Authoring and Transcription Tools for Speech-Based Hypermedia Systems

Barry Arons
MIT Media Laboratory
20 Ames Street, E15-353
Cambridge MA, 02139

Phone: +1 617-253-2245
E-mail: barons@media-lab.mit.edu

Abstract

Authoring is usually one of the most difficult parts in the design and implementation of hypertext and hypermedia systems. This problem is exacerbated if the data to be presented by the system is speech, rather than text or graphics, because of the slow and serial nature of speech. This paper provides an overview of speech-only hypermedia, discusses the difficulties associated with authoring databases for such a system, and explores a variety of techniques to assist in the authoring process.

Speech-Only Hypermedia

Since the introduction of Hypercard for the Macintosh, the ideas behind hypertext systems have become commonplace. The addition of graphics, audio, still images, or video to such systems is helping to create a wealth of new hypermedia applications, but few of these systems take advantage of voice input or output. To create an end-user application, the raw source material must be assembled and structured as part of "authoring" process. For example, the authoring of a videodisc-based hypermedia system involves selecting appropriate video segments, scripting branch points, and creating graphics [7]. These systems generally use mouse interfaces to traverse through screen-based environments.

The authoring ideas discussed in this paper were developed for an experimental "hyperspeech" system that explores ideas of creating and navigating in a *speech-only* hypermedia framework (see [1] for a thorough discussion of the project). The system uses speech recognition to maneuver in a database of digitally recorded speech segments; synthetic speech is used for control information and user feedback. In this research prototype, recorded audio interviews on "the future of the human interface" were segmented by topic, with hypertext-style links added to connect logically related comments and ideas. The user speaks commands to hear supporting or opposing views, comments from a different speaker, or control the state of the system.

This project investigates techniques for presenting "speech as data," allowing a user to navigate by voice through a database of recorded speech *without* any visual cues. The ideas being developed can be applied to create a generalized form of interaction with unstructured speech data. Applications for such a technology include the use of recorded speech, rather than text, as a brainstorming tool or personal memory aid. A hyperspeech system would allow a user to create, organize, sort, and filter "audio notes" under circumstances where a traditional graphical interface would not be practical (e.g., while driving) or appropriate (e.g., for someone who is visually impaired). Speech interfaces are particularly attractive for handheld computers without keyboards or large displays.

Problems of Speech-Based Authoring

While the authoring of traditional hypermedia and multimedia databases is time consuming, the graphical tools used by the hypermedia author are usually similar to the interface presented to the user. Using a two-dimensional display space permits many objects (text windows, graphic images, video) to be displayed and viewed *simultaneously.* Browsing such a display is easy since it relies "on the extremely highly developed visuospatial processing of the human visual system" [3].

Speech and audio, however, exist only as a time varying signal—the auditory system cannot browse through a set of recordings the way the eye can scan a display. Speech interfaces *must* present information *sequentially* while visual interfaces can present information *simultaneously* [5, 10]. These factors lead to significantly different design issues when using speech [15], as opposed to text, video, or graphics. Recorded speech cannot be manipulated, viewed, or organized on a display in the same manner as text or video images. Schematic *representations* of speech signals (e.g., waveform, energy, or magnitude displays) can be viewed in parallel and managed graphically, but the speech signals themselves cannot be heard simultaneously [2].

Computer-Augmented Transcription

In developing this system, recorded interviews were transcribed to text, then manually segmented into logically related views. Starting and stopping points in the sound files that corresponded to the text selections were then found, and related segments were linked to create a highly interconnected hyperspeech database. This entire databased authoring process was very time consuming and painstaking.

One solution to managing voice recordings is to use traditional text (or hypertext) tools to manipulate transcriptions. Unfortunately, the transcription process is tedious, and the transcripts do not capture the prosody, timing, emphasis, or enthusiasm of speech that is important in a hyperspeech system. This section outlines ways that an audio-equipped workstation can help bridge this gap in the hypermedia authoring process.

The technology for the transcription of recorded interviews or dictation is steeped in tradition. A transcriptionist controls an analog tape machine via a foot pedal while entering text into a word processor. Modern transcribing stations have "advanced" features that can speed up or slow down the playback of recorded speech, can display the current location within the tape, and have high-speed search.

The workstation can be programmed to provide all the standard features of stand-alone transcription stations, but can additionally integrate digital signal processing, a high-resolution graphics display, and the ability to directly link text to audio data files. Some of these capabilities are available in expensive dedicated dictation systems, but a better solution is to integrate them into general purpose personal computers and engineering workstations.

Scanning Techniques

Increasing the playback speed of an analog audio tape by more than 20% decreases intelligibility by significantly shifting the pitch upward. Digital signal processing in the workstation can provide a greater range of speed changes *without* changing the pitch [4, 8]. A reasonable increase in speed can be achieved by simply removing periods of silence, or by combining silence removal with accelerated playback. A

dedicated DSP chip is not required, as such algorithms can run in real-time on the main processor of a contemporary workstation. This technology is also useful in presenting speech information to a user of a hyperspeech system, as it helps circumvent the time bottleneck usually associated with the presentation of speech information. Note that accelerating or compressing speech signals does not significantly degrade a listener's ability to comprehend information from the recordings [12].

A related technique is the ability to play intelligible speech while "rewinding" through a digital audio file. Analog tape systems provide little useful information about the signal when it is played completely backwards[1]. Digital systems allow windows of speech (perhaps 250–2000 ms) to be individually played forwards, with the segments themselves presented in reverse order[2]. While the general sense of the recording is reversed and jumbled, each segment is identifiable and intelligible. It now becomes practical to browse backwards through a recording in order to find a particular word or phrase. This method is particularly effective if the window boundaries are chosen to correspond to periods of silence. Note that this technique can also be combined with accelerated playback, allowing both backward and forward scanning at high speeds.

Correlating Text With Recordings

In addition to transcription, a hyperspeech system (and many other speech-based applications) needs to accurately correlate the text with the recorded sound data. Ideally this is done automatically without explicit action by the transcriptionist—as the text is typed, a rough correspondence is made between words and points in the recorded file. An accurate one-to-one mapping between the recording and the transcription is unlikely because of the typist's ability to listen far ahead of letters being typed at any moment [13].

Once a transcript is generated, fine-grained beginning and ending points must be determined for each speech segment. A graphical editor can assist in this process by displaying the text in parallel with a visual representation of the speech signal. This allows the hypermedia author to visually locate pauses between phrases for segments of speech in the hyperspeech database. Specialized text editors can be used for managing transcripts that have inherent structure or detailed descriptions of actions (such as data from psychological experiments that including notations for breathing, background noises, non-speech utterances, etc.) [11].

Authoring the Hyperspeech Database

In the developing of the hyperspeech database for this project, an intermediate approach was taken. Each participant was called by a telemarketing-style program that recorded the responses to a series of questions into separate speech files. Recordings were terminated using silence detection, without manual intervention. The recordings were then manually transcribed on a Sun SPARCstation using a conventional text editor while simultaneously controlling audio playback with a custom built foot pedal. A serial mouse was the foot pedal, with button events controlling the playback of the digital recordings.

After manually analyzing printed transcripts to find interesting speech segments, a separate segmentation utility was used to determine the corresponding begin/end points in the sound file. This utility played small fragments of the recording (125 ms) allowing the database author to determine begin/end points

[1] This is analogous to taking the sentence "This is a test" and presenting it as "tset a is sihT."

[2] This method, for example, could result in a presentation of "test is a This."

within the sound files. Keyboard-based commands analogous to fine, medium, and coarse grained cursor motions in a popular text editor were used to move through the sound file and determine the proper segmentation points. In the data collected for this project, most of the sound segments began and ended on pauses associated with natural phrase boundaries. However, a small number of nodes in the database started or stopped within phrases, and fine-grained sound editing selectivity was needed.

Automated Approaches to Authoring

Unfortunately, fully automatic speaker-independent speech-to-text transcription of spontaneous speech is not practical in the near future. However, there are a variety of techniques that can be employed to completely automate the hyperspeech authoring process.

If an accurate transcript is available, it is possible to automatically correlate the text with syllabic units detected in the recording [6, 9][3]. For a hyperspeech database, this type of tool would allow the hypermedia author to segment the transcripts in a text-based editor, and then create the audio file correspondences as an automated post-process. Even if the processing is not completely accurate, it would provide rough begin/end points that could be tuned manually.

The telemarketing-style program that collected the interview database asked a series of questions that served as the foundation for the organization of the hyperspeech database. In this prototype application, the questions were very broad, and much manual work was required to segment and link the nodes in the database. However, if the process that gathers the speech data asks very specific questions, it is possible to automatically segment and organize recorded messages by semantic content [14]. If the questions are properly structured (and the interviewees are cooperative), the bulk of the nodes in the hyperspeech database can be automatically generated. This technique is particularly powerful for hyperspeech authoring, as it not only creates the content of the database, but can link the nodes as well. This style of automatic tool is also useful for managing less structured data.

A final, and very appealing, technique is to use speech to control the segmentation and linking tasks completely in the speech domain. The hyperspeech system could have additional commands in an special mode that permit voice control of the authoring process. This style of authoring further addresses many of the fundamental research issues of the hyperspeech project. There are many speech and user interface problems to be explored in order to create such a tool, but the experience gained in developing such as system will provide voice interaction techniques that will be useful in a wide range of applications.

Conclusions

During the authoring process, it became painfully clear that continued development of such a system would require significantly better authoring tools and techniques. This paper summarizes a variety of tools related to transcription and authoring that are being investigated at the Media Lab.

Due to the lack of appropriate tools, the most practical way to manually author a hyperspeech database today is though the use of text transcriptions. Through continued work in the area of managing and navigating within audio-only databases, it may someday be practical to author such databases through a voice interface, or completely automate this part of the authoring task.

[3]Related ideas for exploring the content of movies sound tracks are described in [16].

Acknowledgements

Chris Schmandt and Lisa Stifelman assisted in the editing of this paper. Mike Hawley and Walter Bender provided useful technical information and insight. This work was funded by Apple Computer and Sun Microsystems.

References

[1] B. Arons. Hyperspeech: Navigating in speech-only hypermedia. In *Hypertext '91*, pages 133–146. ACM, 1991.

[2] B. Arons. A review of the cocktail party effect. *Journal of the American Voice I/O Society*, 12:35–50, July 1992.

[3] J. Conklin. Hypertext: an introduction and survey. *IEEE Computer*, 20(9):17–41, September 1987.

[4] W. Foulke and T. G. Sticht. Review of research on the intelligibility and comprehension of accelerated speech. *Psychological Bulletin*, 72:50–62, 1969.

[5] W. W. Gaver. Auditory icons: Using sound in computer interfaces. *Human-Computer Interaction*, 2:167–177, 1989.

[6] A. Hu. Automatic emphasis detection in fluent speech with transcription. Unpublished M.I.T. Bachelor's thesis, May 1987.

[7] J. S. Huntley and S. Alessi. Videodisc authoring tools: Evaluating products and a process. *Optical Information Systems*, pages 259–281, 1987.

[8] N. Maxemchuk. An experimental speech storage and editing facility. *Bell System Technical Journal*, 59(8):1383–1395, October 1980.

[9] P. Mermelstein. Automatic segmentation of speech into syllabic units. *Journal of the Acoustic Society of America*, 58(4):880–883, October 1975.

[10] M. J. Muller and J. E. Daniel. Toward a definition of voice documents. In *Proceedings of COIS '90*, 1990.

[11] K. M. Pitman. CREF: An editing facility for managing structured text. A. I. Memo 829, Massachusetts Institute of Technology, February 1985.

[12] A. Richaume, F. Steenkeste, P. Lecocq, and Y. Moschetto. Intelligibility and comprehension of French normal, accelerated, and compressed speech. In *IEEE Engineering in Medicine and Biology Society 10th Annual International Conference*, pages 1531–1532, 1988.

[13] T. A. Salthouse. The skill of typing. *Scientific American*, pages 128–135, February 1984.

[14] C. Schmandt and B. Arons. A conversational telephone messaging system. *IEEE Transactions on Consumer Electronics*, CE-30(3):xxi–xxiv, August 1984.

[15] C. Schmandt and B. Arons. Desktop audio. *Unix Review*, October 1989.

[16] T. G. Aguirre Smith and N. C. Pincever. Parsing movies in context. In *Proceedings of the Summer 1991 USENIX Conference*, pages 157–168. Usenix, 1991.