Simplified Audio Production in Asynchronous Voice-Based Discussions

(Anonymized)

ABSTRACT

We introduce SimpleSpeech, an easy-to-use platform for asynchronous audio communication (AAC) based on automatically-generated speech transcripts. Voice communication adds nuance and expressivity to virtual discussions, but its one-shot nature tends to discourage collaborators from utilizing it. SimpleSpeech addresses these concerns through lightweight tools for deleting and inserting content and adjusting pauses in the audio. Qualitative and quantitative results suggest that novice audio producers, such as high school students, experience greater success and decreased mental workload when using SimpleSpeech as opposed to without editing. The linguistic formality in the voice messages was also studied, and found to form a middle ground between oral and written communication media. When applied to contexts appropriate for this level of formality, such as online learning at-scale, SimpleSpeech could serve the crucial purpose of lowering the barrier to participate in audio-based communication and improving engagement in these collaborative environments.

Author Keywords

Speech editing; transcription-based editing; asynchronous audio communication.

ACM Classification Keywords

H.5.2. User Interface: Voice I/O

INTRODUCTION

Asynchronous audio communication (AAC) is rapidly becoming available to mass audiences through social platforms such as WhatsApp, iMessage, and Facebook. While text is still by far the most prevalent mode of communication on the Internet, audio is desirable in many situations because it allows users to deliver more expressive, nuanced messages than text. AAC also holds considerable potential for improving online education, where voice communication has been shown to improve student-student and student-instructor engagement as well as a sense of the instructor's social presence [15, 21, 28].

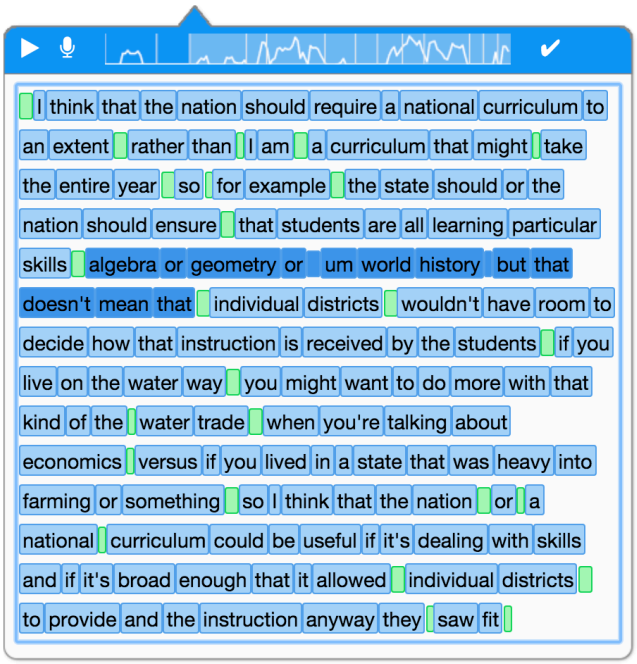

Figure 1. The user interface for SimpleSpeech presents an automatically-generated transcription of a voice comment, which can then be manipulated through normal word-processing operations, such as selection, deletion, and insertion. Word and pause tokens are mapped with audio via time alignment data.

The problem with replacing textual communication with speech, however, is that speakers may face difficulty articulating their ideas vocally. For instance, a 2002 study using Wimba voice boards for discussion forums found that students overwhelmingly preferred text over speech comments, in part because it required them to speak fluently without making errors [20]. Since this problem affects students even in physical classrooms, it could certainly prevent some learners from participating in online oral discussions. AAC platforms in such situations, then, must somehow compensate for the linearity and immutability of audio on the production side.

Our solution is to provide lightweight, easy-to-use editing tools based on automatic speech recognition (ASR)-generated transcripts. Many prior studies have utilized transcription to assist in audio editing [6, 23, 32], but only recently has fast, live editing become possible through advances in ASR technology [7, 24]. We designed and developed an audio production tool, which we call SimpleSpeech, that allows users to delete and insert segments of the recording

in *real-time* by manipulating a tokenized text representation. Our user interface design focuses primarily on simplifying quick word-level editing and visually reinforcing the mapping between the source audio and the text, which helps users edit when transcription errors are present.

Qualitative evidence indicates that SimpleSpeech's simplified interface gave users enough control over the editing process and enabled them to produce more polished audio comments in an online forum discussion. A subsequent quantitative evaluation with high school students showed that the mental workload of recording voice messages was significantly decreased with editing functionality, demonstrating that SimpleSpeech would be a valuable enhancement to online audio communication platforms. Finally, some linguistic characteristics of messages created using AAC are also discussed in comparison to other forms of communication, leading to new considerations and insights on optimal applications of this technology.

RELATED WORK

The linear, sequential nature of voice communication not only precludes skimming and navigation capabilities [10], but can also hamper the speech *production* process. Mistakes in recorded speech are harder to revise than textual typos, mainly due to the lack of lightweight voice editing software [19]. In addition, producing voice is a temporarily linear process which demands the commentator to think and speak simultaneously [19, 35]. Therefore, additional cognitive load arises from the fact that the speaker has to keep speaking to prevent undesirable long pauses. Building on the qualitative implications of these previous works, our study presents a quantitative measure about how such burdens are reduced when the voice production system includes lightweight editing features.

Since lower-level audio waveform editing is an onerous task, speech manipulation tools have been developed that present audio in semantically meaningful higher-level chunks, such as phrases. Acoustic detection of the presence of speech provides binary visual guidance so that users can edit or index the speech recording [1, 14]. On the other hand, a pure acoustic approach had limited resolution of the recognition granularity. Time-aligned automatic speech recognition (ASR) has become a popular tool to achieve the word-level structuring of speech [25, 33]. Compared with acoustic structuring, ASR brings higher temporal resolution with semantic information, but also suffered from high computation load and delay. However, recent technical developments have made ASR faster and more accurate, and we take full benefit of this real-time transcription capability.

Since speech transcription elicits the contents of the recording, researchers have utilized it to assist in visual skimming and navigation. MedSpeak [18] and SCANMail [8] were well recognized as a precursor of such systems that use time-alignment data of the transcript for indexing audio. Since transcription errors tend to obstruct visual comprehension, Vemuri et al. suggested a novel visualization of the transcript that adjusts transcription brightness to the word's ASR confidence score [29].

On the production side, there have been several systems that use a time-aligned transcript for editing audio [23, 32, 34] or video [3, 6]. Among them, Whittaker and Rubin's editing system leveraged users' familiarity to text-editing interfaces, and adopted audio editing in that framework. Since we targeted non-professional users, our interface took the text-editing like approach, but was more geared toward supporting a *live* production process and going beyond editing already-transcribed speech. We thus present versatile and novel features for supporting live production, such as voice insertion, pause extension, and fluid revision of transcription errors.

As in the case of listeners, ASR errors can be detrimental for understanding and skimming audio contents for the purpose of editing [11]. In the MedSpeak interface, Lai et al. provided a a separate graphical window for fixing transcription errors [18]. In a speech production system like SimpleSpeech, though, users could easily get lost between the audio editing and transcription correction modes, so we chose to guide the user's attention through these modes via the movement of the editing caret.

Pauses in speech deliver nuanced meaning such as hesitation or emphasis, so easy and powerful manipulation of pause duration is important. A system called SpeechSkimmer automatically condenses pauses for fast auditory skimming [2]. Other previous systems supported pause editing via a designated button [3] or specialized tags [23]. Rubin et al.'s system used the period key as a shortcut to insert the pause tags, but the duration of the gap was preset and required the use of a separate menu. Our approach is inline with the overall interaction concept of providing a text-like experience, and we adopted the more conventional delete key and spacebar for in-situ removal or arbitrary extension of pause tokens.

DESIGNING SIMPLESPEECH

Building on the capabilities developed in these prior studies, SimpleSpeech is a web-based application for recording and editing short voice messages in a discussion setting. Our design goals were as follows: to support versatile live voice production features with a text-like interface, and to maintain a simple baseline appearance that extends into more complex features through modes and quasi-modes. The appearance of the final SimpleSpeech user interface (UI) is shown in Fig. 1.

Text-Based Editing

There are a wide variety of approaches to audio editing, ranging from waveform-only interfaces such as Audacity and Adobe Audition to semantic speech editors [32] which show only a transcript. For SimpleSpeech, we decided to adopt an interaction paradigm similar to the latter, allowing the user to edit a textual representation which was time-aligned with the audio. This choice was made for two reasons: (1) in general, users are much more familiar with text than with waveform editing; and (2) representing the audio as text would greatly simplify speech editing on the word level, in contrast to the greater complexity of millisecond-level waveform operations.

Accordingly, the UI for SimpleSpeech devotes the majority of the editing panel to the transcription. Users interact with the

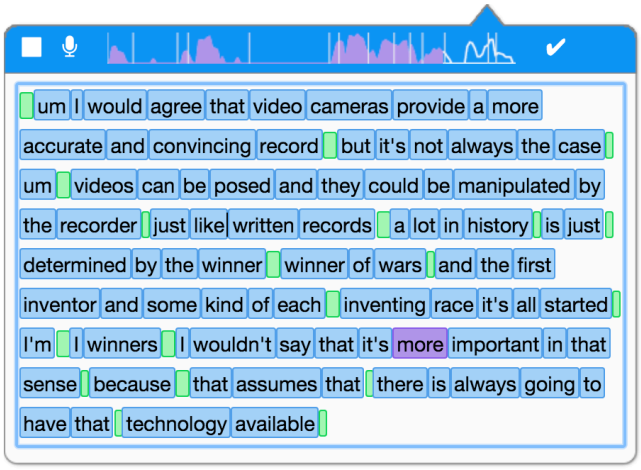

Figure 2. The waveform visual at the top of the SimpleSpeech editor does not support direct millisecond-level editing; instead, it serves as a visual cue connecting the transcript to the working recording. For instance, the waveform is highlighted during playback in tandem with the current word that is being played.

text by selecting, editing, and deleting *tokens*, which are colored blue for words and green for pauses and unrecognized sounds. In addition to deleting tokens using the Backspace key, green pause tokens can be inserted or extended using the space bar. Deleting and inserting tokens results in the appropriate modifications automatically applied to the working audio file.

Reinforcing the Connection Between Audio and Text

We chose to include a waveform visualization as part of the UI in order to remind the user that he or she is ultimately manipulating audio, not text. The waveform incorporates several subtle indications of the mapping between its contents and the transcription, including highlighting the audio corresponding to the current selection of tokens and animating deletions and insertions. We found the presence of a waveform to be a helpful visual indicator of the purpose of SimpleSpeech, although it was not functionally useful *per se*. Without the waveform, users' inclination was to disregard the original speech and use the system as a dictation tool.

Another strategy to reinforce the parallelism between the source audio and the transcription is to highlight the words in the transcript as they are spoken during playback, as shown in Fig. 2. The waveform renders the portion of audio that has already been played in a purple color, which is also used to render the token currently being played back.

Editing Audio vs. Fixing Transcription Errors

Our use of text as a proxy for editing audio rendered it necessary to clearly delineate the capabilities of SimpleSpeech in comparison to a word processor. For instance, direct text input is disallowed in the transcript area to avoid inserting words not present in the original recording. (The inability to move the caret within the tokens visually confirms that the

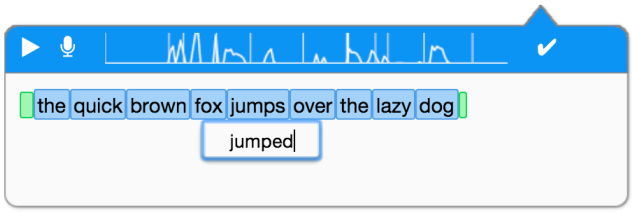

Figure 3. To keep the user interface from becoming cluttered with secondary functionality, the transcription editing feature was implemented as a modal interaction. The pop-up box shown above "opens" the selected tokens for text editing in a separate control element, thus notifying the user that he or she is no longer directly editing the audio.

transcript cannot be edited without correspondence to the audio.) However, we found the capability to edit text of individual words in the transcript to be desirable, especially in the case of ASR errors. The transcription editing functionality is available in a separate mode, accessed by pressing the Return key, and insulates the editing within single words to avoid undermining the cohesiveness of the tokens. (If the user started typing while one or more tokens were selected, this automatically activated the transcription editing mode as well; some pilot users found this more intuitive than using the Return key.)

During pilot testing the need arose for a fast way to play back and pause the audio; however, the conventional keyboard shortcut for playback, the spacebar, was already in use for the pause insertion feature. We resolved this problem by using Shift+space for playing and pausing. In effect, the playback functionality was encapsulated as a *quasi-mode*, a set of distinct features that are accessed while performing a constant action (in this case, pressing the Shift key) [22]. The modal design helps prevent beginning users from being overwhelmed with possible actions while allowing more advanced users rapid access to the higher-level features.

Pilot Study and Design Improvements

We followed an iterative procedure to progressively improve the design and interactions of SimpleSpeech. After building an initial prototype of the application, an informal pilot test was conducted with 5 participants. Each user was given a brief introduction to the software and shown how to use the basic features, then given the scenario of creating an audio response to a written claim on an online forum. (The prompts used in the tests were adapted from the GRE Pool of Issue Topics.) pilot user demographic information here After using the software, users were interviewed to obtain feedback on the prototype, yielding the following modifications:

Pause manipulation. Another important finding in the pilot study was the importance of being able to introduce and adjust pauses between words, not just to remove them. These gaps in the audio help make natural-sounding cuts between audio clips as well as to punctuate claims (e.g., the end of a sentence). The original system only allowed the user to delete pauses, so we added a spacebar action to insert a zero audio signal or fragment of silence from the original audio resource into the rendered message.

Tokenization. The way we had tokenized the transcript originally, the user was able to "enter" the tokens with the caret and change their contents. However, this inline transcript editing behavior confused the pilot study participants, who tried to insert new unrecorded content by typing. We endeavored to clarify these delineations in the next iteration by disabling direct alphanumeric input to the transcript view, and shifting the transcription editing functionality into a modal interaction.

Implementation

Our text-based approach requires a reliable transcription as well as time intervals corresponding to each word; both of these requirements are fulfilled by the IBM Watson Developer Cloud speech-to-text transcription service, which is reported to have a word error rate of 10.4% [26]. For the sake of the cross-platform compatibility, the application was implemented as a web app written in JavaScript, HTML, and CSS. Editing is accomplished by maintaining a data model consisting of one or more user-created audio resources as well as a list of timestamps, each of which links a token in the text area to a time interval within an audio resource. When the user plays back the message, the data model "renders" a complete audio recording by stitching together the audio from each timestamp.

QUALITATIVE EVALUATION

The interaction paradigm of SimpleSpeech was tested in a qualitative assessment to determine (1) the practicability of a lightweight text-based audio editor, (2) the effects of minor transcription errors on audio consumption and production, and (3) the implications of being able to edit audio in an asynchronous online discussion.

Participants were introduced to the functionality of the system, then given two untimed tasks. First, to simulate an asynchronous audio discussion, the test users were asked to listen to an audio comment left by the previous tester and create an original audio response. Next, they received a different, textual prompt and created an audio comment which would be consumed by the next user. In both cases the user was asked to edit his or her recording to be polished and clear. The participants were interviewed at the end of the test; these interviews were transcribed, conversational elements filtered out, and the remaining sentences analyzed via open coding followed by flat coding. Cohen's κ was .78, indicating high reliability between the two coders.

The sample for the study consisted of 9 test subjects (4 male, 5 female; henceforth denoted $P_1, P_2, \ldots, P_9$). All participants were native English speakers. Two individuals, P_2 and P_3, were professional media editors who provided technical feedback and a comparison to pure audio editing; the remainder were interns and high school students.

Results

The coding process resulted in the following themes identified from the user feedback:

The text-based editing paradigm provides sufficient control to render waveform manipulation unnecessary. Most non-professional users felt SimpleSpeech gave them "plenty of control" over the editing process (P_4, P_5, P_6, P_8). The professional editors did note that most people in their field would not find SimpleSpeech adequate for their needs; but, as P_2 conceded, the intended market users "don't have to play with the settings which is why they don't use a professional audio editor." Most participants characterized the editing experience as being a text-focused one, suggesting that the translation to text was in fact a useful proxy for editing audio. The text modality was described as "more accessible, more doable" than pure waveform editing, which could be "scary for people who don't do video stuff" (P_3, P_7).

The primary use of lightweight voice editing is to make fine-grained rather than large-scale adjustments. The most commonly-used manipulation during the qualitative study was the removal of disfluencies (P_1, P_2, P_4, P_5, P_7), followed by pause deletion (P_2, P_3, P_5, P_6, P_8). Only P_1 and P_8 edited large chunks of audio by deleting or rerecording, and P_8 reported doing so only to improve the smoothness of a smaller change in a sentence. Perhaps because SimpleSpeech was presented as a tool to be briefly used to "clean up" recordings, participants focused on removing the "embarrassing" and "awkward" sounds (P_1, P_5).

Transcription is a helpful aid for listening to audio comments despite occasional errors. In many cases, the transcription proved to be an essential element of both the production and the consumption interfaces. To determine the effect of errors in the transcript on listeners, the previous participants' comments were displayed to users with an unedited, errorful ASR transcript. Despite the occasional errors, users still found the transcript to be helpful in allowing them to "see all the points [the speaker was] making instead of having to remember them" (P_4, P_6). For some users, the transcription caused no problems in comprehension, while others experienced errors that required them to pay more attention to the audio (P_8). On the whole, ASR succeeded in "getting the basic idea across" (P_3) but could not stand alone without the original recording.

The linearity of audio leads to a pressure to organize one's thoughts during recording. P_4, P_7, and P_9 described a "psychological sort of ... need to get it all out, and the fact that it won't necessarily be as organized there." Another tester, P_5, had "a tendency to get like a blank slate" in which he "couldn't think of anything to say." The elevated mental task load that P_5 describes could be inherent in oral discussion; P_9 noted that "[it] might just be the fact that I was recording," and that "editing would make it nicer." Because this phenomenon was present despite the ability to edit, we decided to analyze the task load aspect of using SimpleSpeech in the quantitative study.

Awareness of the recipient and the editability of the audio drive up the quality of contributions. Four users mentioned the formality of their recordings (P_1, P_5, P_7, P_9), which they attributed to "an expectation" to edit, given that "I know that I've had that opportunity and someone else would know that I

had that opportunity" (P_8). The speakers' inclination to consider their listeners is exemplified by P_9, when asked why she was motivated to edit her messages:

> Personally I'm editing to express myself a little more in a polished way when I'm writing.... especially if I know someone else is going to review it and be able to respond, I want to make sure I'm as clear as possible and as concise in a way that doesn't really come across when I'm talking.

Listening to another participant before initiating their own comment may have been a factor in determining the users' performance, since the exposure "gave ... an understanding of how long of a comment, or what kind of direction people were trying to take the discussion" (P_9). Editing contributed to the increased quality as well: "Since you have the ability to edit things, it feels like you're talking to somebody who's prepared a point or a conversational view" (P_5). We chose to explore this phenomenon quantitatively to determine if it was real or simply perceived by the speakers, and to what extent it was affected by the ability to edit.

QUANTITATIVE EVALUATION

For our second, quantitative experiment, we intended to assess the efficacy of SimpleSpeech in particular, and also to measure the usefulness of audio editing tools in general for educational discussions.

Procedure

Two between-subject dimensions were studied: students versus teachers, as well as the formality of initial stimulus recordings (see "Stimuli and Formality Measures" below). In addition, the dimension of no-editing versus editing was studied on a within-subject basis. Participants in the study were given two task parts in random order: recording messages without editing functionality (the No Editing, or NE task) and using SimpleSpeech (the Editing, or E task). Each task consisted of "discussion threads," in which users read a prompt statement, listened to another person's opinion on the issue, then produced an original response. Participants responded to two threads for each task, for a total of four messages of about one-minute duration each. (Before starting the E part, participants were given a standardized tutorial to learn how to edit using SimpleSpeech.)

After each task, the NASA Task Load Index (NASA-TLX) questionnaire was used to quantify the pressure or mental task load of producing a voice message [12]. NASA-TLX is a subjective analytical tool that measures task load along six dimensions: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration. After rating the level of each aspect of mental workload from 1 (least workload) to 20 (greatest workload), the subject is asked to compare the scales pairwise to produce a weighted TLX value representing the overall pressure during a situation. Participants in the study completed the TLX procedure once after each task to obtain comparisons between the mental workload induced by no-editing and editing situations.

The quantitative study was conducted at a small suburban public high school in central Ohio with 28 volunteer participants (16 students, ages 16-18, and 12 teachers; 13 male, 15 female). This location was ideal for the study because the sample contained a variety of learning and speaking styles as well as different aptitudes for technology and discussion.

Stimuli and Formality Measures

The initial stimulus recordings for each of the prompt statements were generated by a group of five initial volunteers. Since the qualitative study had indicated the possibility that prior exposure to other individuals' messages could affect users' perception of formality in the discussion, we divided the stimuli into formal and informal sets. Half the participants of the study (Group A) listened to only formal recordings, while the other half (Group B) listened to only informal ones. We hypothesized that the participants in Group A would produce more formal messages due to the stimuli they received.

The criterion used for formality was the F-score, a measure of contextuality introduced by Heylighen and Dewaele in 2002 [13]. The F-score is a purely textual metric based on the frequencies of various parts of speech in a text: nouns, adjectives and prepositions decrease the contextuality and increase the F-score since they are independent of the circumstances around the text, while deictic words such as verbs, adverbs, pronouns, and interjections increase contextuality and decrease the F-score. For our stimulus recordings, the initial participants were asked to plan and edit some of the comments and improvise on the others. After splitting the resulting messages by formality, the average F-score was 53.7 for the Group A messages and 49.4 for the Group B messages, reflecting the greater contextuality of the recordings produced on-the-fly. Group A stimuli also tended to use longer words than those for Group B (4.62 versus 4.38 letters) and tended to be more concise (113 versus 193 words). After obtaining and categorizing these messages, the voices were anonymized by adjusting the pitch randomly.

Stopped here. will return back to this place tmr.

Results

We could understand effects of the live voice editing features by analyzing system logs, speech contents, and the task load survey data. The results fell into the following three categories.

Utilization of Editing Features

The length of speech were about 1 min (M = 57.2, SD = 31.2) per session. The mean word counts of speech per session was 122.3 (SD = 59.1). There was no statistically significant difference of the speech length or word counts by different conditions.

As in the qualitative study, most participants appreciated and took advantage of the ability to edit their messages. They found the interface intuitive and natural thanks to the familiarity with the text-editing interfaces. A few participants mistakenly hit the Delete key on the token when they wanted to fix a transcription error, resulting in the permanent deletion of

the audio token. However, The emphasis in the tutorial that the Delete key deleted the audio permanently did help other participants avoid making this mistake.

On average, users made about 17 edits to each comment (inserting a new recording, inserting a pause, deleting words, or deleting a pause). Of these changes, the vast majority were subtractive: 7 word deletions and 6.3 pause deletions per message. This was again consistent with the findings of the earlier study, which had shown an inclination to remove disfluencies and "awkward" hesitations from the recordings. IMPORTANT!. Always report standard deviation for each statistics. Also report the mean down to the first place of decimal. The formal way to do this is: On average, users made about 17.X edits to each comments (SD = XX.X, inserting a new ....).{...} 7.X word deletions (SD = X.X) and 6.3 pause deletions per message (SD = X.X).

How many insertions?

Indicating how much the users relied on the editing tools, the messages from the E task showed significant differences in the occurrences of pauses and disfluencies.

I would remove this paragraph or briefly discuss in the discussion. It's hard to report such a qualitative implication without having data about the transcription error: Ideally, participants in the E task would have edited both the transcription and the voice to be free of errors, but due to time constraints on participation we discouraged the users from correcting transcript errors (which turned out to be more numerous than expected, especially because of the conversational style). Incorrect transcriptions were problematic for editing in general: Since the associated timestamps were also incorrect, the edits on that segment of audio could produce undesirable results.

Another misconception we observed in a few participants was a tendency to treat SimpleSpeech as a dictation tool. These users paused for long periods of time during recording sessions and neglected to play back the messages during editing. Furthermore, their inclination after stopping a recording session was to go back and correct transcription errors so that the visual representation made sense.

Task Load

Since the NASA-TLX scale is subjective, it does introduce variability between participants due to the differences between their perceived skill at the task [12]. For instance, one participant could rate the recording task at a 3 out of 20, while another could rate the very same task at a 15. Therefore, the strongest comparisons of task load were made in the within-subject dimension, which was the ability or inability to edit.

Overall, the students reported significantly *lower* levels of mental task load or pressure during the E task than the NE task (M_E = 8.7, SD_E = X.X compared to M_{NE} = 10.8, SD_{NE} = X.X, $p < 0.02$ using a two-tailed t-test). The values for the individual components of the TLX, shown in Table 1, yielded the following contributory dimensions on the TLX questionnaire:

- *Temporal demand*. Students rated the temporal demand at 7.8 for the E task, significantly less than the NE rating of 10.2 (t = 2.29, p = 0.037). As described by the TLX form, temporal demand refers to "time pressure due to the rate or pace at which the tasks or task elements occurred" [12]. Students verbally described the increase in time demand reported on the TLX in terms of having to think of words quickly, with the knowledge that every second not filled with speech would be an embarrassing silence.

- *Performance*. Students felt more concern about the quality of their messages in the NE task, rating it at 10.0 compared to 8.3 for the E task ($p < 0.10$). Just as the participants in the prior qualitative study had articulated a desire to make their messages better for the sake of their listeners, the students also evidently wanted to improve their recordings in the NE task. The inability to do so resulted in elevated task load due to performance, while for the E task the stress was lower because they were afforded the chance to correct their mistakes. However, it is worth noting that even despite the capability to edit, the student participants still rated Performance close to the middle of the scale, perhaps representing self-consciousness or comparisons with the stimulus recordings.

- *Effort*. Similarly to performance, students reported having to work significantly harder in the NE task to complete it to their desired level (rated 11.6 compared to 9.1 in the E task, t = 2.79, p = 0.014). This increased effort could correspond to the additional mental activity which had to be expended in order to generate speech fluently and without excessive hesitation.

	Students (N = 16)		**Teachers** (N = 12)	
Task	*E*	*NE*	*E*	*NE*
Mental Demand	9.6	11.1	11.4	10.8
Physical Demand	3.7	2.6	4.0	2.8
Temporal Demand	7.8	10.5*	7.5	10.0
Performance	8.3	10.0$^+$	8.5	9.7
Effort	9.1	11.6*	9.8	10.4
Frustration	7.8	8.9	8.4	10.0
Total (weighted)	8.7	10.8*	9.5	10.6

Table 1. The mental work load ratings reported by students and teachers from recording voice messages. *E* and *NE* refer to the tasks in which editing was allowed and disallowed, respectively. Each value ranges from 1 to 20, indicating the amount that the given descriptor contributed to the participants' overall task load. ($^+$ – $p < 0.10$, * – $p < 0.05$, paired two-tailed comparison of *E* and *NE*)

While the teachers also reported slightly lower average workload levels in the E task, as shown at the right of Table 1, this difference was not significant. In fact, 7 of the 12 participating teachers actually rated the E task as requiring a higher workload than the NE task. This subset of the teachers, 5 of whom were in Group A, reported an average task load greater in the E task than the NE task for *all* dimensions, especially Mental Demand, Performance, and Frustration. The reason for this rating, these teachers explained, was that the availability of the editing tools caused them to feel more worried

about their performance. Editing in turn required them to expend more effort to preserve the existing fluidity of their messages.

Interestingly, this particular group of teachers produced more formal messages than the other teachers (mean F-score 56.6 compared to 53.1), with longer words (4.58 compared to 4.36 letters), and fewer disfluencies (1.3 compared to 2.1 per 100 words). Upon further inspection, the student participant group also contained members who rated the E task as more demanding than the NE task, though fewer in number (4 out of 16); these students also produced much more formal messages than their peers (57.9 compared to 53.5). These participants could have had more experience speaking extemporaneously or felt less inclined to speak conversationally, ultimately leading to SimpleSpeech not being as useful to them.

Overall, the fact that the differences in perception of workload varied so much among teachers indicates that they were not as heavily affected by the ability to edit as the students, who clearly appreciated the security that SimpleSpeech offered.

Formality

Contrary to the hypothesis that prior exposure to audio messages would affect the formality or linguistic traits of new messages, the F-scores of the participants' output was unrelated to the group they were in, as shown in Fig. 2. The average F-score for students was higher for Group A (55.83) than for Group B (53.42), which is a considerable difference in terms of the F-score's scale but not statistically significant. The F-scores for teachers were almost distinguishable, with a difference of only 0.72. In other words, the formality of the recordings was not affected by the stimulus message or even whether the participant was a teacher or a student. Considering that the F-score measures contextuality between the speaker and the audience, and that its value was not affected by the context given before the tasks, the principal source of variation in F-score must have been personal preference in the medium and the scenario of an online forum discussion.

FORMALITY COMPARISON

Given that the F-scores of SimpleSpeech messages were roughly normally distributed and not heavily affected by the experimental conditions, the average F-score of 54.8 is likely to be characteristic of general AAC discussions under similar conditions. Contextuality in the online voice-based forum scenario could be highly indicative of AAC's potential applications, and to our knowledge this trait has not been studied extensively. The closest related studies have pertained to other forms of computer mediated communication (CMC), especially textual ones such as SMS, email, or Facebook posts. For example, Kiesler, Siegel, and McGuire [16] found more equalized group participation and more uninhibited expression of opinions in synchronous text-based CMC than in face-to-face discussions. Asynchronous CMC, similar to a discussion board, induces more prosocial behavior and, in fact, more informal communication styles over time than face-to-face [31]. On the other hand, formality and politeness in emails has been shown to increase as the social distance, status gap, and importance of a request increase [5].

	Students	
Group	*A*	*B*
Formality (F-score)	55.83	53.42
Word Length	4.40	4.44
Disfluencies (per 100 words)	1.59	2.38
Word count	100.66	140.47
Speaking rate	130.39	114.75
	Teachers	
Group	*A*	*B*
Formality (F-score)	54.74	55.45
Word Length	4.50	4.47
Disfluencies (per 100 words)	1.27	1.41
Word count	155.38	130.17
Speaking rate	136.58	137.66

Table 2. Various metrics describing the formality of the audio messages produced by each participant group. Group A listened to more formal initial stimulus recordings than Group B. There were no significant differences in these criteria between the groups, indicating that formality was dependent on the general context of AAC as well as the speaker's preference.

How AAC fits into the complex hierarchy of social dynamics on various platforms is still unknown, so we conducted a comparison of the voice messages composed during this study with corpora of different media. The SimpleSpeech text, the focal point of the comparison, contained – words from – messages. For written documents, we used several sections of the well-known Brown corpus to compile general categories of text: nonfiction, fiction, and technical writing (consisting of government documents, scientific articles, and news) [17]. We obtained chatroom text from the `nps_chat` corpus, face-to-face conversation data from the `webtext` corpus, and telephone data from the `switchboard` corpus, all available as part of the Natural Language Toolkit (NLTK) [4]. Finally, we also analyzed email communication in non-spam messages from the Enron corpus [27], as well as a corpus of Twitter posts [9].

Results

The results of this comparison, shown in Fig. 4, illustrate the middle-ground that AAC takes relative to oral and written media. The least formal and most contextual corpora were those based on oral communication (with the notable exception of web chat messages), while the most formal and least context-dependent were the written texts, including email and Twitter posts. We will note three additional explanations for the formality of each medium based on the ordering of the corpora:

Speaker-audience relationship. Since the F-score is inversely related to contextuality, it is reasonable that the chat and telephone corpora had the lowest F-scores because the participants knew each other and were conversing on a one-to-one basis. On the other hand, the written forms of communication (with the exception of email) were more formal because the audience was defined more loosely and not necessarily acquainted with the speaker. AAC using SimpleSpeech was

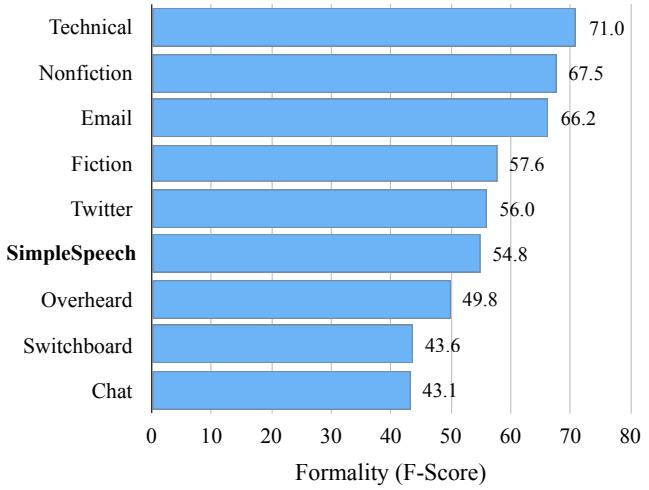

Figure 4. The formality of corpora in different genre and media. The messages produced using SimpleSpeech during the quantitative study are intended to reflect general AAC discussion characteristics, and seem to be more formal than other spoken forms of communication but not as formal as email.

more closely related to the latter condition (as an online forum discussion), which probably contributed to its greater formality compared to the other spoken corpora.

Immediacy of communication. The tendency to speak or write more contextually when the recipient replies immediately explains why the online chat text, though written, was more contextual and less formal than the oral corpora. It also justifies the fact that the email corpus was more formal than all of the other direct communication media. Again, AAC falls toward the more formal end of this spectrum because there is little temporal proximity between the speaker and the audience.

Tendency toward verbosity. Media that pressured the creator to be brief or precise were more formal and less contextual. For instance, writing technical documents requires the preferential use of nouns over pronouns to maximize clarity. Twitter messages are, of course, limited to 140 characters, leading to a greater concentration of meaning that favors less contextual words. For AAC, therefore, the ability to edit could influence the contextuality if discussion members were pressured to trim down their recordings. For our study, however, the participants were not affected by verbosity; though non-edited recordings had on average 10% more words than edited ones, these edits were more concentrated on removing disfluencies than improving concision.

DISCUSSION

In this study two forms of responding to pressure in a communication task were measured: the mental workload involved in completing the task and the degree of formality in the messages created in the task. Using this information, we will evaluate the strengths and weaknesses of SimpleSpeech as a tool for enabling AAC as well as the viability of AAC in educational and collaborative contexts.

Imbalance Between Speaker and Listener

As Grudin notes, it is critical for collaborative software to spread the burden of usage equally on its constituent members. For instance, he cites email as a medium in which "everyone generally shares the benefits and burdens equally" [10]. On the other hand, voice applications create inequality between speaker and listener since the former must expect that the latter will listen thoroughly and carefully to the message, a relatively slow task compared to reading.

However, the premise of SimpleSpeech is that the bias toward the speaker is reversed. ASR transcription can greatly facilitate the listener's task, as has already been demonstrated [8, 30], bringing the workload down and closer to that of reading. Meanwhile, students who record messages could experience a *greater* workload relative to writing because of the linearity of audio, which prevents them from correcting mistakes after the fact and thereby elevates the pressure to do well the first time.

SimpleSpeech was demonstrated to be a useful counterbalance in situations where the speaker's workload is elevated. In the qualitative study, some users noted the pressure "to have organized thoughts" and to "sound composed more" during recording, but that "editing would make it nicer because you can go back and fix the mistakes" (P_2). Furthermore, the level of control was just right for most users: since they focused on deleting the disfluencies and pauses in their speech, the word-tokenized editor for the most part provided exactly the information needed to quickly delete undesirable sounds. For the few users who did want to edit on a larger scale, the audio insertion feature was deemed helpful as well.

In the quantitative evaluation, we found strong evidence to support the use of SimpleSpeech, especially for students. There was a significant decrease in task load on students when given the capability to edit, even in spite of the added time required to listen to the message and perform the editing. The especially compelling factor is the Performance dimension, which decreased by 17% in the Editing task. Several students also mentioned relief at the fact that the voice comments they were producing in the NE task would be anonymized; considering that most applications of AAC would not afford them this security, editing would become even more important to students' comfort level in voice communication. On the other hand, teachers did not find SimpleSpeech as useful, probably because they already perceived their recordings as being of acceptable quality. One teacher reasoned that he was "already used to hearing [his] own voice" from lecturing, a medium where statements cannot be retracted as easily as with SimpleSpeech. However, many teachers did use the editing tools, even though their workload levels were not significantly different with or without this opportunity. This would indicate that the editing tools are a valuable option for producers to have, but users should not be obligated to use them.

Editing and the Quality of Discussion

It is critical to the success of general-purpose AAC that the formality of discussion be controlled to some extent, so that collaborators feel willing to participate. Luckily, the formality of the discussions simulated in this study was not significantly affected by the experimental conditions, indicating that

for the most part, users are likely to adopt their own style for audio messages without being pressured by the discussion context. Furthermore, for students this impetus toward quality is not as problematic, since they felt more relaxed rather than more stressed with editing functionality. If discussion quality were driven up by artificial means, however, such as by grading students on the eloquence of their comments or evaluating employees on the basis of their online interactions, then individuals might gravitate toward "safer" modes of communication over which they feel more control (namely, text). Proper acquaintance with audio editing capabilities is essential for AAC's survival under these pressures toward high-quality production.

Implications for Formality in AAC

AAC has the potential to greatly improve communication in educational, corporate, governmental, and personal contexts. However, it is important that the social dynamics of AAC be taken into account in order to avoid unproductive, undesirable, or unwilling participation in collaborative environments. For instance, discussion groups within an online course would be an ideal use of AAC using SimpleSpeech because students could send messages to a well-defined audience, thereby compensating for the additional formality imposed by the spatiotemporal distance between the participants. The editing tools would also drive students to produce better discussion input, increasing productivity and enhancing the learning experience. On the other hand, enabling editing for personal communication, such as WhatsApp voice messages, would be detrimental to the desired informal speaking style of the platform. Since the contextuality demanded by each situation is different, future audio-based collaboration platforms must consider the factors presented here and tailor their functionality accordingly.

CONCLUSIONS

SimpleSpeech represents a novel contribution to asynchronous audio collaboration because of its lightweight, easy-to-use live editing tools. Its functionality alleviates the pressure associated with the linearity of audio because users have the capability to easily remove superfluous words, pauses, and sounds as well as insert new phrases, all after the fact. Furthermore, we designed SimpleSpeech to be as intuitive as possible and to emphasize the synergy of audio and text, from the visual cues provided by the waveform to the modal interface for correcting transcription errors. Students' use of these editing tools resulted in them feeling much more comfortable producing comments for a general audience than they were without SimpleSpeech. The true utility of this software, then, was to (at least partly) un-linearize audio, even making it more text-like.

Because studies of the linguistic and social characteristics of computer-mediated communication have been mostly limited to textual interactions, we also explored the formality and contextuality of AAC. Our finding that it was roughly in between spoken and written media is not discouraging *per se*; however, the relatively formal characteristics of AAC must be taken into account before such a system is implemented in practice. Nevertheless, we feel that the small-scale edits that users engaged in during this study are reassuring for potential applications of AAC. Removing disfluencies and pauses allows users to feel comfortable with their recording while maintaining the spontaneity of thought in a spoken message.

The results of the qualitative study point to new directions for improving SimpleSpeech. For instance, on initial exposure to the application users initially tended to focus preferentially on the text instead of on the voice. Slightly different visual layouts of the application, such as overlaying or juxtaposing the transcription on a more prominent waveform, could help users understand better that the text is a secondary tool. Another possible feature could be automating certain edits, such as removing disfluencies and hesitations, to improve efficiency and edit quality even further. Additionally, the findings in our quantitative study revealed promising trends concerning the benefits of AAC for online discussion, but may need a larger pool of test participants to attain statistical significance.

Our hope in developing SimpleSpeech is that asynchronous audio communication will gain greater usage in education and other collaborative settings. With the combination of ASR transcription for listeners and low-barrier editing tools for speakers, voice-based communication tools can engage students and improve the quality of collaboration on the Web.

ACKNOWLEDGMENTS

Anonymized

REFERENCES

1. Stephen Ades and Daniel C Swinehart. 1986. *Voice annotation and editing in a workstation environment*. XEROX Corporation, Palo Alto Research Center.
2. Barry Arons. 1993. SpeechSkimmer: Interactively Skimming Recorded Speech. In *Proceedings of the 6th Annual ACM Symposium on User Interface Software and Technology (UIST '93)*. ACM, New York, NY, USA, 187–196. DOI: http://dx.doi.org/10.1145/168642.168661
3. Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for Placing Cuts and Transitions in Interview Video. *ACM Trans. Graph.* 31, 4, Article 67 (July 2012), 8 pages. DOI: http://dx.doi.org/10.1145/2185520.2185563
4. Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.
5. Thomas Cho. 2010. Linguistic Features of Electronic Mail in the Workplace: A Comparison with Memoranda. *Language@Internet* 7, 3 (2010).
6. Juan Casares et al. 2002a. Simplifying Video Editing Using Metadata. In *DIS '02 Proceedings*. London, 157–166.
7. Janet M. Baker et al. 2009. Developments and directions in speech recognition and understanding, Part 1. *Signal Processing, IEEE* 26, 3 (2009).

8. Steve Whittaker et al. 2002b. SCANMail: a voicemail interface that makes speech browsable, readable and searchable. *CHI Letters* 4, 1 (2002).
9. Alec Go, Richa Bhayani, and Lei Huang. 2009. *Twitter Sentiment Classification using Distant Supervision*. Technical Report. Stanford.
10. Jonathan Grudin. 1988. Why CSCW applications fail: Problems in the design and evaluation of organizational interfaces. *ACM Conference on Computer-Supported Cooperative Work* (1988).
11. C Halverson, D Horn, C Karat, and John Karat. 1999. The beauty of errors: Patterns of error correction in desktop speech systems. In *Proceedings of INTERACT99*. 133–140.
12. Sandra G. Hart and Lowell E. Staveland. 1988. Development of NASA-TLX (Task Load Index): Results of empirical and theoretical research. In *Human Mental Workload*, P. A. Hancock and N. Meshkati (Eds.). North Holland Press, Amsterdam.
13. Francis Heylighen and Jean-Marc Dewaele. 2002. Variation in the contextuality of language: an empirical measure. *Foundations of Science* 7 (2002), 239–340.
14. Debby Hindus and Chris Schmandt. 1992. Ubiquitous Audio: Capturing Spontaneous Collaboration. In *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work (CSCW '92)*. ACM, New York, NY, USA, 210–217. DOI: http://dx.doi.org/10.1145/143457.143481
15. Philip Ice, Reagan Curtis, Perry Phillips, and John Wells. 2007. Using asynchronous audio feedback to enhance teaching presence and students' sense of community. *Journal of Asynchronous Learning Networks* 11, 2 (2007).
16. Sara Kiesler, Jane Siegel, and Timothy W. McGuire. 1984. Social Psychological Aspects of Computer-Mediated Communication. *Amer. Psychologist* 39, 10 (1984), 1123–1134.
17. Henry Kučera and W. Nelson Francis. 1967. *Computational Analysis of Present-Day American English*. Brown University Press, Providence.
18. Jennifer Lai and John Vergo. 1997. MedSpeak: Report Creation with Continuous Speech Recognition. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems (CHI '97)*. ACM, New York, NY, USA, 431–438. DOI: http://dx.doi.org/10.1145/258549.258829
19. Philip Marriott. 2002. Voice vs text-based discussion forums: An implementation of Wimba Voice Boards. In *World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, Vol. 2002. 640–646.
20. Philip Marriott and Jane Hiscock. 2002. Voice vs Text-based Discussion Forums: An Implementation of Wimba Voice Boards. In *Proc. E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, M. Driscoll and T. Reeves (Eds.). Chesapeake, VA.
21. Jody Oomen-Early, Mary Bold, Kristin L. Wiginton, Tara L. Gallien, and Nancy Anderson. 2008. Using asynchronous audio communication (AAC) in the online classroom: a comparative study. *Journal of Online Learning and Teaching* 4, 3 (2008).
22. Jef Raskin. 2000. *Humane Interface, The: New Directions for Designing Interactive Systems*. Addison-Wesley.
23. Steve Rubin, Floraine Berthouzoz, Gautham J. Mysore, Wilmot Li, and Maneesh Agrawala. 2013. Content-Based Tools for Editing Audio Stories. In *UIST '13*. 113–122.
24. George Saon, Hong-Kwang J. Kuo, Steven Rennie, and Michael Picheny. 2015. The IBM 2015 English Conversational Telephone Speech Recognition System. *Interspeech* (2015).
25. Christopher Schmandt. 1981. The Intelligent Ear: A Graphical Interface to Digital Audio. In *Proceedings, IEEE International Conference on Cybernetics and Society, IEEE*.
26. Hagen Soltau, George Saon, and Tara N. Sainath. 2014. Joint training of convolutional and non-convolutional neural networks. In *Proceedings of the IEEE Intl. Conference on Acoustic, Speech and Signal Processing*. Florence, 5572–5576.
27. Will Styler. 2011. *The EnronSent Corpus*. Technical Report 01-2011. University of Colorado at Boulder Institute of Cognitive Science, Boulder, CO.
28. Chi-Hsiung Tu and Marina McIsaac. 2002. The Relationship of Social Presence and Interaction in Online Classes. *Amer. Journal of Distance Education* 16, 3 (2002).
29. Sunil Vemuri, Philip DeCamp, Walter Bender, and Chris Schmandt. 2004a. Improving Speech Playback Using Time-compression and Speech Recognition. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '04)*. ACM, New York, NY, USA, 295–302. DOI: http://dx.doi.org/10.1145/985692.985730
30. Sunil Vemuri, Philip DeCamp, Walter Bender, and Chris Schmandt. 2004b. Improving Speech Playback Using Time-Compression and Speech Recognition. *CHI Letters* 6, 1 (2004).
31. Joseph B. Walther. 1995. Relational Aspects of Computer-Mediated Communication: Experimental Observations over Time. *Organization Science* 6, 2 (1995), 186–203.
32. Steve Whittaker and Brian Amento. 2004. Semantic Speech Editing. *CHI Letters* (2004), 527–534.

33. Lynn Wilcox, Ian Smith, and Marcia Bush. 1992. Wordspotting for Voice Editing and Audio Indexing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '92)*. ACM, New York, NY, USA, 655–656. DOI: http://dx.doi.org/10.1145/142750.150715

34. Dongwook Yoon, Nicholas Chen, François Guimbretière, and Abigail Sellen. 2014. RichReview: blending ink, speech, and gesture to support collaborative document review. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*. ACM, 481–490.

35. Dongwook Yoon, Nicholas Chen, Bernie Randles, Amy Cheatle, Corinna E. Loeckenhoff, Steven J. Jackson, Abigail Sellen, and François Guimbretière. 2016. Deployment of a Collaborative Multi-Modal Annotation System for Instructor Feedback and Peer Discussion. In *Proceedings of the 19th ACM Conference on Computer Supported Cooperative Work & Social Computing*. ACM.