*a*Seamless Scripting and Recording of Audio Narratives

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

Leave Authors Anonymous
for Submission
City, Country
e-mail address

ABSTRACT

ACM Classification Keywords

H.5.2. Information Interfaces and Presentation (e.g. HCI): User Interfaces - Graphical user interfaces

Author Keywords

Audio recording, scripting, transcript-based editing

INTRODUCTION

Audio recordings are a common form of communication used in voice-overs, podcasts, audio books and e-learning. Closest to everyday conversation, audio recording is a medium with relatively low barriers to entry. So, it is used by many laymen who are not professional producers or writers to distribute information. A common workflow for creating such audio recordings involves three main steps: writing a script, recording audio and editing audio. To create a compelling audio recording, users go through several iterations of these steps.

Consider the case of recording a voice-over for a video. The narrator does an initial recording based on a prepared script. Afterwards, while placing it on top of the video, the producer wants to make some changes to parts of the narration e.g. to control the timing of a specific word, or to match changes made in the shots after the audio was recorded. While some of these edits can be done from the existing recording using an audio editing software, others require editing the script and re-recording the altered parts. Similarly, consider an audio recording of an online lecture. After the initial publication, the lecturer may want to re-record or add parts e.g. to keep examples up to date, or to address common questions that came up from viewers. These are but a few of many scenarios where producers iterate back and forth between script writing, audio recording and audio editing.

Most existing audio editing systems provide functionalities necessary to support the latter two steps: recording and editing audio. However, the first step, writing or editing the script, is usually overlooked or treated as a completely separate task. This is the case even when scripts play a key role in recording and editing speech.

In this paper, we present an interface that supports and links all three steps of the aforementioned workflow. Our system addresses challenges that span the process of creating audio narratives, including (1) iterating back and forth between script and audio recordings, and (2) combining multiple audio recordings into a single final track. Our interface is inspired from familiar document editors and text merge tools, which are easy to learn.

RELATED WORK

Adobe Story [2], FinalDraft [6] and Celtx [5] are examples of software applications dedicated to script writing. They support collaboration, automatic formatting, navigation and planning for future production, but they treat the script as a text document that is essentially separate from the recordings. In fact, in our preliminary interview of lay and professional audio producers, we found that many of them use general-purpose document editors like Google Docs [8] or Microsoft Word [11] to prepare their scripts.

At the recording and editing stage, Adobe Audition [1], Avid ProTools [4], GarageBand [7] and Audacity [3] are among the most popular digital audio workstations (DAWs). These tools allow users to edit audio by manipulating waveforms in a multi-track timeline interface. They also provide a wide variety of low-level signal processing functions. However, since they are designed to serve as general-purpose audio production systems, they include many features that are not directly relevant for creating audio narratives whose main content is speech. Hindenburg Systems [9] develops tools that are specifically targeted for audio narratives. Still, they are primarily concerned only with the audio and they do not deal with the script directly.

Recently, several researchers have explored using audio transcripts to support text-based navigation and editing of audio. Whittaker and Amento [16] demonstrate that users prefer editing voicemail through its transcript instead of its waveform. Inspired by similar intuition, Casares et al. [13] and Berthouzoz et al. [12] enable video navigation and editing through time-aligned transcripts. Rubin et al. [15] extend this approach to audio narratives and propagate edits in the transcript text to the corresponding speech track. These systems all focus on editing the audio on the assumption that all the recording is done beforehand. *Narration Coach* developed by Rubin et al. supports an iterative narration recording process, but instead it assumes that the user has a fixed input script. Its focus, providing capture-time feedback to improve speech performance, is also very different from ours. Our work also

takes advantage of text-based navigation and editing, but unlike these systems, we support an dynamic workflow where both the audio recordings and the underlying script can be continuously updated.

KEY IDEAS

To learn about current practices and challenges for recording audio narratives, we interviewed ten professional lecturers and two video producers who regularly created audio recordings as part of online lectures that were published on platforms, including YouTube, Udacity, EdX and MITx. Following are several key insights we gained from the pilot interviews.

Scripts play a major role during recording. All of the lecturers prepared written materials about what they were going to speak before they started recording. The format and level-of-details of these scripts varied. For instance, one lecturer used his lecture slides containing images and a list of bullet points as his script. Another lecturer typed a thorough word-for-word transcription of what he was going to say in a text document. Another person used a handwritten outline. In all cases, while they were recording, they kept the scripts within their view and depended on them to construct their narrative.

Script planning, recording, and editing is an iterative process. Even when a thorough word-for-word script is prepared beforehand, often the recording does not follow the script exactly. The speaker may remember and add more details while recording, or find a more natural way of saying a written sentence. In some cases, major changes are made to the script after some recording has happened. The iteration may happen over a longer period of time. One lecturer noted that he periodically revisited and re-recorded parts of the lecture to include up-to-date examples.

The final track is created by mixing and matching multiple recordings. Most users recorded multiple takes, and then edited them using an audio editing software to produce the final recording. Many of them noted that aligning the multiple takes, finding the best take of a given part, and seamlessly merging the cuts were the most time consuming and tedious tasks.

USER INTERFACE

Motivated by these insights, we developed the VoiceScript interface to support iterative script writing and audio recording/editing. As shown in Figure **??**, our interface consists of two types of documents: a *Master Script* document (left) shows the current status of the final track (i.e. what has been recorded) on top of the script (i.e. what was planned to be recorded), while a *Transcript* document (right) displays the verbatim transcript of individual audio takes. When the user records a new take, the system analyzes its is analyzed and aligned with the current master script.

As the user records new takes, VoiceScript timealigns the recording with the *Master Script*.

We first describe our interface through an example workflow, and then present the main algorithms used in our system in the 5 section.

Our system displays two types of documents: the master-script

and transcripts. The *master-script*, representing the current audio narrative, is the document that users work on. Users start by writing on the master-script what they plan to record. At this stage, the master-script is like an ordinary word document or script. Once the users start recording, the text corresponding to each take appears in a separate *transcript* document tab. If the *aligned-view* is turned on, the master-script and the currently selected transcript is segmented.

The user can click on a button to use that segment of the audio. Now that part appears in text color to indicate. If the user has multiple takes, they can go back and forth. If they go to the All Tab, they can see multiple takes of each sentence. Also edit the master script, just as they would edit a text document. If the user edits a recorded portion of the script, the relevant section is makred as dirty to indicate that it should be re-recorded.

ALGORITHMIC METHODS

Transcribing the audio recording

We use the IBM Speech to Text Service [10] to obtain a verbatim transcript of each audio take in real time. The service produces a time stamp for each word indicating its start and end time. It also segments the transcript into *utterances* where each utterance is separated by a longer silent gap in the speech (longer than 500 ms). While automatic speech recognition is imperfect, we have found that the results were accurate enough for the purpose of alignment (next section) and for understanding the transcript.

Aligning the transcript to the master script

Once we have a verbatim transcript of an audio take, we compute the global word-to-word alignment between the transcript and the master script using the Needleman-Wunsch (NW) algorithm [14]. NW allows for insertions and deletions, which accounts for differences in the two documents.

Co-segmenting the transcript and the master script

RESULTS

INFORMAL USER EVALUATION

CONCLUSIONS

ACKNOWLEDGMENTS

REFERENCES

1. 2016a. Adobe Audition. `http://www.adobe.com/products/audition.html`. (April 2016). Accessed: 2016-04-02.

2. 2016b. Adobe Story. `https://story.adobe.com/en-us/`. (Apr 2016). Accessed: 2016-04-02.

3. 2016. Audacity. `http://www.audacityteam.org/`. (April 2016). Accessed: 2016-04-02.

4. 2016. Avid Protools. `http://www.avid.com/en/pro-tools`. (April 2016). Accessed: 2016-04-02.

5. 2016. Celtx. `https://www.celtx.com/index.html`. (April 2016). Accessed: 2016-04-02.

6. 2016. Final Draft. `https://www.finaldraft.com/`. (April 2016). Accessed: 2016-04-02.

7. 2016. GarageBand. `http://www.apple.com/mac/garageband/`. (April 2016). Accessed: 2016-04-02.

8. 2016. Google Docs. `https://www.google.com/docs/about/`. (April 2016). Accessed: 2016-04-02.

9. 2016. Hindenburg Journalist Pro. `http://hindenburg.com/products/hindenburg-journalist-pro`. (April 2016). Accessed: 2016-04-02.

10. 2016. IBM Speech to Text Service. `https://www.ibm.com/smarterplanet/us/en/ibmwatson/developercloud/doc/speech-to-text/`. (April 2016). Accessed: 2016-04-02.

11. 2016. Micorsoft Word. `https://products.office.com/en-us/word`. (April 2016). Accessed: 2016-04-02.

12. Floraine Berthouzoz, Wilmot Li, and Maneesh Agrawala. 2012. Tools for placing cuts and transitions in interview video. *ACM Transactions on Graphics (TOG)* 31, 4 (2012), 67.

13. Juan Casares, A Chris Long, Brad A Myers, Rishi Bhatnagar, Scott M Stevens, Laura Dabbish, Dan Yocum, and Albert Corbett. 2002. Simplifying video editing using metadata. In *Proceedings of the 4th conference on Designing interactive systems: processes, practices, methods, and techniques*. ACM, 157–166.

14. Saul B Needleman and Christian D Wunsch. 1970. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology* 48, 3 (1970), 443–453.

15. Steve Rubin, Floraine Berthouzoz, Gautham J Mysore, Wilmot Li, and Maneesh Agrawala. 2013. Content-based tools for editing audio stories. In *Proceedings of the 26th annual ACM symposium on User interface software and technology*. ACM, 113–122.

16. Steve Whittaker and Brian Amento. 2004. Semantic speech editing. In *Proceedings of the SIGCHI conference on Human factors in computing systems*. ACM, 527–534.