ScriptFree: Designing Speech Preparation Systems with Adaptive Visual Reliance Control on Script

Jeungmin Oh
KAIST
Daejeon, South Korea
jminoh@kaist.ac.kr

Darren Edge
Microsoft Research
Cambridge, United Kingdom
darren.edge@microsoft.com

Uichin Lee
KAIST
Daejeon, South Korea
uclee@kaist.ac.kr

CHI'20,, April 25–30, 2020, Honolulu, HI, USA
ACM 978-1-4503-6819-3/20/04.
https://doi.org/10.1145/3334480.3382896

Abstract

Manuscript speech is a common type of speech in various official events. However, we often observe that many speakers simply read the script without making eye contact, thereby lowering audience engagement. Practice with proper tools could benefit a speaker considerably. In this work, we iteratively designed *ScriptFree*, an adaptive speech practice environment where off-the-shelf automatic speech recognition (ASR) is leveraged to measure a speaker's preparation level, and accordingly, the script is adaptively compressed to reduce the speaker's visual reliance toward script mastery. The user study results confirmed that *ScriptFree* helped the participants to successfully improve their speech over multiple practice iterations. The results have significant design implications for building adaptive speech practice systems.

Author Keywords

Public speaking; Speech preparation; Memory building

CCS Concepts

•Human-centered computing → **Human computer interaction (HCI);** User studies;

Introduction

Speaking from a manuscript is a popular form of speech delivery observed across a wide range of formal events, including ceremonial speeches, press briefings, company announcements, legal statements, religious services, and funding pitches. In such manuscript speech, the precise wording is important and should be spoken accurately according to a prepared script. In addition to such formal events, many people prepare and use manuscripts for their speeches to reduce public speaking anxiety or communication apprehension, or to compensate for the perceived difficulty of speaking in a non-native language [4].

Speakers who deliver manuscript speeches often simply read the written script directly, without any practice, because it takes less time and effort to read than to recite from memory. However, while speakers are looking at their scripts they cannot make eye contact with the audience, which reduces audience engagement with both speaker and speech [12, Chapter 14]. Furthermore, written scripts rarely follow the patterns of natural speech, and reading them verbatim only adds to a sense of artificiality for audiences [2, Chapter 14]. Thus, it is important for speakers to practice verbalizing naturally and confidently so they can recite sentences from memory with minimal script reliance.

Researchers have attempted to design interactive tools for supporting speech preparation and practice in recent years. Trinh et al. [10] proposed an integrated rehearsal environment that augments existing slideware with extended authoring, cued-recall testing, and spoken rehearsal. Various speech analytics and feedback mechanisms, especially nonverbal ones, have also been designed to help speakers deliver a better speech, for example by analyzing speech rate and eye contact [3], delivering planned pacing feedback [7, 9], and monitoring pauses and filler sounds [8] and body energy and openness [1]. Checking whether speakers have correctly followed their script, however, still requires primitive methods such as recording their speech and asking other people to check their current status [6]. Recent advances in off-the-shelf automatic speech recognition (ASR) technologies have the potential to augment or replace human-coached rehearsals by automatically measuring preparation levels and adaptively reducing visual reliance on scripts. However, there remains a lack of interactive tools for supporting script mastery.

In this paper, we present the iterative design of *ScriptFree*, an adaptive speech practice system for reducing visual reliance on written scripts, by leveraging off-the-shelf ASR. The system allows speakers to practice the prepared script by automatically comparing transcripts recognized by ASR with adaptive script compression according to the user's level of performance. As a result, speakers can gradually lower their visual reliance on the script. The key contributions to our work can be summarized as follows: 1) the iterative design and implementation of *ScriptFree*; 2) an investigation into the use of ASR feedback for speech preparation; and 3) the results of a user study leading to practical design implications for adaptive support of speech practice.

Design: To develop an initial understanding of how speakers interact with ASR-based feedback, we built an exploratory prototype that quantifies the speech preparation process. We selected several metrics that mirror the kinds of feedback given by human coaches. The prototype automatically quantifies keyword coverage (the ratio of keywords spoken among pre-selected keywords), peeking frequency (how often a speaker peeks at the script by touching the screen), and number of extended pauses (silent periods longer than two seconds). These are visualized as a bar graph, with spoken/unspoken keywords highlighted after every trial.

Participants: Seven participants (4 males and 3 females; ages 21-28 years with a mean of 25) were recruited from an online community. Participation was encouraged for those who are comfortable expressing ideas and feedback in English. Each participant was compensated with $15 cash.

Methodology: Participants began with a brief introductory session about the prototype and nine sentences to deliver. They selected keywords that they considered important and which could serve as a visual cue for remembering the content. Then, participants practiced their speech using the prototype for 20 minutes. Subsequently, we conducted interviews about the practice experiences. Each experiment took approximately one hour, and sessions were video recorded for further analysis. We analyzed the recorded videos and interview data for findings that could be reflected in the next iteration of the prototype.

Lessons Learned: The analysis taught us the followings.

- **Progressive increase in challenge level**: Participants had difficulties practicing more than one sentence at a time in the early stages. Participants were observed sub-vocalizing the script several times before making an initial attempt to verbalize it without the script. This led to the idea that speakers should be encouraged to read the full script in the first instance before gradually internalizing the content over subsequent iterations.
- **Progressive reduction in peekable script**: Participants used the script-peeking action for constant reassurance, and appeared reluctant to recall text from memory as long as there was a chance that they might fail. The result was continued visual reliance on the full script. The lesson is that reducing the script itself might provide a more control over the user's progression than attempts to reduce reading time though a peek-counting penalty metric.
- **Interactive and actionable feedback**: Participants had difficulties in using the feedback measures to improve their practice. For instance, given the number of peeks, participants did not know what to do to reduce this measure on the next round. To address this issue, revised feedback should actively shape the speaker's actions in ways that rely less on interpretation and discretionary corrective action.
- **Accommodation of ASR fallibility**: Participants expressed concerns about inaccurate keyword coverage measures arising from ASR recognition errors rather than speech errors. Such errors are hard to avoid despite recent technological advances. Accordingly, sympathetic accommodation of possible ASR errors is required for a satisfying user experience, and this requires more lenient and error-tolerant approaches to transcript-script comparison.

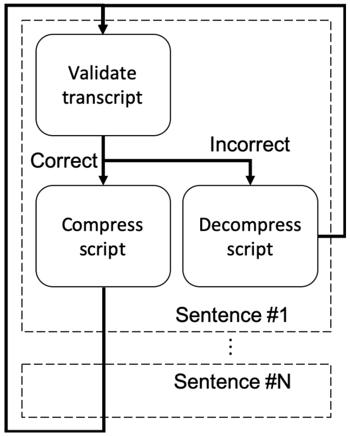


Figure 1: The overview of adaptive script compression.

Proof-of-concept Prototype Design

The lessons from the initial prototype guided us to design *ScriptFree*, a speech practice system for reducing visual reliance on scripts. The improved design allows users to practice sentence by sentence. As users succeed in verbalizing each sentence, the system adaptively reduces the visible portion of the sentence. By repeating this process, users eventually lower their visual reliance on the script. The system is built upon the following components: 1) transcript validation; and 2) adaptive adjustment of visual reliance on the script.

Transcript Validation

To measure how well a user verbalizes script, the system transcribes their speech in real time and evaluates the similarity of the recognized transcript to the prepared script.

$$presence(w_s, w_t) = \left\{\begin{matrix} 1 \text{ if } D_{edit}(w_s, w_t) \leq TH_{w_s} \\ 0 \text{ if } D_{edit}(w_s, w_t) > TH_{w_s} \end{matrix}\right\}$$

$$similarity(s_s, s_t) = \frac{\sum_{s=1}^{N_s} \sum_{t=1}^{N_t} presence(w_s, w_t)}{N_S}$$

Mitigating errors in transcripts is a challenge, as observed in a preliminary study. Our system leverages an edit distance metric, widely used for spelling correction and DNA sequence matching, that quantifies the similarity between two strings by counting the number of operations required to transform one string into the other [5]. The system used word edit distance to check whether a word in a transcript appears in a script. If the edit distance between a word in the transcript (w_t) and a word in the script (w_s) is smaller than or equal to the word threshold (TH_{w_s}), the system regards the word as present in the transcript. For example, if "answering" is recognized as "entering"' due to recognition error, the system is able to correct the word's presence in the transcript with the edit distance calculated smaller than or equal to the threshold. The sentence similarity is calculated as the ratio of correctly spoken words normalized by the number of words in the script. If the transcript-script similarity is higher than the sentence threshold (TH_s), the transcript is determined to be correctly spoken. The sentence threshold is determined based on the transcript from the first reading trial with the uncompressed script. Consequently, every sentence maintains its own sentence threshold values depending on the user's performances.

For the implementation, we used *Levenshtein* edit distance, which is one of the commonly used distances for a string-to-string correction problem [11], and set the word threshold for each word as one-third of the length of the target word. Lastly, a slack value of 0.8 was multiplied to the initial sentence similarity value to decide sentence threshold so that the system had a buffer against occasional ASR errors.

Adaptive Script Compression

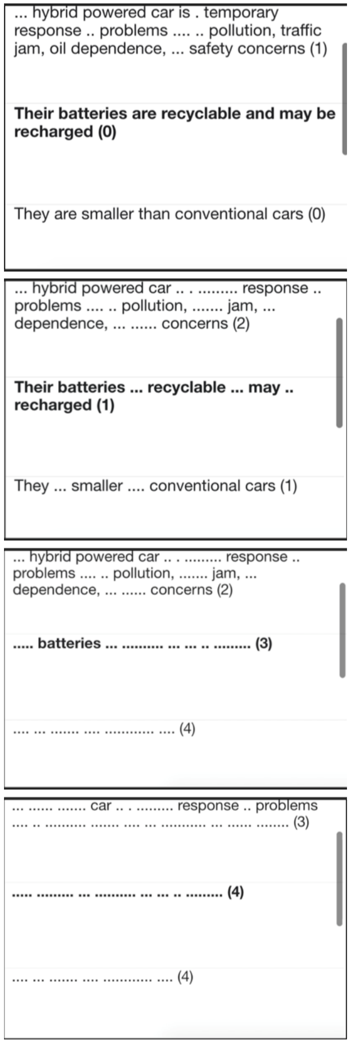
Using transcript validation, the system controls the order of sentences. We used sentences as the unit of practice because: 1) the initial prototype showed that users had difficulty in practicing multiple sentences at a time from the beginning; and 2) a sentence is a natural unit of human speech. The user begins by reading the given sentences one by one. Then, the user progresses to the next sentence if the transcript is spoken correctly according to transcript validation. After the first iteration, sentence thresholds are all configured to be the similarity value measured from the actual recording.

The system adaptively adjusts the compression level of sentences according to the user's speaking performance, as determined by transcript validation. The more compressed a script, the fewer the words visible (see Figure 2). If the user succeeds in producing a correct transcript for the given sentence, the compression level is incremented. Likewise, if an incorrect transcript is produced, the compression level is decremented. In case of a failure, the user should try the sentence again until a passable transcript is generated. In the end, users are expected to reach the highest compression level, which implies that they have achieved freedom from visual reliance on the script. The overall process is described in Figure 1. Unlike the initial prototype, this mechanism provides more interactive and directive feedback without the need to interpret statistics after every utterance.

Evaluation

Implementation

We developed the final prototype based on JavaScript, using IBM Watson for ASR. If there is a pause of more than two seconds, the prototype transcribes the recording and validates the sentence. The system ignores a transcript of less than four words because short transcripts are frequently generated because of environmental noises and unintentional sounds (e.g., dragging a chair or coughing). It also filters out words that have less than three phonemes



... hybrid powered car is . temporary response .. problems .... .. pollution, traffic jam, oil dependence, ... safety concerns (1)

Their batteries are recyclable and may be recharged (0)

They are smaller than conventional cars (0)

... hybrid powered car .. . . response .. problems .... .. pollution, ... jam, ... dependence, ... concerns (2)

Their batteries ... recyclable ... may .. recharged (1)

They ... smaller .... conventional cars (1)

... hybrid powered car .. . . response .. problems .... .. pollution, ... jam, ... dependence, ... concerns (2)

..... batteries ... . (3)

.... (4)

... . car .. . . response .. problems .... (3)

.... (4)

.... (4)

Figure 2: Screenshots of the final prototype (the more compressed sentences are toward the bottom)

for the following reasons: 1) short words are often the result of recognition errors; and 2) relatively long words are more likely to be content words, and they should not be overwhelmed by function words (e.g., articles and prepositions). To achieve the progressive reduction of the script, we adopted the text compression technique for slide notes proposed in previous work [10]. This method resulted in five levels of compressed sentences, including original text and no text. Compressed words are represented as dots that have the same length as the original word, in order to maintain consistency of layout.

Methodology

We recruited seven participants from a university website (two males and five females, ages 24-32 years with a mean of 27). All participants had several public presentation experiences. We recruited people who can speak English fluently as determined by accurate transcription by IBM Watson ASR when reading test sentences.

We briefed participants that they were to prepare for a manuscript speech. During the experiment, participants were provided with sentences from a public website representing a sales pitch for a hybrid vehicle. The script consisted of six sentences, and they were different in content and length (mean: 13.50 words; sd: 5.54). We first explained how the prototype worked and then allowed participants several trial runs with sample sentences.

After being given two minutes to grasp the content of the script, participants were allowed to practice for 15 minutes using the prototype. The participant was alone in an isolated room to minimize distractions. The user's screen and voice were recorded. Overall, each session took about one hour and finished with a semi-structured interview. Interview questions belonged to three main categories: 1) general experiences during the experiment; 2) the adaptivity of the interface (e.g., compression level, differences in difficulty by sentence); and 3) suggestions for improvement.

Results

Participants successfully improved over several iterations, as shown in Figure 3. The observed compression levels ranged from 0 (full script) to 4 (no script). The initial task from level 0 to level 1 was performed with the full script. Participants performed 7.7 iterations on average (std: 1.4), and the mean compression level reached was 3.3 (std: 0.9). On a scale from 0 to 4, such a mean compression level of 3.33 represents a high degree of content mastery.

A majority of participants agreed that the system was useful for practice. P2 commented, *"Following this method repeatedly, I thought I could memorize content without spending much time and effort."* P3 also responded, *"For sure, I felt like it helped me memorize things better. I was just filling in gaps....Whenever I said something incorrectly, that mistake actually made me remember those particular words better because this method made me repeat the sentence again."*

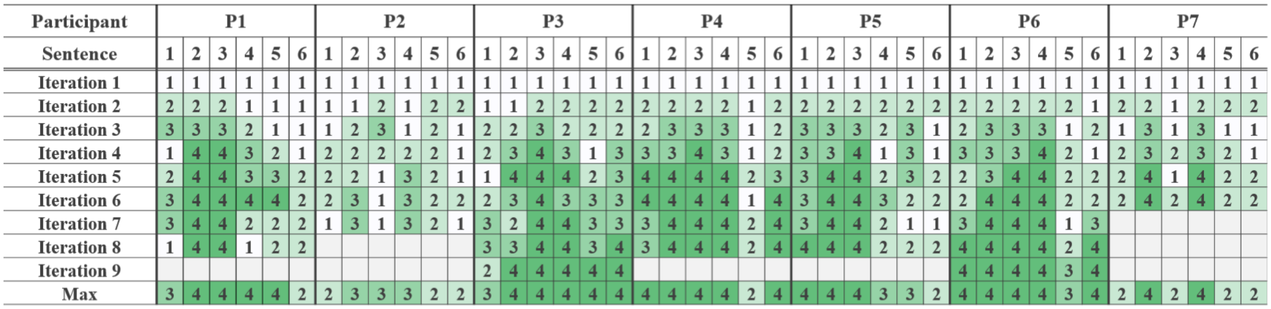
Even though most recognitions were unproblematic, some participants reported instances of ASR errors. P1 stated, *"I reached the third level but got back to the original sentences due to the recognition problem. It made me go back to the first level even when I thought I speak correctly."*

Design Implications

We now describe design implications for adaptive speech practice systems based on analysis of study data.

The Need for Dynamic Script Compression

Although the participants performed the given task successfully, we found room for improvement in terms of script compression as the perceived difficulty increment between compression levels varied from sentence to sentence. For



Participant	P1						P2						P3						P4					
Sentence	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Iteration 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Iteration 2	2	2	2	1	1	1	1	1	2	1	2	2	1	1	2	2	2	2	2	2	2	2	1	2
Iteration 3	3	3	3	2	1	1	1	2	3	1	2	1	2	2	3	2	2	2	2	3	3	3	1	2
Iteration 4	1	4	4	3	2	1	2	2	2	2	2	1	2	3	4	3	1	3	3	3	4	3	1	2
Iteration 5	2	4	4	3	3	2	2	2	1	3	2	1	1	4	4	4	2	3	4	4	4	4	2	3
Iteration 6	3	4	4	4	4	2	2	3	1	3	2	2	2	3	4	3	3	3	4	4	4	4	1	4
Iteration 7	3	4	4	2	2	2	1	3	1	3	2	1	3	2	4	4	3	3	3	4	4	4	2	4
Iteration 8	1	4	4	1	2	2							3	3	4	4	3	4	3	4	4	4	2	4
Iteration 9													2	4	4	4	4	4						
Max	3	4	4	4	4	2	2	3	3	3	2	2	3	4	4	4	4	4	4	4	4	4	2	4

Participant	P5						P6						P7					
Sentence	1	2	3	4	5	6	1	2	3	4	5	6	1	2	3	4	5	6
Iteration 1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Iteration 2	2	2	2	2	2	2	2	2	2	2	2	1	2	2	1	2	2	2
Iteration 3	3	3	3	2	3	1	2	3	3	3	1	2	1	3	1	3	1	1
Iteration 4	3	3	4	1	3	1	3	3	3	4	2	1	2	3	2	3	2	1
Iteration 5	3	4	4	2	3	2	2	3	4	4	2	2	2	4	1	4	2	2
Iteration 6	3	4	4	3	2	2	2	4	4	4	2	2	2	4	2	4	2	2
Iteration 7	3	4	4	2	1	1	3	4	4	4	1	3						
Iteration 8	4	4	4	2	2	2	4	4	4	4	2	4						
Iteration 9							4	4	4	4	3	4						
Max	4	4	4	3	3	2	4	4	4	4	3	4	2	4	2	4	2	2

Figure 3: Transition of compression levels of script for each iteration (darker color indicates more compressed script)

example, P3 said, *"For long sentences, it was harder to make the sentence right because more words disappeared in a single-level transition (for long sentences)."* This variation of difficulty between sentences was attributed to the removal of words at a constant ratio for all sentences, and thus, the actual number of removed words would differ depending on the sentence length.

To tackle these issues, more dynamic approaches to script compression should be considered. One option is that on an incorrect utterance, the system should initially retain the same compression level but reveal a different subset of words. As P1 suggested, *"When transitioning from the second level to the third level, very important words [which served as cues] disappeared at once. It would be easier if I repeated the same level with different words again."* In addition, it would also be helpful to apply compression based on the user's performance for each word. P5 noted, *"Especially, rather than removing (a fixed set of) words by level, it would be better if it removes what I repeatedly speak correctly."* P6 also mentioned, *"The level of compression was fine, but I expected that the correctly spoken words would disappear and the incorrect words would remain."*

The Need for Sentence Linking Practice

Several users mentioned that the system needs to help train the connection between sentences as well as the sentences themselves. P1 commented, *"Because I was focusing on disappearing words too much, it hindered me from obtaining the overall flow of the content."* Even though scripts repeatedly present the same sentence order, users still had difficulties learning the overall flow.

The system should therefore be extended to allow and encourage users to speak multiple sentences continuously. One alternative was suggested by P4, who stated, *"I usually skip the wrongly spoken sentences....I will eventually practice the wrongly spoken sentence again when I get back (to the sentence)."* P5 also gave similar feedback: *"I practice one sentence first, and then go to the second. Again, I start from the first and repeat it in a cumulative way."*

The Need for Personalized Control of Practice Flow

Further fine-tuning of practice flow based on practice performance was suggested by several participants. P2 said, *"For the sentence I repeatedly speak incorrectly, I hope the system shows it more frequently."* This implies that the system would benefit from adaptive scheduling of sentences such that when a user repeatedly fails on a particular sentence or transition, it appears again sooner and more frequently in the future. The design of such scheduling could draw inspiration from the many approaches to cued-recall, spaced-repetition learning embodied by flashcard applications.

Conclusions & Future Work

We have introduced *ScriptFree*, a speech practice system for adaptively reducing visual reliance on script. It leverages ASR and algorithms for transcript validation and adaptive script compression. The system controls the flow of practice sentences by compressing the script based on a speaker's performance. For future work, we plan to extend the system by incorporating additional aspects of speech performance such as speech rate, volume dynamics, and gesture level as well as complex script compression techniques. Diverse speech practice scenarios (e.g., a longer script and non-native language) will be also considered.

Acknowledgments

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. R7124-16-0004, Development of Intelligent Interaction Technology Based on Context Awareness and Human Intention Understanding).

REFERENCES

[1] Ionut Damian, Chiew Seng (Sean) Tan, Tobias Baur, Johannes Schöning, Kris Luyten, and Elisabeth André. 2015. Augmenting Social Interactions: Realtime Behavioural Feedback using Social Signal Processing Techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '15)*. 565–574. DOI: http://dx.doi.org/10.1145/2702123.2702314

[2] Judith N. Martin Jess K. Alberts, Thomas K. Nakayama. 2009. *Human Communication in Society, Second Edition*. Peachpit Press.

[3] Kazutaka Kurihara, M Goto, and J Ogata. 2007. Presentation sensei: a presentation training system using speech and image processing. In *Proceedings of the 9th international conference on Multimodal interfaces (ICMI '07)*. 358–365. DOI: http://dx.doi.org/10.1145/1322192.1322256

[4] Xiang Li and Jun Rekimoto. 2014. SmartVoice: a presentation support system for overcoming the language barrier. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. 1563–1570. DOI: http://dx.doi.org/10.1145/2556288.2557161

[5] Gonzalo Navarro. 2001. A guided tour to approximate string matching. *ACM computing surveys (CSUR)* 33, 1 (2001), 31–88.

[6] Ryo Okamoto and Akihiro Kashihara. 2011. Back-review support method for presentation rehearsal support system. In *Knowlege-Based and Intelligent Information and Engineering Systems*. Springer, 165–175.

[7] Bahador Saket, Sijie Yang, Hong Tan, Koji Yatani, and Darren Edge. 2014. TalkZones: Section-based Time Support for Presentations. In *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services (MobileHCI '14)*. 263–272. DOI: http://dx.doi.org/10.1145/2628363.2628399

[8] Jan Schneider, Dirk Börner, Peter van Rosmalen, and Marcus Specht. 2015. Presentation Trainer, your Public Speaking Multimodal Coach. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction (ICMI '15)*. 539–546. DOI: http://dx.doi.org/10.1145/2818346.2830603

[9] Diane Tam, Karon E MacLean, Joanna McGrenere, and Katherine J Kuchenbecker. 2013. The design and field observation of a haptic notification system for timing awareness during oral presentations. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '13)*. ACM, 1689–1698.

[10] Ha Trinh, Koji Yatani, and Darren Edge. 2014. PitchPerfect: Integrated Rehearsal Environment for Structured Presentation Preparation. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '14)*. 1571–1580. DOI: http://dx.doi.org/10.1145/2556288.2557286

[11] Robert A Wagner and Michael J Fischer. 1974. The string-to-string correction problem. *Journal of the ACM (JACM)* 21, 1 (1974), 168–173.

[12] Jason S. Wrench. 2012. *Public Speaking: Practice and Ethics*.