Using Confidence Scores to Improve Eyes-free Detection of Speech Recognition Errors

Sadia Nowrin
Michigan Technological University
Houghton, MI, USA
snowrin@mtu.edu

Keith Vertanen
Michigan Technological University
Houghton, MI, USA
vertanen@mtu.edu

Abstract

Conversational systems rely heavily on speech recognition to interpret and respond to user commands and queries. Despite progress on speech recognition accuracy, errors may still sometimes occur and can significantly affect the end-user utility of such systems. While visual feedback can help detect errors, it may not always be practical, especially for people who are blind or low-vision. In this study, we investigate ways to improve error detection by manipulating the audio output of the transcribed text based on the recognizer's confidence level in its result. Our findings show that selectively slowing down the audio when the recognizer exhibited uncertainty led to a 12% relative increase in participants' ability to detect errors compared to uniformly slowing the audio. It also reduced the time it took participants to listen to the recognition result and decide if there was an error by 11%.

CCS Concepts

• **Human-centered computing** → **Empirical studies in HCI**.

Keywords

Voice user interfaces, error correction, speech recognition, text-to-speech (TTS), eyes free input

ACM Reference Format:
Sadia Nowrin and Keith Vertanen. 2025. Using Confidence Scores to Improve Eyes-free Detection of Speech Recognition Errors. In *The PErvasive Technologies Related to Assistive Environments (PETRA '25), June 25–27, 2025, Corfu Island, Greece.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3733155.3734896

1 Introduction

In recent years, there have been notable advancements in Automatic Speech Recognition (ASR) technology, enabling eyes-free interaction [8, 10, 15, 29] and improving accessibility for devices without a visual display (e.g. Amazon Echo, Google Home). Speech recognition can help make interfaces accessible for individuals with motor impairments [5, 18, 19, 22, 30] as well as those who are blind [2]. While deep learning models have advanced ASR accuracy [3, 17], real-world ASR performance is often negatively impacted by background noise, speaker variations, and speaker disfluencies [12]. Despite efforts to improve recognition accuracy in noisy environments using large language models [31, 32], only a modest relative improvement of 5.7% was obtained [31]. Azenkot and Lee [2] observed that blind users spent a significant amount of time correcting errors when performing speech dictation tasks. Speech error correction involves a two-step process: 1) detecting errors, and 2) correcting errors [28]. To fully realize the potential of speech recognition technology, accurate error detection and correction are crucial. In this paper, we focus on the first step, error detection.

1.1 Related work

Numerous prior studies have investigated using visual feedback to represent a speech recognizer's confidence in its result [4, 9, 21, 23–25, 27]. However, visual feedback may not be possible for individuals with visual impairments, for sighted users in situations in which they cannot visually attend to their device, or when using a device without a screen such as a smart speaker. Identifying errors in conversational systems without visual feedback can be challenging for a variety of reasons. Firstly, text-to-speech (TTS) audio can be hard to understand, especially when errors involve short or similar-sounding words [6]. In a study with sighted users [13], participants missed approximately 50% of recognition errors when the TTS audio was played at a rate of 200 words per minute (wpm). Understanding TTS can be even more difficult in noisy environments [7]. Finally, errors may occur infrequently, lulling users into trusting the recognizer [20].

Beyond sighted users, blind individuals also face challenges with detecting recognition errors through audio-only feedback. Hong et al. [14] found no significant difference in ASR error identification between blind users (42%) and sighted users (38%). This was despite blind users' extensive experience with synthesized speech. This suggests that experience with synthesized speech alone may not be sufficient for improving error detection, highlighting the need to explore alternative approaches to enhance audio-based feedback.

In this study, we examine how users can detect speech recognition errors through audio-only feedback. Similar to Hong and Findlater [13], we investigate the impact of various TTS manipulations on users' ability to detect ASR errors. Hong and Findlater found error detection improved when TTS was delivered at 200 wpm, or even slower at 100 wpm, compared to a higher speech rate of 300 wpm. In comparison to past work, we investigate adjusting the audio feedback using the speech recognizer's *confidence score*. The confidence score indicates how certain the ASR system is about the accuracy of its result [11]. Additionally, we investigate the error detection in both *common phrases* where all words were in-vocabulary, and *challenging phrases* where at least one word was out-of-vocabulary (e.g. acronyms, proper names).

PETRA '25, Corfu Island, Greece

ACM ISBN 979-8-4007-1402-3/25/06
https://doi.org/10.1145/3733155.3734896

We studied the effect of confidence scores on eyes-free error detection by testing four audio annotations: default speech rate, slow speech rate, slow speech rate for low confidence recognitions, and playing a beep for low confidence recognitions. We found slowing down the TTS audio based on the confidence score led to 85% accurate error detection, outperforming uniform slowing of the audio which had a detection accuracy of 76%. Despite the increased audio length resulting from the selective slowing according to the confidence score, the participants only experienced a slight 7% increase in the time it took them to review the recognition results compared to the default speech rate condition.

2 User Study

The goal of the user study was to investigate whether modulating the audio presentation of the speech recognition results based on a recognizer's confidence score could improve the ability of users to detect errors.

2.1 Participants

We recruited 48 participants (15 female, 32 male) aged from 21–68 (mean=36, sd=11.5) via Amazon's Mechanical Turk, an online crowdsourcing platform. We opted for an online study in order to to ensure participant safety during the COVID-19 pandemic. Participants were compensated at a rate of $10 (USD) per hour. Participants completed the experiment in 27 minutes on average. All participants self-reported being native English Speakers. 67% of the participants agreed that they frequently used speech interfaces with 22% agreeing that computers had difficulty understanding their speech.

At the end of the study, participants rated various statements using a 7-point scale with one denoting strongly disagree and seven denoting strongly agree. Statements included how easy it was to identify recognition errors in each of the four audio annotation conditions and whether they could anticipate which sentences would likely have errors. See Appendix A for our exact questionnaires.

2.2 Study Design

We employed a within-subject experimental design with four counterbalanced conditions:

- **AllNormal** — The recognition result was synthesized into speech and played at 200 wpm. This is similar to the default speaking rate of commercial TTS systems.
- **AllSlow** — The result was played at 70% of the default speaking rate, equivalent to 140 wpm. This was somewhat faster than the 100 wpm used by Hong and Findlater [13]. We selected 140 wpm as a compromise between slowing the speech to help users spot errors and avoiding excessive listening time.
- **UncertainSlow** — If the confidence score was below a threshold, the result was played at 140 wpm.
- **UncertainBeep** — If the confidence score was below a threshold, a one-second beep tone was played at the beginning followed by the result played at the default speaking rate of 200 wpm.

In the UncertainSlow and UncertainBeep conditions, we used a *confidence threshold* to determine whether to slow the TTS audio or add a beep. To establish this threshold, we conducted a pilot study with 12 participants. The pilot was conducted similar to our main study but used an initial guess for the threshold. We recognized the 480 utterances collected during the pilot using Google's speech-to-text service.[1] We tested different thresholds measuring: 1) the true positive rate (TPR), the proportion of utterances containing one more or recognition errors that were correctly identified as having an error, and 2) the false positive rate (FPR), the proportion of utterances with no errors that were incorrectly identified as having an error. We evaluated the trade-off between the TPR and the FPR at different thresholds with a receiver operating characteristic (ROC) curve. Based on this analysis, we selected a threshold of 0.93, aiming to balance sensitivity (0.85) and specificity (0.75) to detect a high percentage of errors while avoiding too many false positives.

2.3 Procedure

Using a web application, participants first signed a consent form and completed a demographic questionnaire. Participants then read a set of instructions and completed two practice tasks to familiarize themselves with the task. The audio was played at the default speaking rate in the practice trial. At the start of each condition, we provided participants with a description of how the audio annotation worked for the current condition.

Participants recorded a sentence for each task which was transcribed by Google's speech-to-text service and then synthesized via Google's TTS service.[2] Speech Synthesis Markup Language (SSML) was used in the TTS request to slow the speech rate or add a beep. Following a delay caused by the speech-to-text and TTS processing (averaging around four seconds), participants listened to the audio of the recognition result which was generated using a female voice (en-US). Participants were allowed to play the audio only once.

After listening to the result, we asked participants if the reference sentence matched the audio. If they answered no, indicating a speech recognition error, we asked them to locate the incorrect or missing words, as well as any incorrect additional words that may have appeared between two words in the reference sentence (Figure 1). Participants could only detect errors after the audio finished playing, simulating a real-world scenario where users cannot interrupt the system to correct errors during the initial playback. Finally, participants completed a final questionnaire about their experience in each condition and the study as a whole.

We selected phrases from a collection of 407 Twitter phrases [26]. This set included 194 common phrases containing all in-vocabulary words and 213 challenging phrases containing at least one out-of-vocabulary word. Out-of-vocabulary words were those not appearing in a list of 100,000 frequent English words. Challenging phrases included proper nouns and abbreviations that might be difficult for the speech recognizer. We used phrases with 5–10 words. Participants were randomly assigned 40 phrases. Each condition included five common and five challenging phrases that were presented in random order.

3 Results

In total, we collected 1,920 utterances. Google's recognizer had a word error rate (WER) of 15% on these utterances. Our analysis

[1] https://cloud.google.com/speech-to-text
[2] https://cloud.google.com/text-to-speech

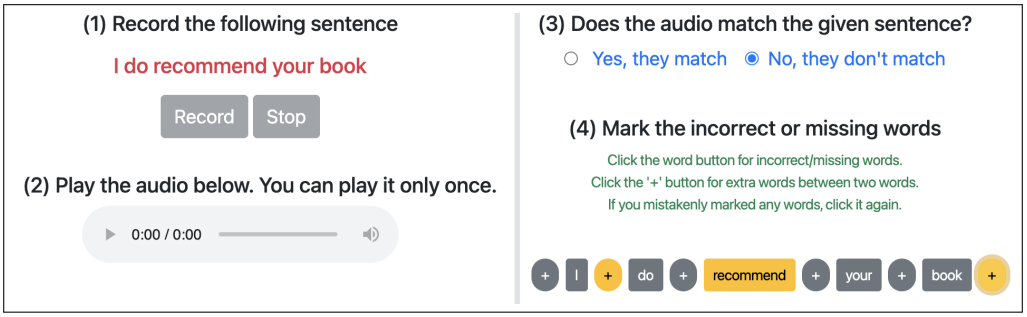
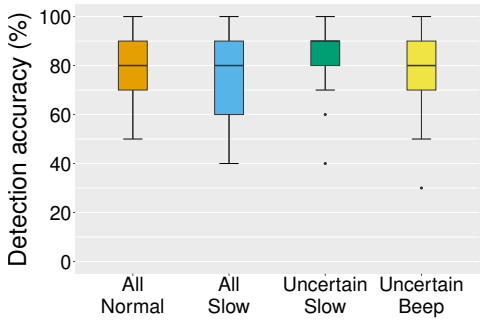
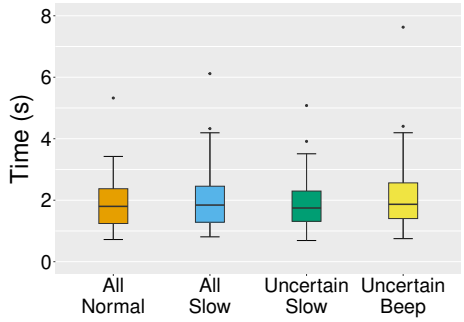


Figure 1: Screenshot of our web application. In part (1), the user records a reference sentence, in this case "I do recommend your book". In part (2), after recognition, a control appears allowing playback of the recognition result. In this case the recognition was "I really do command your book group". In part (3), the user specifies if there were any recognition errors. If they answer "No", part (4) shows buttons for each word in the reference sentence as well as plus buttons between all words. The word buttons allow the user to specify a word was recognized incorrectly or was missing in the result. The plus buttons allow the user to specify if extra words were recognized in between reference words. Buttons toggled on are highlighted in yellow.



(a) Accuracy detecting the presence of errors



(b) Detection Time

Figure 2: Comparison of the detection accuracy (top) and the detection time (bottom) in the four different study conditions.

includes two key measures: the accuracy of the error detection and the detection time. We conducted a one-way repeated measures ANOVA to compare the four conditions. In cases where the normality assumptions were violated (Shapiro-Wilk test, $p < 0.05$), we employed the non-parametric aligned rank test (ART). We used the Wilcoxon signed-rank test to compare the WER between the common phrases and the challenging phrases.

3.1 Error Detection Accuracy

We calculated how often users correctly determined whether the recognition result contained any errors (i.e. by selecting yes or no after hearing the audio). As shown in Figure 2a, the proportion of correct error detection was higher in UncertainSlow (85%) compared to AllNormal (80%), AllSlow (76%), and UncertainBeep (79%). A non-parametric ART test revealed a significant difference ($F_{3,141} = 4.48$, $\eta_p^2 = 0.087$, $p = 0.005$). Post-hoc pairwise comparisons with Bonferroni correction found a significant difference between the UncertainSlow and AllSlow conditions ($p = 0.002$). However, no significant differences were observed between UncertainSlow and AllNormal ($p = 0.1$), UncertainSlow and UncertainBeep ($p = 0.2$), AllNormal and AllSlow ($p = 0.9$), AllNormal and UncertainBeep ($p = 1.0$), or AllSlow and UncertainBeep ($p = 1.$).

In contrast to the previous study by Hong and Findlater [13] that reported improved error detection with a slow speech rate, our study did not find a significant difference in error detection between other pairs. However, our results suggest that slowing down the audio playback only when necessary might help users to better detect the presence of errors compared to uniformly slowing down the playback. As shown in Figure 2a, the variance in per-participant error detection performance was smaller in the UncertainSlow ($SD = 11.5$) condition compared to AllNormal ($SD = 12.9$), AllSlow ($SD = 15.8$), and UncertainBeep ($SD = 16.6$). This may indicate the confidence score-based slowing helped some users avoid substantially lower accuracy compared to the average.

Unfortunately, we lacked sufficient data to reliably analyze the impact of different audio annotations on participants' ability to identify the specific locations of the errors. This was due to not every participant experiencing a sufficient number of recognition errors in each condition. However, we were did conduct some analysis by aggregating errors across all conditions. We found users correctly located 49% of all errors. Broken down by type of recognition error, they located 2% of insertion errors, 49% of substitution errors, and 62% of deletion errors. Actual substitution and deletion errors were

identified by aligning the reference and recognition transcripts using the Levenshtein distance algorithm [16]. We determined actual insertion errors by manual review.

Across all conditions, the ratio of locating errors was 48% for challenging phrases and 52% for common phrases. This indicates participants missed nearly half of the errors in the transcribed text and this was not strongly influenced by the difficultly of the sentence. In particular, participants struggled to identify insertions, suggesting that detecting and correcting added words may necessitate greater attention.

In our study, participants were presented with both common and challenging phrases. The WER was significantly higher at 17% for challenging phrases compared to 12% for common phrases (r = -0.97, $p < 0.001$). This suggests that our approach of using challenging phrases to elicit more recognition errors was effective.

For common phrases, the proportion of correct error detection was higher in UncertainSlow (90.9%) compared to AllNormal (84.7%), AllSlow (83.5%), and UncertainBeep (82.5%). A non-parametric ART test revealed a significant difference ($F_{3,141} = 9.40$, $\eta_p^2 = 0.167$, $p < 0.001$). Post-hoc pairwise comparisons with Bonferroni correction found significant differences between AllSlow and UncertainSlow ($p = 0.0003$), and between AllSlow and AllNormal ($p = 0.0001$). No significant differences were observed between UncertainSlow and UncertainBeep ($p = 1.0$) or between AllNormal and UncertainBeep ($p = 1.0$).

For challenging phrases, the proportion of correct error detection was also higher in UncertainSlow (79.9%) compared to UncertainBeep (75.9%), AllNormal (75.6%), and AllSlow (68.5%). An ART test revealed a significant difference ($F_{3,141} = 11.52$, $\eta_p^2 = 0.197$, $p < 0.001$). Post-hoc pairwise comparisons with Bonferroni correction found significant differences between AllNormal and AllSlow ($p = 0.0002$), and between AllNormal and UncertainBeep ($p = 0.0003$). No significant differences were observed between UncertainSlow and UncertainBeep ($p = 0.2$), or between UncertainSlow and AllNormal ($p = 0.07$).

Comparing results between the common and challenging phrases, it is evident that error detection accuracy decreases for challenging phrases. However, the relative advantages of UncertainSlow remained consistent and had the highest detection accuracy across both common (90.9%) and challenging (79.9%) phrases. This suggests that selectively slowing playback as done in the UncertainSlow condition effectively supports error detection across varying task difficulties.

3.2 Detection Time

We measured the *detection time* from the end of the audio playback to the participant's response of yes or no. Average detection times were similar: 1.98 seconds in AllNormal, 2.03 seconds in All Slow, 1.86 seconds in Uncertain Slow, and 2.26 seconds in Uncertain Beep condition (Figure 2b). These differences were not significant ($F_{3,141} = 0.69$, $\eta_p^2 = 0.014$, $p = 0.56$).

We also calculated the *total time* it took participants to listen to the audio and to respond yes or no. The total time was 4.52 s in AllNormal, 4.85 s in UncertainSlow, 5.43 s in AllSlow, and 5.42 s in UncertainBeep. As expected given the similar detection times we observed between conditions, playing at the normal rate with no beep was the fastest. Always slowing the playback or adding the beep for uncertain recognitions resulted in a 20% increase in total time compared to normal playback. Using a slower speaking rate for uncertain results instead of the beep was more time efficient, resulting in only a 7% increase in total time compared to normal playback and an 11% decrease compared to always slowing down the audio.

3.3 Subjective Feedback

Participants rated how easy it was to identify errors under four conditions on a 7-point Likert scale (1=strongly disagree, 7=strongly agree). As shown in Figure 3, 94% of participants found AllNormal easy, followed closely by 92% for AllSlow and UncertainBeep, and 86% for UncertainSlow. The Friedman test indicated no significant difference in participants' ratings across the four conditions ($\chi^2(3) = 2.94$, $p = 0.40$).

Participants also rated whether they could anticipate when a sentence was likely to result in a recognition error. A majority felt they could anticipate errors based just on the sentence with 62% expressing some level of agreement (slightly agree, agree, or strongly agree). In contrast, 24% of participants expressed some level of disagreement (slightly disagree, disagree, or strongly disagree). The remaining 14% were neutral.

4 Discussion

In our study, we used four audio annotations to assess participants' ability to detect errors in their transcribed speech. Our result suggests that using the recognizer's confidence in its results to change how we present the result audio can help users detect errors.

One limitation of our study is we only considered native English speakers who were sighted. Blind users, for example, may have more experience listening to TTS, which could impact their ability to detect errors in TTS audio. Additionally, non-native speakers with ascents may have different experiences with speech recognition technology. Future research should explore how diverse populations detect errors when using speech recognition.

We do not know the environment our crowdsourced participants completed our study, but it is likely many were in a quiet environment. In real-world use, speech recognition users may be exposed to various types of noise and distractions that could affect their ability to detect errors. Moreover, users may be engaged in other tasks while using voice assistants (e.g. driving or exercising), which could also affect their ability to detect errors. Future studies could investigate how different contexts impact users' ability to detect speech recognition errors.

Our evaluation used a single static confidence threshold determined by our pilot testing. Instead, a system could try and dynamically adjust a user's threshold based on observing their interactions with previous recognition results. For example, if a result was above the threshold but the user corrected it, this may signal that a lower threshold is needed.

To create a realistic task, we had participants record themselves speaking sentences and used a state-of-the-art speech recognizer to present their actual transcription results. While we could have artificially forced recognition errors, we wanted to study the impact

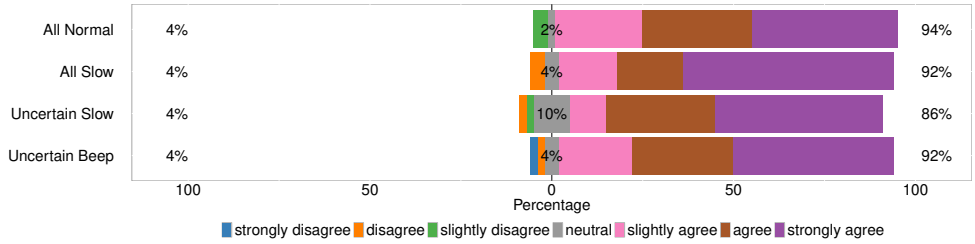


Figure 3: Participant feedback on how easy it was to identify errors in each condition. The percentages on the left are the portion of participants who strongly disagreed, disagreed, or slightly disagreed with the statements. The percentages in the middle correspond to the portion who were neutral. The percentages on the right correspond to those who strongly agreed, agreed, or slightly agreed.

of our interventions in a more externally valid task with accurate speech recognition, but with some phrases containing difficult words (as might happen in real-world use). However, this resulted in our data lacking sufficient recognition errors for each participant and condition for fine-grained analysis of users' ability to locate errors. We suggest future work consider additional ways to ensure sufficient errors in each condition such as: 1) a longer or multi-session study with more utterances per condition, 2) adding noise to participant's audio to increase errors, or 3) occasionally presenting the second best recognition result.

Our experimental application first recorded a participant's entire spoken utterance before sending it to a remote server for speech recognition. Once the client received the recognition result, it was sent to another remote server to generate the TTS audio. This resulted in participants waiting around four seconds to hear their recognition result. A real-world interface could reduce this latency by: 1) streaming audio to the speech recognizer, 2) performing speech recognition and TTS on the same server, and 3) streaming TTS audio to the client as it is generated.

We used the same TTS voice for all conditions. Future work could explore the impact of the specific TTS voice or technology (e.g. concatenative versus neural TTS). Since synthetic speech can be generated in a hyperarticulate style [1], it would be interesting to investigate using hyperarticulate speech for low-confidence results. This might both help draw attention to the potential error and help users better discriminate true errors from false positives.

We changed the presentation of the entire recognition result based on the utterance confidence score. If word-level confidence scores are available, future work could test modifying the audio of individual words suspected of being incorrect. This could be done by changing a word's speaking style, rate, or by adding auditory markers near the word. Another possibility would be to repeat suspected word errors to help users verify if they are true errors.

5 Conclusion

Our study investigated whether changing the audio of a speech recognition result based on its confidence score helped participants detect more recognition errors. We changed the audio either by slowly the TTS or by adding a beep tone. We found using confidence scores showed a trend toward improved error detection including reducing the variability in the detection accuracy of our users. However, the 85% detection accuracy of selectively slowing playback was not statistically different than the 80% accuracy of the baseline condition that played all results at normal speed. This highlights the need for further research to validate and expand upon these findings. Nevertheless, we believe our results can help inform the design of more effective error detection features for eyes-free speech interaction.

Acknowledgments

This material is based upon work supported by the NSF under Grant No. IIS-1909248.

References

[1] Matthew P Aylett. 2005. Synthesising hyperarticulation in unit selection TTS. In *INTERSPEECH 2005-Eurospeech, 9th European Conference on Speech Communication and Technology, Lisbon, Portugal, September 4-8, 2005*. 2521–2524.

[2] 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*. ACM, Bellevue Washington, 1–8. doi:10.1145/2513383.2513440

[3] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: a framework for self-supervised learning of speech representations. In *Proceedings of the 34th International Conference on Neural Information Processing Systems (NIPS'20)*. Curran Associates Inc., Red Hook, NY, USA, 12449–12460.

[4] Larwan Berke, Christopher Caulfield, and Matt Huenerfauth. 2017. Deaf and Hard-of-Hearing Perspectives on Imperfect Automatic Speech Recognition for Captioning One-on-One Meetings *(ASSETS '17)*. Association for Computing Machinery, New York, NY, USA, 155–164. doi:10.1145/3132525.3132541

[5] Jeff A. Bilmes, Patricia Dowden, Howard Chizeck, Xiao Li, Jonathan Malkin, Kelley Kilanski, Richard Wright, Katrin Kirchhoff, Amarnag Subramanya, Susumu Harada, and James A. Landay. 2005. The vocal joystick: a voice-based human-computer interface for individuals with motor impairments. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing - HLT '05*. Association for Computational Linguistics, Vancouver, British Columbia, Canada, 995–1002. doi:10.3115/1220575.1220700

[6] Moira Burke, Brian Amento, and Philip Isenhour. 2006. Error Correction of Voicemail Transcripts in SCANMail. 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, 339–348. doi:10.1145/1124772.1124823

[7] Michelle Cohn and Georgia Zellou. 2020. Perception of concatenative vs. neural text-to-speech (TTS): Differences in intelligibility in noise and language attitudes. In *Proceedings of Interspeech*.

[8] Jiayue Fan, Chenning Xu, Chun Yu, and Yuanchun Shi. 2021. Just Speak It: Minimize Cognitive Load for Eyes-Free Text Editing with a Smart Voice Assistant. In *The 34th Annual ACM Symposium on User Interface Software and Technology*. ACM, Virtual Event USA, 910–921. doi:10.1145/3472749.3474795

[9] Kazuki Fujiwara. 2016. Error Correction of Speech Recognition by Custom Phonetic Alphabet Input for Ultra-Small Devices. In *Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems* (San Jose, California, USA) *(CHI EA '16)*. Association for Computing Machinery, New York, NY, USA, 104–109. doi:10.1145/2851581.2890380

[10] Debjyoti Ghosh, Can Liu, Shengdong Zhao, and Kotaro Hara. 2020. Commanding and Re-Dictation: Developing Eyes-Free Voice-Based Interaction for Editing

Dictated Text. *ACM Transactions on Computer-Human Interaction* 27, 4 (Aug. 2020), 1–31. doi:10.1145/3390889

[11] L. Gillick, Y. Ito, and J. Young. 1997. A probabilistic approach to confidence estimation and evaluation. In *1997 IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 2. 879–882 vol.2. doi:10.1109/ICASSP.1997.596076 ISSN: 1520-6149.

[12] Sharon Goldwater, Dan Jurafsky, and Christopher D. Manning. 2010. Which words are hard to recognize? Prosodic, lexical, and disfluency factors that increase speech recognition error rates. *Speech Communication* 52, 3 (March 2010), 181–200. doi:10.1016/j.specom.2009.10.001

[13] Jonggi Hong and Leah Findlater. 2018. Identifying Speech Input Errors Through Audio-Only Interaction. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–12. doi:10.1145/3173574.3174141

[14] Jonggi Hong, Christine Vaing, Hernisa Kacorri, and Leah Findlater. 2020. Reviewing Speech Input with Audio: Differences between Blind and Sighted Users. *ACM Trans. Access. Comput.* 13, 1, Article 2 (apr 2020), 28 pages. doi:10.1145/3382039

[15] Jizhou Huang, Haifeng Wang, Shiqiang Ding, and Shaolei Wang. 2022. DuIVA: An Intelligent Voice Assistant for Hands-free and Eyes-free Voice Interaction with the Baidu Maps App. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. ACM, Washington DC USA, 3040–3050. doi:10.1145/3534678.3539030

[16] Vladimir I Levenshtein et al. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. Soviet Union, 707–710.

[17] Alexander H. Liu, Wei-Ning Hsu, Michael Auli, and Alexei Baevski. 2023. Towards End-to-End Unsupervised Speech Recognition. In *2022 IEEE Spoken Language Technology Workshop (SLT)*. 221–228. doi:10.1109/SLT54892.2023.10023187

[18] Oussama Metatla, Alison Oldfield, Taimur Ahmed, Antonis Vafeas, and Sunny Miglani. 2019. Voice User Interfaces in Schools: Co-designing for Inclusion with Visually-Impaired and Sighted Pupils. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, Glasgow Scotland Uk, 1–15. doi:10.1145/3290605.3300608

[19] Sadia Nowrin, Patricia OrdóñEz, and Keith Vertanen. 2022. Exploring Motor-impaired Programmers' Use of Speech Recognition. In *The 24th International ACM SIGACCESS Conference on Computers and Accessibility*. ACM, Athens Greece, 1–4. doi:10.1145/3517428.3550392

[20] JM Noyes and CR Frankish. 1994. Errors and error correction in automatic speech recognition systems. *Ergonomics* 37, 11 (1994), 1943–1957.

[21] Agnès Piquard-Kipffer, Odile Mella, Jérémy Miranda, Denis Jouvet, and Luiza Orosanu. 2015. Qualitative investigation of the display of speech recognition results for communication with deaf people. In *Proceedings of SLPAT 2015: 6th Workshop on Speech and Language Processing for Assistive Technologies*. 36–41.

[22] Alisha Pradhan, Kanika Mehta, and Leah Findlater. 2018. "Accessibility Came by Accident": Use of Voice-Controlled Intelligent Personal Assistants by People with Disabilities. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems (CHI '18)*. Association for Computing Machinery, New York, NY, USA, 1–13. doi:10.1145/3173574.3174033

[23] Kathleen J. Price and Andrew Sears. 2005. Speech-based text entry for mobile handheld devices: An analysis of efficacy and error correction techniques for server-based solutions. *International Journal of Human-Computer Interaction* 19, 3 (2005), 279–304. doi:10.1207/s15327590ijhc1903_1 Funding Information: The authors thank Aether Systems, Inc. for their support of this research. This material is based upon work supported by the National Science Foundation under Grants IIS–9910607 and IIS–0121570. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF)..

[24] Brent N Shiver and Rosalee J Wolfe. 2015. Evaluating alternatives for better deaf accessibility to selected web-based multimedia. In *Proceedings of the 17th international ACM SIGACCESS conference on computers & accessibility*. 231–238.

[25] Than Htut Soe, Frode Guribye, and Marija Slavkovik. 2021. Evaluating AI assisted subtitling. In *Proceedings of the 2021 ACM International Conference on Interactive Media Experiences* (Virtual Event, USA) *(IMX '21)*. Association for Computing Machinery, New York, NY, USA, 96–107. doi:10.1145/3452918.3458792

[26] 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*. ACM, Glasgow Scotland Uk, 1–14. doi:10.1145/3290605.3300821

[27] Keith Vertanen and Per Ola Kristensson. 2008. On the benefits of confidence visualization in speech recognition. In *CHI '08: Proceedings of the SIGCHI conference on Human Factors in computing systems* (Florence, Italy). ACM, 1497–1500.

[28] Keith Vertanen and Per Ola Kristensson. 2009. Automatic selection of recognition errors by respeaking the intended text. In *2009 IEEE Workshop on Automatic Speech Recognition & Understanding*. 130–135. doi:10.1109/ASRU.2009.5373347

[29] Alexandra Vtyurina, Adam Fourney, Meredith Ringel Morris, Leah Findlater, and Ryen W. White. 2019. Bridging Screen Readers and Voice Assistants for Enhanced Eyes-Free Web Search. In *The World Wide Web Conference*. ACM, San Francisco CA USA, 3590–3594. doi:10.1145/3308558.3314136

[30] Amber Wagner, Ramaraju Rudraraju, Srinivasa Datla, Avishek Banerjee, Mandar Sudame, and Jeff Gray. 2012. Programming by voice: a hands-free approach for motorically challenged children. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*. ACM, Austin Texas USA, 2087–2092. doi:10.1145/2212776.2223757

[31] Yiming Wang, Jinyu Li, Heming Wang, Yao Qian, Chengyi Wang, and Yu Wu. 2022. Wav2vec-Switch: Contrastive Learning from Original-Noisy Speech Pairs for Robust Speech Recognition. In *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 7097–7101. doi:10.1109/ICASSP43922.2022.9746929 ISSN: 2379-190X.

[32] Felix Weninger, Hakan Erdogan, Shinji Watanabe, Emmanuel Vincent, Jonathan Le Roux, John R. Hershey, and Björn Schuller. 2015. Speech Enhancement with LSTM Recurrent Neural Networks and its Application to Noise-Robust ASR. In *Latent Variable Analysis and Signal Separation (Lecture Notes in Computer Science)*, Emmanuel Vincent, Arie Yeredor, Zbyněk Koldovský, and Petr Tichavský (Eds.). Springer International Publishing, Cham, 91–99. doi:10.1007/978-3-319-22482-4_11

A Questionnaire

Figure 4 shows the questions we asked participants at the start of the study. Figure 5 show the questions we asked participants at the end of the study.

INITIAL QUESTIONNAIRE

Finding and Correcting Speech Recognition Errors

Age (approximate): _____

Gender: _____

How much do you agree or disagree with the following statements (**X a single circle**)?

#	Statements	Strongly disagree						Strongly agree
1	I consider myself a ***fluent English speaker***	o 1	o 2	o 3	o 4	o 5	o 6	o 7
2	When I speak English, I have a ***non-native accent***	o 1	o 2	o 3	o 4	o 5	o 6	o 7
3	I ***frequently use speech recognition*** to control or enter text on my computer, mobile device, or smart speaker	o 1	o 2	o 3	o 4	o 5	o 6	o 7
4	When I speak English, ***people have trouble understanding me***	o 1	o 2	o 3	o 4	o 5	o 6	o 7
5	When I speak English, ***computers have trouble understanding me*** (e.g. Siri, Alexa, Google Assistant)	o 1	o 2	o 3	o 4	o 5	o 6	o 7

Figure 4: Our initial questionnaire asked participants their age and gender. They then rated five statements about their English ability and their experience with speech recognition. Statements were rated on a 7-point Likert scale.

Post-Experiment QUESTIONNAIRE

Finding and Correcting Speech Recognition Errors

How much do you agree or disagree with the following statements (**X a single circle**)?

#	Statements	Strongly disagree						Strongly agree
1	Identifying errors was easy in "Normal Speech Rate" condition.	o 1	o 2	o 3	o 4	o 5	o 6	o 7
2	Identifying errors was easy in "Slow Speech rate" condition.	o 1	o 2	o 3	o 4	o 5	o 6	o 7
3	Identifying errors was easy in "Slow Speech rate for Uncertain Recognition" condition.	o 1	o 2	o 3	o 4	o 5	o 6	o 7
4	Identifying errors was easy in "Beep for Uncertain Recognition" condition.	o 1	o 2	o 3	o 4	o 5	o 6	o 7
5	Before listening to the audio, I had a good idea if there will be any recognition errors.	o 1	o 2	o 3	o 4	o 5	o 6	o 7
6	What did you like about the experiment?							
7	What did you dislike about the experiment?							

Figure 5: Our final questionnaire asked participants to rate five statements about their ability to find errors in the experiment's four conditions. They also rated their ability to anticipate sentences that would likely have recognition errors. Statements were rated on a 7-point Likert scale. We also asked open two open ended questions about what they liked or disliked about the experiment.