BeParrot: Efficient Interface for Transcribing Unclear Speech via Respeaking

Riku Arakawa[*]
Carnegie Mellon University
Pittsburgh, USA
rarakawa@cs.cmu.edu

Hiromu Yakura[*]
University of Tsukuba / National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Japan
hiromu.yakura@aist.go.jp

Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST)
Tsukuba, Japan
m.goto@aist.go.jp

ABSTRACT

Transcribing speech from audio files to text is an important task not only for exploring the audio content in text form but also for utilizing the transcribed data as a source to train speech models, such as automated speech recognition (ASR) models. A post-correction approach has been frequently employed to reduce the time cost of transcription where users edit errors in the recognition results of ASR models. However, this approach assumes clear speech and is not designed for unclear speech (such as speech with high levels of noise or reverberation), which severely degrades the accuracy of ASR and requires many manual corrections. To construct an alternative approach to transcribe unclear speech, we introduce the idea of *respeaking*, which has primarily been used to create captions for television programs in real time. In respeaking, a proficient human respeaker repeats the heard speech as shadowing, and their utterances are recognized by an ASR model. While this approach can be effective for transcribing unclear speech, one problem is that respeaking is a highly cognitively demanding task and extensive training is often required to become a respeaker. We address this point with *BeParrot*, the first interface designed for respeaking that allows novice users to benefit from respeaking without extensive training through two key features: *parameter adjustment* and *pronunciation feedback*. Our user study involving 60 crowd workers demonstrated that they could transcribe different types of unclear speech 32.2 % faster with BeParrot than with a conventional approach without losing the accuracy of transcriptions. In addition, comments from the workers supported the design of the adjustment and feedback features, exhibiting a willingness to continue using BeParrot for transcription tasks. Our work demonstrates how we can leverage recent advances in machine learning techniques to overcome the area that is still challenging for computers themselves with the help of a human-in-the-loop approach.

CCS CONCEPTS

• **Human-centered computing → Interaction techniques**; *Human computer interaction (HCI)*.

KEYWORDS

respeak, speech transcription, automated speech recognition

ACM Reference Format:
Riku Arakawa, Hiromu Yakura, and Masataka Goto. 2022. BeParrot: Efficient Interface for Transcribing Unclear Speech via Respeaking. In *27th International Conference on Intelligent User Interfaces (IUI '22), March 22–25, 2022, Helsinki, Finland*. ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3490099.3511164

1 INTRODUCTION

The importance of speech transcription is widely acknowledged because the acquired text can be used in diverse situations, e.g., searching audio content [25], increasing the accessibility of videos [11], analyzing language production [21], and developing various speech processing models such as speech recognition [24] and voice conversion [1, 32]. However, manual transcription is time-consuming and tedious; thus, previous studies have attempted to address this issue by utilizing speech processing technology. Specifically, a post-correction approach has been applied to handle this issue, in which an automated speech recognition (ASR) model is first applied to the speech to transcribe, and then a human corrects errors in the recognition results while listening to the audio. It has been confirmed that the improvement of ASR models allows people to correct fewer errors in the recognition results, thereby reducing the total time for transcription [16, 18].

In other words, the efficiency of the post-correction approach is highly dependent on the accuracy of ASR models. In particular, this approach is known to be less effective when the performance of the ASR models is low, e.g., when the audio to transcribe is unclear due to its recording condition. Gaur et al. [4] and Sperber et al. [36] experimentally demonstrated that the word error rate (WER)[1] is desirable to be less than 30% for the post-correction approach to be effective. Therefore, previous studies have focused on applying this approach to the transcription of clear speech, e.g., audio from TED Talks [4, 36], news [45], and lectures [22]. As a result, to the best of our knowledge, an optimal approach to transcribe unclear speech (e.g., speech with a lot of noise or reverberation) remains underexplored.

In this paper, we introduce the idea of *respeaking* [13, 27, 35] to the transcription of unclear speech. It has primarily been used to create subtitles for television programs in real time [13, 19, 30]. Rather than directly inputting the audio of the programs into an

[*] These authors contributed equally and are ordered alphabetically.

IUI '22, March 22–25, 2022, Helsinki, Finland
© 2022 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9144-3/22/03.
https://doi.org/10.1145/3490099.3511164

[1] WER denotes the percentage of words that are not correctly recognized. The lower value of WER indicates that the recognition results contain fewer errors and are more accurate.

ASR model, respeaking involves a human *respeaker* who repeats the speech of the programs as if speech shadowing [20]. The respeaker's utterance is inputted into an ASR model that outputs the corresponding transcription in real time. The obtained transcription is used as subtitles after the respeaker or another person manually corrects errors as necessary. This approach assumes that the ASR model can recognize the utterance of the respeaker more accurately than the original speech in television programs, thereby reducing the number of required manual error corrections.

We presume that respeaking can be used to improve the efficiency of transcribing unclear speech. Specifically, people can transcribe speech with noise or reverberation by repeating the speech content in a quiet environment with a clear voice and inputting their utterance into an ASR model. In this manner, the utterance can be recognized accurately using a contemporary ASR model, and then they can post-correct fewer errors in the recognition results. Thus, we can expect that the time required to transcribe will be reduced compared to the case where they directly input the unclear speech into the ASR model and post-correct errors.

However, it is not trivial to determine whether this approach is feasible because the effectiveness of respeaking is highly dependent on the proficiency of the respeaker [30, 31]. In particular, respeakers incur high cognitive demand; i.e., they are required to simultaneously listen to what is being said in order to repeat it without delay and memorize the speech content for post-correction of errors in the recognition result. In addition, respeakers should be able to utter clearly without stuttering or stammering such that utterances are transcribed accurately by the ASR models. Thus, respeaking is considered a profession, and dedicated training programs have been developed at universities and television stations [29, 31]. In other words, it would be challenging for novice users to employ respeaking for transcription.

Therefore, we propose *BeParrot*, an efficient interface for transcription via respeaking. This interface is designed to allow users to transcribe not only clear speech but also unclear speech even when they are not proficient in respeaking. It is enabled by utilizing the history of how a user has interacted with the interface (e.g., retrying utterance and editing recognition results), which would reflect the user's ability of respeaking. On the basis of the interaction history, we implemented two key features in BeParrot, *parameter adjustment* and *pronunciation feedback*. The parameter adjustment feature automatically updates two parameters, the playback speed and length of each speech segment, because these parameters determine the difficulty of respeaking according to previous studies [29–31]. For example, when the user has retried an utterance of the same segment, the user is likely to have trouble in respeaking the segment, and the playback speed can then be automatically decreased. The user of BeParrot can also manually adjust these parameters when they become accustomed to respeaking and want to increase the playback speed or segment length. Furthermore, the pronunciation feedback feature presents phonemes that are difficult to be recognized by the ASR model when pronounced by the user, which is derived from the history of their manual error corrections. Then, the user can leverage the feedback for improving their pronunciations, especially of those the ASR model would not recognize correctly.

We evaluated the effectiveness of BeParrot on different types of speech data by involving 60 crowd workers. The results successfully demonstrated that the workers could transcribe the speech more efficiently with BeParrot than a conventional post-correction approach. Particularly, the effectiveness of BeParrot was confirmed when the speech is unclear and difficult to recognize using ASR models, e.g., speech with high levels of noise or reverberation [8, 37, 38]. In addition, comments from the workers qualitatively supported the design of BeParrot, expressing their affirmative reception of both the parameter adjustment and pronunciation feedback features, as well as suggesting room for further improvements.

2 RELATED WORK

Our goal in this paper is to support transcription tasks via respeaking. To situate our work, in this section, we first cover previously proposed interfaces for supporting speech transcription, most of which adopt the post-correction of the recognition results of ASR models. We then describe the concept of respeaking and how it has been used as well as the difficulties novice users encounter when performing respeaking.

2.1 Interface for Supporting Transcription

To moderate the importance of speech transcription and its time cost, there are several studies that propose supporting interfaces for transcription tasks, as mentioned in Section 1. For example, as one of the initial works, Barras et al. [2] proposed an integrated text editor that visualizes a speech wave. Given the development of ASR techniques, the post-correction approach has become widely used to assist speech transcription [16–18, 22, 36, 45]. For example, Liu and Soong [16] developed a handwriting user interface that allows users to correct errors in a convenient manner. Luz et al. developed a 3D game interface to support a collaborative correction process by motivating users [17].

These approaches assume that the speech to transcribe can be recognized by ASR models with certain accuracy. For example, the WER of the pre-correction transcription was reported to be 21.5% in the study conducted by Luz et al. [18] and 28.8% in the study conducted by Miro et al. [22]. This setting is in agreement with the findings of Gaur et al. [4] and Sperber et al. [36], who concluded that the WER is desirable to be less than 30% for the post-correction approach to be effective.

However, depending on the nature of the audio source, this is not always feasible even with contemporary ASR models. Specifically, the performance of the ASR models is degraded when the speech is unclear, e.g., when the speech contains high levels of noise or reverberation. For example, Tsunoo et al. [39] reported a WER of 48.6% with their ASR model on a dataset of unclear speech [47] even though they leveraged various data augmentation techniques involving reverberation simulation and adversarial training. Yet, previous studies for supporting speech transcription often evaluated their approach using clear speech, e.g., audio from TED Talks [4, 36], as mentioned in Section 1. We acknowledge that the within-dataset tuning of ASR models can be performed to achieve high accuracy [16, 36], but it limits their applicability to a wide variety of speech data. These limitations motivated us to develop an efficient interface that can work without dependence on the audio source.

2.2 Respeaking

As mentioned in Section 1, respeaking has been practically used by many broadcasting stations, e.g., BBC [19] and NHK[2] [10, 13], to create subtitles for news programs or sportscasts in real time. To minimize the delay when creating subtitles, respeakers must repeat the speech clearly without delay or stammering such that the repeated speech can be transcribed accurately by an ASR model. Therefore, broadcasting stations assume professional respeakers who have completed specialized training [30, 31]. For example, Prazak et al. [27] described how Czech Television leverages respeaking, stating that respeakers are required to undergo 75 hours of in-house training. In addition, Waes et al. [43] found that respeakers are required to employ a variety of strategies to limit information loss as much as possible.

Given that respeaking demands high proficiency, previous studies did not explore the possibilities of novice users transcribing via respeaking extensively. In fact, the difficulties associated with mastering respeaking were discussed by Ghyselen et al. [5] as a reason why they did not adopt this approach for transcribing a dialect corpus in their study. In addition, Sperber et al. [35] empirically demonstrated that the effectiveness of respeaking is strongly dependent on how accurately the respeaker's utterances are recognized in a study where two speakers tried transcribing TED Talks via respeaking. Vashistha et al. [40–42] employed a workaround in their studies by adopting segmenting and majority-voting. Here, they deployed crowdsourcing transcription tasks in which speech data were divided into small segments (less than 6 s), and the speech content of each segment was uttered by five crowd workers via respeaking. The final transcription was obtained via a majority vote among the recognition results of the repeated speech.

We expect that respeaking can be a powerful tool for speech transcription if it is made easier for novice users. In addition, we believe that it is especially efficient for transcribing unclear speech, which is typically challenging using the conventional post-correction approach (Section 2.1). This is because respeakers will be able to utter clear speech that is recognizable by ASR models without depending on the quality of the original audio. Therefore, in this study, we attempted to develop an effective interface to make respeaking more accessible by novice users. By doing so, we aimed to demonstrate its effectiveness in transcribing different types of unclear speech, which, to the best of our knowledge, has not been explored extensively in previous studies.

3 PROPOSED INTERFACE

In this section, we introduce the proposed *BeParrot* interface. We first discuss two key features that help novice users perform respeaking based on the findings of previous studies that investigate the training of respeaking. We then describe the implementation of BeParrot in detail.

3.1 Design

In speech transcription tasks, an audio clip to transcribe is typically divided into short segments in advance, and each segment is transcribed sequentially by users. In our case, we assume that each segment is transcribed via respeaking along with manual correction. However, as mentioned in Section 2.2, respeaking is a highly challenging task that is difficult for novice users. Thus, prior to developing BeParrot, we referred to studies on training programs for professional respeakers [29–31]. Based on the findings of these studies, we designed two features that we expected would help novice users utilize respeaking without requiring extensive training.

The first feature is parameter adjustment that allows users to control both the playback speed of the speech and the length of each speech segment dynamically during the task. Relative to playback speed, Fresco [29] reported that, in some respeaker training programs, trainees attempt to identify an optimal playback speed through multiple steps. They stated that the optimal speed varies significantly for different people, and the speed affects respeaking performance. These observations informed us to design the adjusting feature of the playback speed. In addition, the length of each speech segment was made adjustable to allow users to control their cognitive load during the task. In fact, Fresco [29] reported that trainees identified "multitasking" as the most difficult aspect of respeaking. Specifically, respeakers must clearly repeat what is being said while listening to and remembering the speech, as we discussed in Section 1. Thus, we anticipated that allowing users to adjust the segment length based on their ability would be helpful.

In addition to making these two parameters adjustable by users, we enabled BeParrot to automatically adjust the parameters based on users' interaction history. Specifically, if a user is experiencing difficulty in respeaking (e.g., retrying a specific speech segment over and over), BeParrot automatically reduces the playback speed and segment length. This is because the effectiveness of such automated adjustments has been confirmed previously in the development of a tool to support language learning via speech shadowing [48].

The second feature is pronunciation feedback, where users are prompted to be careful about pronouncing specific phonemes. Given the nature of respeaking, utterances should be recognized accurately by the ASR model, and respeakers must utter clearly without stuttering or stammering [30]. Thus, we attempted to increase awareness of certain phonemes in utterances that are difficult to hear or are likely to be misrecognized by the ASR model. This feedback feature can be achieved by analyzing how users manually correct the recognition results, which we describe in detail in the next section.

3.2 Implementation

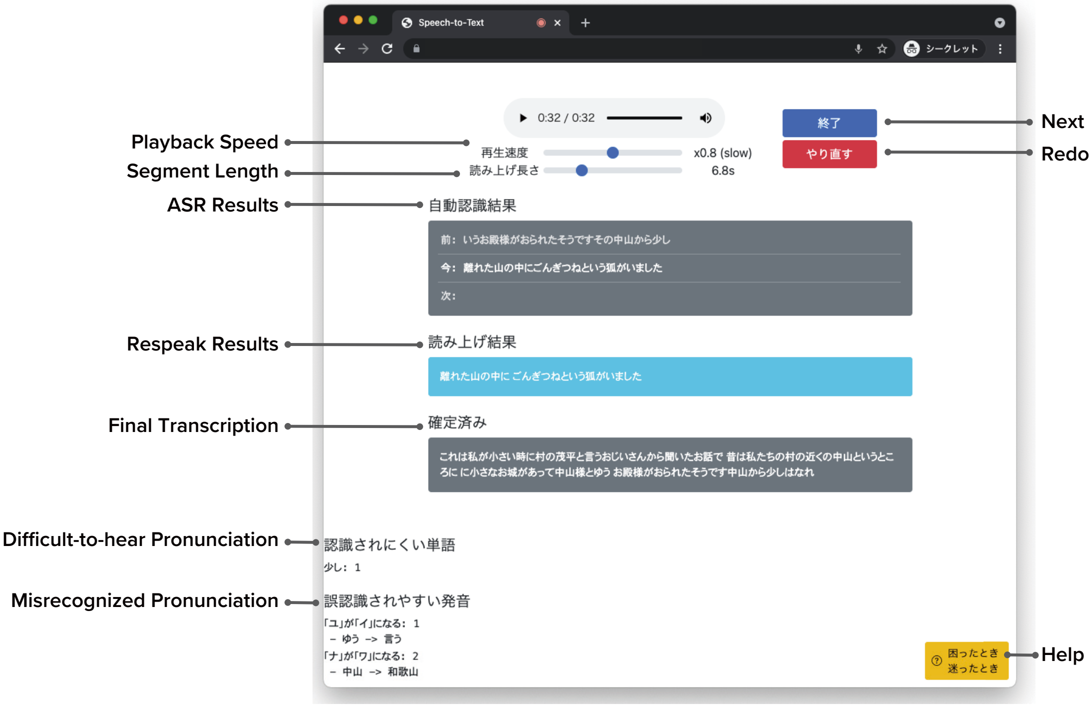
To realize these two key features, BeParrot was implemented in the form of a web-based interface (Figure 1). We implemented the interface in Japanese because we conducted a user study on a Japanese crowdsourcing platform[3], as we describe later in Section 4.

As mentioned in Section 3.1, the entire audio clip to transcribe is first divided into short pieces using a voice activity detection technique [34], specifically `webrtcvad`[4]. The split speech pieces, which are approximately from 0.1 to 3 sec, are then concatenated into a segment until its length exceeds the parameter specifying the length of each speech segment using a greedy algorithm. Once

[2] NHK is a Japanese government-owned public broadcasting station.

[3] https://www.lancers.jp

[4] https://github.com/wiseman/py-webrtcvad



再生速度 x0.8 (slow)

読み上げ長さ 6.8s

自動認識結果

前：いうお殿様がおられたそうですその中山から少し

今：離れた山の中にごんぎつねという狐がいました

次：

読み上げ結果

離れた山の中に ごんぎつねという狐がいました

確定済み

これは私が小さい時に村の茂平と言うおじいさんから聞いたお話で 昔は私たちの村の近くの中山というところに に小さなお城があって中山様とゆう お殿様がおられたそうです中山から少しはなれ

認識されにくい単語

少し：1

誤認識されやすい発音

「ユ」が「イ」になる：1
 – ゆう –> 言う
「ナ」が「ワ」になる：2
 – 中山 –> 和歌山

Figure 1: BeParrot interface. The interface was implemented in Japanese, and each text outside the window indicates corresponding meanings in English.

the parameter is adjusted either manually or automatically, the concatenation is recalculated.

In BeParrot, a user is supposed to transcribe each of the concatenated segments in order via respeaking. Here, the user's utterance of a segment is recognized sequentially using a streaming ASR model, and the recognition result appears in "Respeak Results" in real time. Note that the user can edit errors in the text manually using a keyboard if necessary. Once the transcription for the segment is finalized by clicking "Next," it is concatenated to the entire transcription shown in "Final Transcription." Otherwise, if the user wants to utter the same segment again, they can do so by clicking "Redo."

In a similar manner to the post-correction approach mentioned in Section 2.1, the audio source to transcribe is recognized using an ASR model in advance, and the result is presented in "ASR Results" for reference. Here, the user can observe three transcriptions in a row that correspond to the speech of the previous, current, and post segment, respectively. In addition, there is a "Help" button in the bottom-right area that the user can press at any time to view information about the usage of BeParrot.

To achieve the parameter adjustment feature as described in Section 3.1, we implemented two sliders, i.e., the "Playback Speed" and "Segment Length" sliders, in BeParrot to allow users to adjust the playback speed and segment length, respectively. Note that these parameters are also adjusted automatically based on the interaction records of the individual user. Specifically, if the user retries uttering the same segment multiple times, BeParrot identifies the user as having trouble uttering it without stuttering or stammering and slows the playback speed by 0.85. If the user stops playback in the middle of the segment, the user may encounter "multitasking" difficulties with the long segment. Thus, BeParrot reduces the length of consecutive segments automatically to the length they stopped at with the decay parameter of 0.5.

The bottom-left area of BeParrot is used for the pronunciation feedback feature introduced in Section 3.1. In "Difficult-to-hear Pronunciation," words that are not recognized by the streaming ASR model and added manually by the user using the keyboard are listed in the order of the number of additions. In addition, "Misrecognized Pronunciation" presents a list of moras that are often misrecognized by the ASR model based on how the user corrected the recognition results of their utterances. This list is obtained by calculating the optimal edit operations between the mora sequences of the original and corrected word using the Wagner–Fischer algorithm [44]. The calculated edit operations over speech segments allow BeParrot to identify how many times each mora was corrected and to rank those that are frequently misrecognized.

Here, two ASR models are employed in BeParrot; one processes the audio clip in advance, and the other processes the user's utterances sequentially in real time. For the former model, we used Conformer [7], which is one of the state-of-the-art ASR models based on both Transformer and CNN architectures. The Conformer model was trained on a Japanese corpus using ESPNet [46], which is an open-source end-to-end speech processing toolkit. For the latter ASR model, we used Google Speech-to-Text API[5], which allows us to transcribe streaming audio in real time. The latency of respeaking (i.e., from the time a user utters to the time the corresponding recognition result appears on "Respeak Results") is approximately 800 ms. Note that, when using BeParrot, the user is required to wear headphones to avoid the original speech from being mixed with the user's utterances.

4 USER STUDY

To confirm the effectiveness of the proposed BeParrot interface, we conducted a user study involving crowd workers. We asked the workers to transcribe different types of speech data using BeParrot or the conventional post-correction approach. Then, we compared the time they spent in transcribing the same speech and the accuracy of the obtained transcriptions between the two approaches (i.e., respeaking and post-correction).

4.1 Materials

For this user study, we prepared three types of speech data, i.e., *clear*, *radio*, and *historical* speech. Each type comprised two audio clips of three minutes; thus, six clips in total were considered. In addition, we recruited crowd workers in Japan (Section 4.2); thus, we prepared the clips in Japanese.

The clear clips were taken from TED Talks recorded in a quiet environment. The radio clips were prepared to evaluate the effectiveness of BeParrot on noisy speech because they were taken from a radio broadcast in which two professional speakers debated while background music was played. The historical clips were more challenging because they were taken from two lectures from the 1970s. These clips contain high levels of reverberation as well as noise because the lectures were recorded using a single microphone located in a large auditorium.

For each clip, we prepared its transcription text as the ground truth from the corresponding source (e.g., the TED website for the clear clips) to evaluate the accuracy of transcriptions obtained in this study. Using the ground truth, we confirmed in advance that the clips were sufficiently unclear to transcribe using the ASR model such that they resulted in a character error rate (CER) of more than 30%[6] in Table 1.

4.2 Design

We employed a between-participant design across the *baseline* and *proposed* conditions. Here, we did not include a condition where a participant performs transcription tasks via respeaking without the support of BeParrot, so to speak, *vanilla* condition. This is because, in our initial exploration, we found that novice users could not conduct respeaking in such a *vanilla* condition, which is attributed to the difficulties associated with conducting respeaking (see Section 2.2).

For the baseline condition, we prepared a web-based interface that allowed the crowd workers to transcribe speech using the post-correction approach. Similar to the interface used by Gaur et al. [4], the web-based interface comprised an audio player and presentation of the recognition results obtained by the ASR model. We also replicated keyboard shortcuts that Gaur et al. [4] implemented to allow the workers to play/pause or rewind 5 s. After being presented the instructions on how to use this interface, the workers assigned to the baseline condition were asked to transcribe one of the six prepared clips (Section 4.1).

For the crowd workers assigned to the proposed condition, we provided BeParrot to transcribe the speech. Since we assumed workers who are unaccustomed to respeaking, we asked them to first practice the use of BeParrot by transcribing a clear speech of 30 s[7] after watching an instructional video. This process took approximately 3.5 minutes on average. Then, we asked the workers to transcribe one of the prepared clips in the same manner as those in the baseline condition. Here, the implementation of BeParrot was customized to record workers' interactions, e.g., retrying an utterance, changing playback speed, and correcting errors using a keyboard.

In addition, the workers were asked to complete a questionnaire after they finished the transcription task. The questionnaire included the items from NASA-TLX [3, 9] to compare the workers' cognitive loads between the two conditions. In addition, for the workers assigned to the proposed condition, there were questions to collect their opinions regarding the usage of BeParrot, e.g., "how useful was the parameter adjustment feature?," "how useful was the pronunciation feedback feature?," and "please write down anything else you noticed about the experience of using this interface." Their responses to these questions were later analyzed using open coding [33] to enumerate major topics.

For each of the two conditions, we recruited 30 crowd workers and randomly assigned them one of the six clips, resulting in five workers for each clip. The recruitment process was performed on a Japanese crowdsourcing platform, and they were paid approximately $10 for their participation. For the proposed condition, we required the workers to use headphones or earphones (not speakers) so they could perform respeaking, as we mentioned in Section 3.2. In addition, to exclude data of workers who have an experience of respeaking in the past, two of the authors independently examined the questionnaire responses from the workers of the proposed condition; however, no cases confirmed.

4.3 Measure

To evaluate the effectiveness of BeParrot, we prepared two measures, i.e., time and character error rate (CER). Here, we measured the time each worker spent transcribing clips of the same length and compared this across the two conditions to verify whether BeParrot contributed to the efficiency of the transcription task. We also compared the CER of the obtained transcriptions to confirm the effect of BeParrot on transcription accuracy. Note that CER is widely used to evaluate the quality of transcription in languages without space delimiters [14], including Japanese [12], because the WER value calculated for such languages depends on the quality of morphological analysis. Still, the value of CER generally correlates with the value of WER while it is expected to be lower than WER, as in Petridis et al. [26].

[5] https://cloud.google.com/speech-to-text/docs/streaming-recognize

[6] As mentioned in Section 4.3, the value of WER generally correlates with the value of CER, and thus, the higher value of CER indicates that the recognition results contain more errors. In addition, the value of WER is usually higher than the value of CER.

[7] We note that the clip of clear speech used for the practice was a recitation of a famous children's story and independent of the six clips we prepared in Section 4.1.

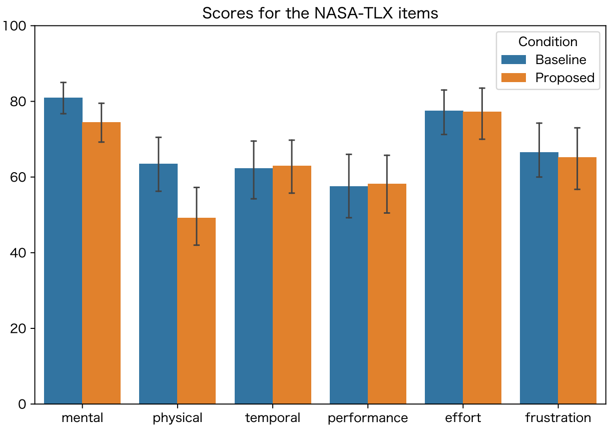


Figure 2: Comparison of worker cognitive load based on NASA-TLX (means and 95% confidence intervals).

4.4 Results

The results are shown in Table 1. According to the t-test, we found that the workers assigned to the proposed condition spent significantly less time transcribing the radio ($p = 0.037, d = 1.01$) and historical ($p = 0.026, d = 1.08$) clips. On the other hand, we could not find significant differences in CER for all clips. These points imply that BeParrot contributed to the reduction of the time required to transcribe unclear speech (32.1 % in total for the radio and historical clips[8]) compared to the conventional post-correction approach without reducing transcription accuracy significantly. The fact that the time required to transcribe clear speech was not significantly different reconfirms the findings of previous studies suggesting the effectiveness of the post-correction approach for clear speech (Section 2.1).

In addition, it is notable that the baseline condition showed a significant increase in the time required to transcribe unclear speech ($p = 0.004$) according to the one-way ANOVA. Here, the recognition results for the radio and historical clips obtained by the ASR model that were presented to the workers exhibited a CER value that was greater than 30 %. Thus, the time increase can be said to be analogous to previous studies [4, 36], which demonstrated that recognition results in the post-correction approach should be sufficiently accurate (e.g., less than 30 % WER). On the other hand, the proposed condition did not exhibit a significant difference in the time spent across the three types of speech. This highlights the effectiveness of BeParrot, as it did not exhibit a time increase against such unclear speech.

We also compared the workers' responses for the NASA-TLX items between the two conditions in the same manner as Zhang et al. [48]. The results are shown in Figure 2, where the higher score indicates a higher stress level for each of the six items. In total, we could not find a significant difference in the obtained scores, which implies that BeParrot allowed the workers to complete the transcription task without requiring extra stress compared to the conventional post-correction approach. However, the scores indicate that the amount of physical effort required for the transcription task was reduced with the proposed condition. This suggests that respeaking is less physically demanding compared to correcting erroneous recognition results using a keyboard.

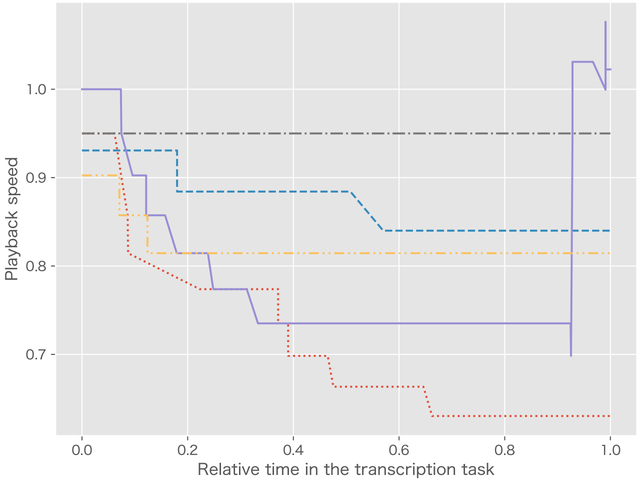


Figure 3: Transition of playback speed in the relative time of the transcription task (each line represents the transition of different workers assigned to the same clip).

Furthermore, the logged interaction records of the workers suggested the effectiveness of the parameter adjustment feature. For example, Figure 3 shows the transition of the playback speed that the workers assigned to the same clip experienced as a result of the manual and automated adjustment. As can be seen, the playback speed was gradually reduced automatically until it reached a specific value that seemed to be an optimal value for each worker. In other words, the playback speed BeParrot stopped slowing down (i.e., the speed that allowed each worker to perform respeaking without retrying) differed for each worker. This confirms our design rationale (Section 3.1), which was based on the report by Fresco [29]. In addition, from the point that the worker denoted by the purple solid line increased the playback speed at the last part of the task, we can infer that the worker got familiar with respeaking and did so. While we further discuss in Section 4.5, we consider that these results suggest the effectiveness of the design of BeParrot.

4.5 User Comments

As explained in Section 4.2, we asked the crowd workers assigned to the proposed condition to complete a questionnaire and analyzed their responses. Overall, they considered BeParrot to be useful and expressed their willingness to continue using it.

> This was my first time using this kind of transcription tool, but once I got used to it, I was able to transcribe a little faster, which I found very useful.
>
> I felt that using the voice input function would make the transcription process smoother and less stressful.
>
> It was very easy because what I had to do was limited to minor adjustments as long as I focused on listening.

[8] $100 - \frac{1823.2\,\text{(radio, proposed)} + 1408.4\,\text{(historical, proposed)}}{2758.1\,\text{(radio, baseline)} + 2004.6\,\text{(historical, baseline)}} = 32.14_{(\%)}$

Table 1: Time crowd workers spent transcribing and CER of the obtained transcription.

Speech type	Time			CER (%)		
	Baseline (s)	Proposed (s)	Reduction (%)	ASR	Baseline	Proposed
Clear	1632.7 (±224.6)	1505.8 (±426.8)	7.8	6.16	3.73 (±0.57)	5.75 (±1.02)
Radio	2758.1 (±273.6)	1823.2 (±253.4)	**33.9**	30.72	19.05 (±1.56)	24.81 (±2.69)
Historical	2004.6 (±205.0)	1408.4 (±140.4)	**29.7**	48.18	29.25 (±5.16)	29.10 (±2.17)

In addition, our design of BeParrot, which we described in Section 3.1, received positive comments. In particular, the workers frequently commented on the parameter adjustment feature. For the playback speed, some workers indicated the usefulness of the control slider, and others mentioned that the automated speed adjustment was effective.

> Every time I retried the transcription, it automatically slowed down the playback speed, which saved me from having to do it myself and made the task easier.
>
> The slowing down of the playback speed made it easier to listen and was very useful for correcting mistakes.
>
> It played faster the first time and slower since the second time, which was helpful in allowing me to transcribe more efficiently and accurately.

We found responses indicating that, as well as the adjustment of the playback speed, the adjustment of the segment length eased the complexity of respeaking.

> When the segment was long, I had my hands full memorizing the content, and it was difficult to read it aloud simultaneously. Thus, I actually utilized the adjustment feature. When dealing with complicated contents or unfamiliar fields, I think that the length adjustment function is indispensable to make sure that I can understand each phrase. Then, the relistening to long segments can be avoided, and the time required in total can be shortened.
>
> I felt it very difficult to transcribe a long segment. The system then automatically adjusted the length of the segment to be shorter, and after that, the transcription process became much easier.

These responses confirm that the parameter adjustment feature helped novice users overcome the difficulty of respeaking, as we intended (Section 3.1). We note that this feature is newly introduced for respeaking with BeParrot. Previous applications of respeaking to the subtitle creation for television programs (Section 2.2) did not allow such a feature since changing the playback speed or retrying the same segment makes it impossible to create subtitles in real time.

For the feedback feature, some workers said that it helped them become more aware of their pronunciation habits.

> The presentation of the number of incorrect pronunciations was very helpful. I realized that I have a bad tongue and I needed to be more careful in my daily life.
>
> It was useful because it made me aware of my pronunciation, especially about the "la" and "na" columns *[moras starting with the consonants of "l" and "n"]* that were displayed as often mispronounced.

However, some responses suggested that monitoring the feedback while respeaking was difficult.

> I was engrossed in uttering what I heard and did not look at the feedback at all.
>
> My mind was so occupied with the transcription task that I did not think to refer to it much.

One worker mentioned that they had trouble improving their pronunciation from the display.

> The presentation of pronunciations that tend to be misrecognized let me carefully utter them. Still, they were not recognized accurately.

These responses suggest that there is room for improvement regarding the presentation of the feedback. For example, showing the feedback after a user completes a transcription task is a possible option to ease the difficulty of referring to the feedback during respeaking. This would give them time to reflect on the feedback and consider how to improve their pronunciation in future tasks. Computationally providing auditory feedback might also be helpful for them to improve their pronunciation [23].

5 LIMITATION AND FUTURE WORK

The results presented in Sections 4.4 and 4.5 demonstrate the effectiveness of BeParrot as a tool for transcribing unclear speech. The results also indicate that its design (i.e., the parameter adjustment and pronunciation feedback) helped novice users perform respeaking. In this section, we discuss current limitations of our work and consider potential future research directions.

First, as the key component of BeParrot, we assume that the accuracy of the ASR model that recognizes respeakers' utterances should be high. This limits the applicability of BeParrot; it can be difficult to use BeParrot for speech for which accurate ASR models have not been developed (e.g., speech in languages with low resources). Consequently, evaluations in various languages and cultural backgrounds would be demanded to increase the generality of our results [15].

Second, despite our efforts to use different types of audio clips (Section 4.1), we need to evaluate the effectiveness of BeParrot when it is used to transcribe more diverse types of speech. For example, given the nature of respeaking, the difficulty of using BeParrot may increase when the speech to transcribe contains spontaneous conversation by multiple people with many overlaps. The effectiveness of BeParrot in transcribing longer speech (e.g., over an hour) is also worth to be examined. Such a study will further clarify its longitudinal learning effect, which includes both that a user becomes increasingly familiar with the interface and that the user improves their pronunciation based on the feedback feature. Longer transcription tasks would also reveal the effect on users' cognitive load compared with conventional approaches.

As one of the future works, we want to further investigate how BeParrot can be used to transcribe historical speech recordings, as it has many use cases, e.g., an analysis of dialect variations [28] and an exploration of historical information [6]. We believe BeParrot can be especially effective in these cases because the noise or reverberation conditions of such historical recordings are often unknown. Then, it would be challenging to improve the transcription accuracy using signal processing techniques for noise and reverberation reduction and data augmentation techniques in the training of ASR models.

Another future work includes the online adaptation of the streaming ASR model for each user based on their interaction records. Although we utilized the records to facilitate their respeaking via the automatic parameter adjustment and pronunciation feedback, updating the ASR model on the spot such that it is tuned for a specific user could further improve the effectiveness of BeParrot. This is feasible via further engineering efforts because we can obtain the corrected transcription and the corresponding utterance during user interactions. We will explore how such online adaptation of the ASR model can increase the accuracy of the transcription of users' utterances and help their transcription tasks overall.

6 CONCLUSION

In this paper, we proposed BeParrot, an efficient interface for transcribing unclear speech via respeaking. BeParrot features parameter adjustment and pronunciation feedback to enable novice users to conduct respeaking without extensive training. Through a user study involving 60 crowd workers, we examined how BeParrot helps users transcribe different types of unclear speech, such as speech with high levels of noise or reverberation. The results demonstrate that, compared with a conventional approach, BeParrot makes transcription tasks of unclear speech 32.2 % faster without losing the accuracy of transcriptions. From this study, we can infer that transcribing unclear speech should be still hard for either a human or a computer alone. At the same time, BeParrot would be an illustrative example in which humans and computers achieve promising results in such a task by working together via a dedicated interface.

ACKNOWLEDGMENTS

This work was supported in part by JST ACT-X JPMJAX200R, JSPS KAKENHI 21J20353, and MIC SCOPE.

REFERENCES

[1] Riku Arakawa, Shinnosuke Takamichi, and Hiroshi Saruwatari. 2019. Implementation of DNN-Based Real-Time Voice Conversion and Its Improvements by Audio Data Augmentation and Mask-Shaped Device. In *Proceedings of the 10th ISCA Speech Synthesis Workshop*. ISCA, Grenoble, France, 93–98. https://doi.org/10.21437/ssw.2019-17

[2] Claude Barras, Edouard Geoffrois, Zhibiao Wu, and Mark Liberman. 2001. Transcriber: Development and Use of a Tool for Assisting Speech Corpora Production. *Speech Communication* 33, 1–2 (2001), 5–22. https://doi.org/10.1016/S0167-6393(00)00067-4

[3] James C. Byers, Alvah C. Bittner, and Susan G. Hill. 1989. Traditional and Raw Task Load Index (TLX) Correlations: Are Paired Comparisons Necessary?. In *Proceedings of the 1989 Annual International Industrial Ergonomics and Safety Conference*. Taylor & Francis, Philadelphia, PA, 481–485.

[4] Yashesh Gaur, Walter S. Lasecki, Florian Metze, and Jeffrey P. Bigham. 2016. The Effects of Automatic Speech Recognition Quality on Human Transcription Latency. In *Proceedings of the 13th Web for All Conference*. ACM, New York, NY, 23:1–23:8. https://doi.org/10.1145/2899475.2899478

[5] Anne-Sophie Ghyselen, Anne Breitbarth, Melissa Farasyn, Jacques Van Keymeulen, and Arjan van Hessen. 2020. Clearing the Transcription Hurdle in Dialect Corpus Building: The Corpus of Southern Dutch Dialects as Case Study. *Frontiers in Artificial Intelligence* 3 (2020), 10. https://doi.org/10.3389/frai.2020.00010

[6] Michael Gref, Joachim Köhler, and Almut Leh. 2018. Improved Transcription and Indexing of Oral History Interviews for Digital Humanities Research. In *Proceedings of the 11th International Conference on Language Resources and Evaluation*. ELRA, Paris, France, 3124–3131.

[7] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-Augmented Transformer for Speech Recognition. In *Proceedings of the 21st Annual Conference of the International Speech Communication Association*. ISCA, Grenoble, France, 5036–5040. https://doi.org/10.21437/Interspeech.2020-3015

[8] Reinhold Haeb-Umbach, Jahn Heymann, Lukas Drude, Shinji Watanabe, Marc Delcroix, and Tomohiro Nakatani. 2021. Far-Field Automatic Speech Recognition. *Proc. IEEE* 109, 2 (2021), 124–148. https://doi.org/10.1109/JPROC.2020.3018668

[9] Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of Empirical and Theoretical Research. *Advances in Psychology* 52 (1988), 139–183. https://doi.org/10.1016/s0166-4115(08)62386-9

[10] Shinichi Homma, Akio Kobayashi, Takahiro Oku, Shoei Sato, Toru Imai, and Tohru Takagi. 2008. New Real-Time Closed-Captioning System for Japanese Broadcast News Programs. In *Proceedings of the 11th International Conference on Computers Helping People with Special Needs*. Springer Berlin Heidelberg, Berlin, Germany, 651–654. https://doi.org/10.1007/978-3-540-70540-6_93

[11] Richang Hong, Meng Wang, Mengdi Xu, Shuicheng Yan, and Tat-Seng Chua. 2010. Dynamic Captioning: Video Accessibility Enhancement for Hearing Impairment. In *Proceedings of the 18th ACM International Conference on Multimedia*. ACM, New York, NY, 421–430. https://doi.org/10.1145/1873951.1874013

[12] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan. 2017. Advances in Joint CTC-Attention Based End-to-End Speech Recognition with a Deep CNN Encoder and RNN-LM. In *Proceedings of the 18th Annual Conference of the International Speech Communication Association*. ISCA, Grenoble, France, 949–953.

[13] Toru Imai, Atsushi Matsui, Shinichi Homma, Takeshi Kobayakawa, Kazuo Onoe, Shoei Sato, and Akio Ando. 2002. Speech Recognition with a Re-Speak Method for Subtitling Live Broadcasts. In *Proceedings of the 7th International Conference on Spoken Language Processing*. ISCA, Grenoble, France, 5 pages.

[14] Oh-Wook Kwon and Jun Park. 2003. Korean Large Vocabulary Continuous Speech Recognition with Morpheme-based Recognition Units. *Speech Communication* 39, 3–4 (2003), 287–300. https://doi.org/10.1016/S0167-6393(02)00031-6

[15] Sebastian Linxen, Christian Sturm, Florian Brühlmann, Vincent Cassau, Klaus Opwis, and Katharina Reinecke. 2021. How WEIRD is CHI?. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, Article 143, 14 pages. https://doi.org/10.1145/3411764.3445488

[16] Peng Liu and Frank K. Soong. 2006. Word Graph Based Speech Recognition Error Correction by Handwriting Input. In *Proceedings of the 8th ACM International Conference on Multimodal Interfaces*. ACM, New York, NY, 339–346. https://doi.org/10.1145/1180995.1181059

[17] Saturnino Luz, Masood Masoodian, and Bill Rogers. 2010. Supporting Collaborative Transcription of Recorded Speech with a 3D Game Interface. In *Proceedings of the 14th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems*, Vol. 6279. Springer Berlin Heidelberg, Berlin, Germany, 394–401. https://doi.org/10.1007/978-3-642-15384-6_42

[18] Saturnino Luz, Masood Masoodian, Bill Rogers, and Chris Deering. 2008. Interface Design Strategies for Computer-Assisted Speech Transcription. In *Proceedings of the 20th Australasian Computer-Human Interaction Conference*, Vol. 287. ACM, New York, NY, 203–210. https://doi.org/10.1145/1517744.1517812

[19] Alison Marsh. 2006. Respeaking for the BBC. *inTRAlinea* 1700 (2006), 1 pages.

[20] William D. Marslen-Wilson. 1985. Speech shadowing and speech comprehension. *Speech Communication* 4, 1-3 (1985), 55–73. https://doi.org/10.1016/0167-6393(85)90036-6

[21] Lise Menn and Nan Bernstein Ratner (Eds.). 1999. *Methods for Studying Language Production*. Psychology Press, Hove, UK. https://doi.org/10.4324/9781410601599

[22] Juan Daniel Valor Miró, Joan Albert Silvestre-Cerdà, Jorge Civera, Carlos Turró, and Alfons Juan. 2015. Efficiency and Usability Study of Innovative Computer-Aided Transcription Strategies for Video Lecture Repositories. *Speech Communication* 74 (2015), 65–75. https://doi.org/10.1016/j.specom.2015.09.006

[23] Jack Mostow and Gregory Aist. 2001. *Evaluating Tutors That Listen: An Overview of Project LISTEN*. MIT Press, Cambridge, MA, 169–234.

[24] Ali Bou Nassif, Ismail Shahin, Imtinan Basem Attili, Mohammad Azzeh, and Khaled Shaalan. 2019. Speech Recognition Using Deep Neural Networks: A Systematic Review. *IEEE Access* 7 (2019), 19143–19165. https://doi.org/10.1109/ACCESS.2019.2896880

[25] Jun Ogata, Masataka Goto, and Kouichirou Eto. 2007. Automatic Transcription for a Web 2.0 Service to Search Podcasts. In *Proceedings of the 8th Annual Conference of the International Speech Communication Association*. ISCA, Grenoble, France, 2617–2620.

[26] Stavros Petridis, Themos Stafylakis, Pingchuan Ma, Georgios Tzimiropoulos, and Maja Pantic. 2018. Audio-Visual Speech Recognition with a Hybrid CTC/Attention Architecture. In *Proceedings of the 2018 IEEE Spoken Language Technology Workshop*. IEEE, New York, NY, 513–520. https://doi.org/10.1109/SLT.2018.8639643

[27] Aleš Pražák, Zdeněk Loose, Josef V. Psutka, Vlasta Radová, and Josef Psutka. 2020. Live TV Subtitling through Respeaking with Remote Cutting-Edge Technology. *Multimedia Tools and Applications* 79, 1–2 (2020), 1203–1220. https://doi.org/10.1007/s11042-019-08235-3

[28] Margaret E. L. Renwick and Rachel M. Olsen. 2017. Analyzing Dialect Variation in Historical Speech Corpora. *The Journal of the Acoustical Society of America* 142, 1 (2017), 406–421. https://doi.org/10.1121/1.4991009

[29] Pablo Romero-Fresco. 2012. Respeaking in Translator Training Curricula. *The Interpreter and Translator Trainer* 6, 1 (2012), 91–112. https://doi.org/10.1080/13556509.2012.10798831

[30] Pablo Romero-Fresco. 2020. *Subtitling through Speech Recognition: Respeaking*. Routledge, London, UK. https://doi.org/10.4324/9781003073147

[31] Pablo Romero-Fresco and Carlo Eugeni. 2020. Live Subtitling through Respeaking. In *The Palgrave Handbook of Audiovisual Translation and Media Accessibility*, Lukasz Bogucki and Mikolaj Deckert (Eds.). Springer International Publishing, Cham, Switzerland, 269–295. https://doi.org/10.1007/978-3-030-42105-2_14

[32] Berrak Sisman, Junichi Yamagishi, Simon King, and Haizhou Li. 2021. An Overview of Voice Conversion and Its Challenges: From Statistical Modeling to Deep Learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 132—-157. https://doi.org/10.1109/TASLP.2020.3038524

[33] Anselm L Strauss snf Juliet M Corbin. 1990. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Sage Publications, Newbury Park, CA.

[34] Jongseo Sohn, Nam Soo Kim, and Wonyong Sung. 1999. A Statistical Model-Based Voice Activity Detection. *IEEE Signal Processing Letter* 6, 1 (1999), 1–3. https://doi.org/10.1109/97.736233

[35] Matthias Sperber, Graham Neubig, Christian Fügen, Satoshi Nakamura, and Alex Waibel. 2013. Efficient Speech Transcription through Respeaking. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association*. ISCA, Grenoble, France, 1087–1091.

[36] Matthias Sperber, Graham Neubig, Satoshi Nakamura, and Alex Waibel. 2016. Optimizing Computer-Assisted Transcription Quality with Iterative User Interfaces. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*. ELRA, Paris, France, 1986–1992.

[37] Constantin Spille, Birger Kollmeier, and Bernd T. Meyer. 2018. Comparing Human and Automatic Speech Recognition in Simple and Complex Acoustic Scenes. *Computer Speech and Language* 52 (2018), 123–140. https://doi.org/10.1016/j.csl.2018.04.003

[38] Viet Anh Trinh and Michael I. Mandel. 2021. Directly Comparing the Listening Strategies of Humans and Machines. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), 312–323. https://doi.org/10.1109/TASLP.2020.3040545

[39] Emiru Tsunoo, Kentaro Shibata, Chaitanya Narisetty, Yosuke Kashiwagi, and Shinji Watanabe. 2021. Data Augmentation Methods for End-to-end Speech Recognition on Distant-Talk Scenarios. *arXiv* abs/2106.03419 (2021), 5 pages.

[40] Aditya Vashistha, Abhinav Garg, and Richard J. Anderson. 2019. ReCall: Crowdsourcing on Basic Phones to Financially Sustain Voice Forums. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, Article 169, 13 pages. https://doi.org/10.1145/3290605.3300399

[41] Aditya Vashistha, Pooja Sethi, and Richard J. Anderson. 2017. Respeak: A Voice-based, Crowd-powered Speech Transcription System. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, 1855–1866. https://doi.org/10.1145/3025453.3025640

[42] Aditya Vashistha, Pooja Sethi, and Richard J. Anderson. 2018. BSpeak: An Accessible Voice-based Crowdsourcing Marketplace for Low-Income Blind People. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, Article 57, 13 pages. https://doi.org/10.1145/3173574.3173631

[43] Luuk Waes, Mariëlle Leijten, and Aline Remael. 2013. Live Subtitling with Speech Recognition: Causes and Consequences of Text Reduction. *Across Languages and Cultures* 14, 1 (2013), 15–46. https://doi.org/10.1556/acr.14.2013.1.2

[44] Robert A. Wagner and Michael J. Fischer. 1974. The String-to-String Correction Problem. *Journal of the ACM* 21, 1 (1974), 168–173. https://doi.org/10.1145/321796.321811

[45] Xiangdong Wang, Ying Yang, Hong Liu, and Yueliang Qian. 2017. Improving Speech Transcription by Exploiting User Feedback and Word Repetition. *Multimedia Tools and Applications* 76, 19 (2017), 20359–20376. https://doi.org/10.1007/s11042-017-4714-x

[46] Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin andJahn Heymann andMatthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-End Speech Processing Toolkit. In *Proceedings of the 19th Annual Conference of the International Speech Communication Association*. ISCA, Grenoble, France, 2207–2211. https://doi.org/10.21437/Interspeech.2018-1456

[47] Shinji Watanabe, Michael I. Mandel, Jon Barker, and Emmanuel Vincent. 2020. CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings. *arXiv* abs/2004.09249 (2020), 7 pages.

[48] Xinlei Zhang, Takashi Miyaki, and Jun Rekimoto. 2020. WithYou: Automated Adaptive Speech Tutoring With Context-Dependent Speech Recognition. In *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*. ACM, New York, NY, Article 195, 1–12 pages. https://doi.org/10.1145/3313831.3376322