation required and uses the location information sent by the TTS to determine the target range of text on which to perform the required operation. The CTRL delegates this operation to WP, which performs the required operation using the location information received from the CTRL. Once the operation completes, the controller sends a resume signal back to the TTS, which resumes reading of the text after retrieving the previously saved location. Thus, EDITalk effectively integrates the ASR and the TTS engine to facilitate a complete eyes and hands-free interaction with the text.

(2) Sentence-Level Operation. The TTS engine's location awareness enables it to send its current location of translation of the text to the controller. With this information, the controller can calculate the sentence boundaries. Since the onus of identifying sentence boundaries is no longer on the user, this design effectively neutralizes the user's cognitive load of recall.

(3) Command Structure. The ASR engine in EDITalk constantly listens in for any incoming user utterance. The system pauses real-time as the user starts to speak and performs the intended operation once a valid command is recognized. All system operations (including text entry) are enforced through their respective command syntax. The default (fail-safe) behavior of the system, in case of an unrecognized command and unmatched (or, missing) target phrase is to pause the TTS for half a second and 2 seconds respectively and resume reading from the beginning of the interrupted sentence. In the latter case, the system voices an audio prompt to the user, indicating the error. Thus,

[4] It is matched to the occurrence which is nearest to the current cursor location.

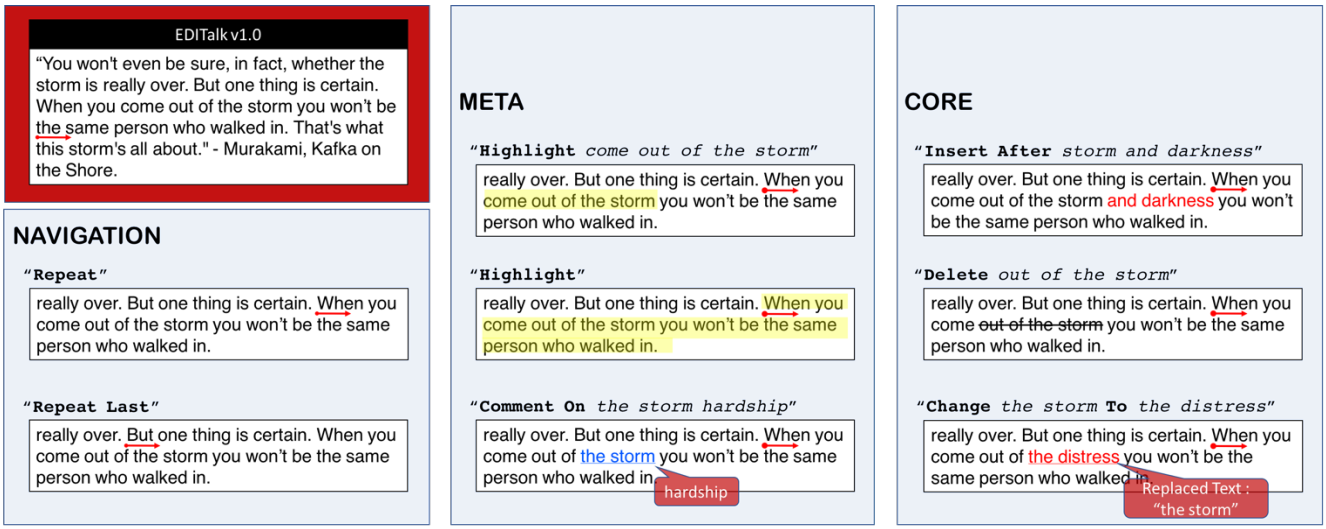

Figure 2. Schematic of EDITalk's command design. The red arrow within each box indicates the TTS location and is not part of the visual interface. The top-left corner block denotes the context and the TTS location when the user barges in. The blue blocks show sample utterances (above the individual boxes) for Meta, Core and Navigation commands. For each utterance, the context of use and resulting system action has been shown. The callouts in the Comment and Change commands are part of EDITalk's review mode functionalities and pop up on mouse over the edited text.

EDITalk's command structure was designed to overcome the limitation of existing dictation applications in accommodating *system* and *human-factor limitations* (to address L2).

Fig. 2 shows the command structure design of EDITalk. Sentence level delete (not shown in Fig. 2) and highlight are simpler than their phrase-level counterparts since the required utterance is as simple as **Highlight** or **Delete** which then triggers the system to highlight/delete the current sentence. For sentence level comment (not shown), **Comment** <*phrase*> records the phrase as a comment to the current sentence. In the elicitation study, user behavior did not exhibit nonlinear navigation within the text, owing to finite short-term memory and the linear, temporal nature of audio. To simulate user behavior to go back to the previous sentence, we designed the **Repeat Last** command that places the TTS and cursor at the start of the previous sentence. Ambiguity on the object of a sentence level operation at sentence boundaries is resolved by allowing the user an experimentally determined reaction time. Thus, even if the TTS moves on to the next sentence, the current context remains unchanged until the user expires the reaction time limit, beyond which she would need the **Repeat Last** command to rewind the context back to the previous sentence.

The other user intents supported in EDITalk's command structure are: **Restart** (rewinds the TTS location to the beginning of the text), **Repeat Para** (repeats current paragraph) and **Pause**/**Resume** (pause/resume system action to incoming user utterances) (*Navigation group*); **Note** (to append notes at the end of the document), **Insert** (inserts text at the end of the current sentence) and **Insert Before** (*Core group*); **Undo**/**Redo** (History Module). In all, a total of 19 user intents are currently supported by EDITalk's command design.

(4) System Feedback. EDITalk provides both explicit and implicit feedback for operations performed on the text (to address L3). Explicit feedback informs the user in the cases of an erroneous command structure (e.g., **Replace** <*parameter_1*>), an unmatched target phrase (e.g., **Delete** <*phrase*>, where <*phrase*> could not be found in the text), on completion of the text to speech translation and a system pause. Implicit feedback is provided when the system repeats the line containing the altered text, after a core operation.

(5) Track Changes. One of the design principles for EDITalk was to facilitate the tracking of all changes made eyes-free, once the user regains visual access to the text.

USABILITY STUDY: EVALUATION

EDITalk was designed to enable the user to achieve eyes-free word processing in the mobile context. We were interested to know: (1) Can participants complete given word processing tasks without any visual feedback, using EDITalk? (2) If they can complete the tasks, can they do it accurately? (3) Do participants require repeated attempts to complete a task trial? (4) Do participants actually find EDITalk useful for eyes-free word processing? With these goals in mind, we conducted a user study to find out how participants use EDITalk to complete a given set of word processing tasks, without visual feedback in a mobile scenario.

Study Design

Participants

We recruited 12 participants (6 females, 6 males, mean age=27.67 years, SD=6.27 years, P1-P12) from within the university community. All of them self-reported to be frequent users of word processing applications. 4 participants (~ 33%) self-reported as having high voice-input usage experience, while 8 participants (~ 66%) self-reported as having low voice-input usage experience. 3 were native English speakers and all participants were fluent in English at university level.

Procedure and Apparatus

We began by training our participants on how to use EDITalk to perform navigation (*repeat*, *restart*), meta-level operations (*highlight*, *comment*), core operations (*insert*, *delete*, *replace*) and other operations like *pause* and *resume*. This familiarized them with the system as well as accustomed them to the audio from TTS and also to the act of dictating to the system. We then conducted a practice exercise with our participants. They were given a set of 10 tasks each testing a word processing operation. We tested the participants only on the most recurrent user operations on text as observed through our elicitation study (Table 2).

For the practice, the participants were asked to walk around in a low-noise lab environment while listening to the text. This precluded the possibility of any visual feedback of the text and simulated a mobile scenario. They were equipped with a pair of Bose QC35 noise cancellation headphones (with microphone). Thus, their interaction with the text was completely eyes and hands-free.

After the practice session, participants were asked to perform the same set of tasks as in the practice, under the same set of constraints of a mobile eyes-free environment. The order of the tasks was the same for all the participants. Each task was associated with a distinct short paragraph (mean word count=48, SD=8.63). The paragraphs for the actual study were different from the paragraphs used for the practice session. The participants were asked to listen to the paragraph associated with a given task and perform the instructed word processing operation upon hearing a pre-informed trigger. A sample instruction to a participant looked like: "You need to ask the computer to highlight the current line whenever you encounter the name of an animal, like a *cat* or a *dog* (the trigger)". For each task, the paragraphs were excerpts from children's stories written in simple English to rule out any bias due to language complexity.

Each paragraph had at least one and at most three triggers. The participants had no prior knowledge of the number of triggers. Hence, they were obliged to listen through to the end of the text. At the end of each task, we allowed the participants to see the outcome.

Outcome Measures

1) *Accuracy* = Number of correctly performed operations / Number of operations needed to complete the task.

2) Utterance Precision of the participants for each task as, *Utterance Precision* = Number of utterances needed to complete the task / Total number of utterances made by the participant.

3) Perceived Utility Score (7-point Likert scale, where 1 is strongly disagree and 7 is strongly agree).

Finally, we encouraged the participants to share feedback on their experience of using the system.

Results and Discussion

Accuracy and Precision

The participants could easily complete all the tasks under the constraints of an eyes-free mobile scenario, with high levels of accuracy and precision. Fig. 3 shows accuracy and utterance precision for all tasks, averaged across all the participants.

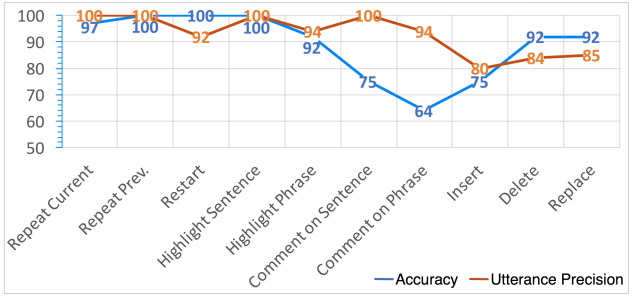

Figure 3. Accuracy and Utterance Precision (in %) for all tasks, averaged across 12 participants.

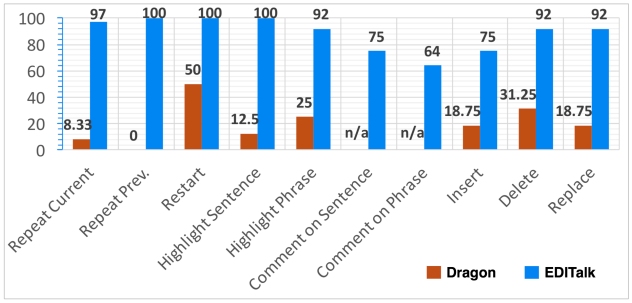

Figure 4. Accuracy comparison between Dragon and EDITalk, for all operations except *comment* operation on phrase and sentence objects.

Fig. 4 shows an accuracy comparison chart between Dragon Professional and EDITalk. This chart should however be interpreted as a way to contextualize that EDITalk addresses the limitations of using existing dictation applications for eyes-free word processing and enables such operations in the mobile context.

Qualitative User Assessment

10 out of our 12 participants (> 83%) mentioned that they were happy with the system and found it quite useful and intuitive as it provides a new way of interaction with text. P9, who had some prior experience with dictation applications said, *"(I) loved the system. It was really easy and fun to edit text on the go"*, while P11 who is an infrequent user of his phone's virtual assistant said that he was pleasantly surprised having used such a system and found it to be *"very intuitive and very helpful"*. P5 believed our prototype to have *"a lot of potential going forward"*. The enthusiasm with which the participants received EDITalk was echoed in their Likert scale (7-point scale, with 1=strongly disagree, 7=strongly agree) responses to 3 closed-ended questions in the post-study questionnaire - (a) *I found the system useful in achieving eyes-free word processing.* (mean=5.92, SD=0.67); (b) *I could be sure that what I had instructed was what was actually performed.* (mean=5.58, SD=0.67); (c) *Amount of confirmation/feedback from the system was apt.* (mean=5.17, SD=1.03).

Analysis of Error Source

Almost 50% of participants faced difficulties in getting the ASR engine to recognize the *insert* and *comment* keywords. For most of them, *insert* was misrecognized as "inside" and *comment* as "commence". This accounts for the lower accuracy levels for the two operations. A few participants failed to identify the target phrase to the system, resulting in a failed task attempt. P7 tried several times to delete the phrase "was sitting" but each time his phrase was misrecognized as "was setting". As a result, he received a "Phrase not found" feedback from the system each time. It was confusing for him since when the TTS repeated the sentence back to him, he could clearly hear "was sitting" but on trying to delete it failed every time. The problem can be attributed to his pronunciation of the word "sitting", but it sheds light on a bigger problem - inability of the system to disambiguate between homophones.

P7 and P12 mentioned that they found the amount of feedback from the system to be inadequate. This was reflected by their Likert scale response (4 and 3 respectively) to the statement: "*Amount of confirmation/feedback from the system was apt.*" P12 mentioned that she would prefer to have confirmation for *highlight* and *comment* operations.

It was interesting to note that while *comment* operations had lower accuracy levels, they had very high precision, whereas core operations like *replace* and *delete* had higher accuracy levels than precision levels. We found this to be related to the differences in feedback between the meta and core operations. Since meta operations had no explicit feedback to confirm whether the given task trial was performed successfully, the participants did not make repeated attempts to correct failed attempts at a task trial. This resulted in high precision but low accuracy. However, since core operations change the text content, the participants had implicit feedback of whether the operation resulted in a success. Hence, they made repeated attempts at a failed task trial, resulting in a high accuracy but low precision.

LIMITATIONS AND FUTURE WORK

As a proof of concept prototype, EDITalk does not provide all the functionalities needed to overcome all identified challenges of mobile eyes-free interaction with text. Currently, the system lacks in its ability to provide adequate feedback for *highlight* and *comment* operations. Also, it does not facilitate users to listen to their recorded comments on the text. Also, as a proof of concept, EDITalk was designed for the general use mobility scenario. Specific solutions targeted to specific user groups such as the visually impaired is a logical next step.

EDITalk's trigger-based study design precludes certain user behaviors that may occur in real-world scenarios. A few such possible scenarios might be: (i) the user is thinking while editing, hence, mumbling or faltering, or overriding old utterances with newer ones, unsure of which one to finalize as the intended edit; (ii) the user utterance contains non-lexical sounds like *umm*, *uh-huh*, *erm*; (iii) the user commits an error while uttering the target phrase (in sample utterance, "Delete *X*", *X* is the target phrase). While EDITalk tries to address these issues through its fail-safe features, ensuring system stability in all such scenarios, users still need to adapt to the system. For example, if the users are unsure of what exact change they want to effect in the text and falter in their utterance, it is more viable to record the change as a comment, since in absence of interactions specific to handling unclear utterances, EDITalk's fail-safe would reject the utterance and start over from the beginning of the interrupted sentence. Further, users would need to be able to utter the target phrase correctly. Failure to do so would demand more attempts to achieve the intended operation and reduce utterance precision.

Our current system design does not take advantage of natural cues like prosodic variations (a faltering utterance might be dragged or low amplitude) present in the user utterance. Future research can leverage such information to better guide the system to pick up the right user intention and improve utterance precision and accuracy.

Another real-world challenge would be to address potential issues arising from ambient noise interference and social acceptability of *EDITalking* in public. Currently, the success of our system in high ambient noise environments depends upon the noise-cancellation capability of the microphone used. Integrating ambient noise cancellation in EDITalk's software would provide for a more reliable and consistent performance across different users and environments. In addition, future work can leverage participant feedback on desired new features. Participants

(n=2) indicated that the addition of an augmented device (e.g. ring) to support navigation would be good to have. P12 stated that it would be useful to have a system that can disambiguate between homophones.

The current research should be extended to leverage the full power of conversation in future systems. We suggest, (1) exploring the use of prosody in voice to distinguish between the original text and system feedback for meta operations like *highlight* or *comment*; (2) designing interaction for TTS to read back comments recorded on the text; (3) exploring the use of augmented devices to facilitate simple operations on the text, to improve usability; (4) fitting in an automatic alignment model [16] in EDITalk's design framework, and (5) devising a way to distinguish between homophones. Most of these extensions can be designed with additional engineering work.

CONCLUSION

EDITalk is designed to bridge the gap between the user's need for mobile eyes-free interaction with text and limitations of using existing dictation applications in trying to achieve it. Results show that EDITalk's interaction design enables the user to achieve eyes-free word processing with high accuracy and precision levels. The prototype garnered positive feedback from all of our participants and demonstrated promising potential to design targeted solutions for specific user groups such as the visually impaired. We believe EDITalk has real potential to spur future work in the space of designing specialized eyes-free text interaction systems. This would be another step towards an exciting future in the conversational paradigm of user interactions.

ACKNOWLEDGEMENTS

This research was funded by the NExT research centre, supported by the National Research Foundation, Prime Minister's Office, Singapore under its IRC@SG Funding Initiative and the Wallenberg Autonomous Systems and Software Program (WASP Sweden). We thank Philippa Beckman and Erik Tobin for proofreading, and Vanitha Selvarajan for her generous help with designing Fig. 1.

REFERENCES

1. Shiri Azenkot and Nicole B. Lee. 2013. Exploring the use of speech input by blind people on mobile devices. In *Proceedings of the 15th International ACM SIGACCESS Conference on Computers and Accessibility - ASSETS '13*, 1–8. https://doi.org/10.1145/2513383.2513440
2. Keith Brown, J. Lai, and N. Yankelovich. 2006. Speech Interface Design. In *Encyclopedia of Language & Linguistics*. 764–770. https://doi.org/10.1016/B0-08-044854-2/00920-2
3. N. Dahlbäck, A. Jönsson, and L. Ahrenberg. 1993. Wizard of Oz studies - why and how. *Knowledge-Based Systems* 6, 4: 258–266. https://doi.org/10.1016/0950-7051(93)90017-N
4. CA Halverson, DB Horn, CM Karat, and J Karat. 1999. The Beauty of Errors: Patterns of Error Correction in Desktop Speech Systems. *INTERACT*, pp. 133-140.
5. Sk Kane, Jo Wobbrock, and Re Ladner. 2011. Usable gestures for blind people: understanding preference and performance. *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*: 413–422. https://doi.org/10.1145/1978942.1979001
6. TB Martin. 1980. Practical speech recognizers and some performance effectiveness parameters. *Trends in speech recognition*. Retrieved September 9, 2017 from http://ci.nii.ac.jp/naid/10020895638/
7. Donald A Norman. 2004. *Emotional Design: Why We Love (or Hate) Everyday Things*. https://doi.org/10.1111/j.1537-4726.2004.133_10.x
8. Sharon Oviatt. 2000. Taming recognition Errors with a Multimodal Interface. *COMMUNICATIONS OF THE ACM* 43, 9: 45–51. https://doi.org/10.1145/348941.348979
9. Ian J. Pitt and Alistair D N Edwards. 1996. Improving the usability of speech-based interfaces for blind users. *Annual ACM Conference on Assistive Technologies, Proceedings*: 124–130. https://doi.org/10.1145/228347.228367
10. TL Roberts and TP Moran. 1983. The evaluation of text editors: methodology and empirical results. *Communications of the ACM* 26, April: 265–283. https://doi.org/10.1145/2163.2164
11. J Schalkwyk, D Beeferman, F Beaufays, and B Byrne. 2010. "Your word is my command": Google search by voice: a case study. *Advances in Speech*. Retrieved September 1, 2017 from https://doi.org/10.1007/978-1-4419-5951-5_4
12. A Sears, CM Karat, K Oseitutu, and A Karimullah. 2001. Productivity, satisfaction, and interaction strategies of individuals with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the*. Retrieved September 9, 2017 from https://doi.org/10.1007/s102090100001

13. A Stent, A Syrdal, and T Mishra. 2011. On the intelligibility of fast synthesized speech for individuals with early-onset blindness. *The proceedings of the 13th international*. Retrieved September 9, 2017 from http://dl.acm.org/citation.cfm?id=2049574

14. Bernhard Suhm, Brad Myers, and Alex Waibel. 2001. Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction* 8, 1: 60–98. https://doi.org/10.1145/371127.371166

15. M Turunen, J Hakulinen, and N Rajput. 2012. Evaluation of mobile and pervasive speech applications. *Speech in Mobile and*. Retrieved September 1, 2017 from https://doi.org/10.1002/9781119961710.ch8

16. Keith Vertanen and Per Ola Kristensson. 2009. Automatic selection of recognition errors by respeaking the intended text. In *Proceedings of the 2009 IEEE Workshop on Automatic Speech Recognition and Understanding, ASRU 2009*, 130–135. https://doi.org/10.1109/ASRU.2009.5373347

17. J. R. Williams. 1998. Guidelines for the Use of Multimedia in Instruction. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting* 42, 20: 1447–1451. https://doi.org/10.1177/154193129804202019

18. Nicole Yankelovich and Jennifer Lai. 1999. Designing speech user interfaces. In *CHI '99 extended abstracts on Human factors in computing systems - CHI '99*, 124. https://doi.org/10.1145/632716.632793

19. Bo Yi, Xiang Cao, Morten Fjeld, and Shengdong Zhao. 2012. Exploring user motivations for eyes-free interaction on mobile devices. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems CHI 12* c: 2789. https://doi.org/10.1145/2207676.2208678

20. The Best Voice Recognition Software of 2017 | Top Ten Reviews. Retrieved September 20, 2017 from http://www.toptenreviews.com/business/software/best-voice-recognition-software/

21. The best voice recognition software of 2017 | TechRadar. Retrieved September 20, 2017 from http://www.techradar.com/news/the-best-voice-recognition-software-of-2017