

Semantic Speech Editing

Steve Whittaker

Sheffield University
211 Portobello St, Sheffield, S1 4DP, UK.
s.whittaker@shef.ac.uk

Brian Amento

AT&T Labs-Research
180 Park Ave, Florham Park, NJ, 07932, USA.
brian@research.att.com

ABSTRACT

Editing speech data is currently time-consuming and error-prone. Speech editors rely on *acoustic* waveform representations, which force users to repeatedly sample the underlying speech to identify words and phrases to edit. Instead we developed a semantic editor that reduces the need for extensive sampling by providing access to *meaning*. The editor shows a time-aligned errorful transcript produced by applying automatic speech recognition (ASR) to the original speech. Users visually scan the words in the transcript to identify important phrases. They then edit the transcript directly using standard word processing ‘cut and paste’ operations, which extract the corresponding time-aligned speech. ASR errors mean that users must supplement what they read in the transcript by accessing the original speech. Even when there are transcript errors, however, the semantic representation still provides users with enough information to target what they edit and play, reducing the need for extensive sampling. A laboratory evaluation showed that semantic editing is more efficient than acoustic editing even when ASR is highly inaccurate.

Categories & Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems - audio input/output, evaluation/methodology; H.5.2 [Information Interfaces and Presentation]: User Interfaces - evaluation/methodology, prototyping, voice I/O; H.1.2 [Models and Principles]: User/Machine Systems - human factors, human information processing; H.5.5 [Information Interfaces and Presentation]: Sound and Music Computing - methodologies and techniques; H.4.3 [Information Systems Applications]: Communications Applications.

General Terms: Design, Experimentation, Human factors.

Keywords: Speech editing, acoustic representations, transcripts, speech browsing, speech retrieval, speech recognition.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI 2004, April 24–29, 2004, Vienna, Austria.

Copyright 2004 ACM 1-58113-702-8/04/0004...\$5.00.

INTRODUCTION

Speech is an important informational medium. Large amounts of valuable spoken information are exchanged in meetings, voicemail and public debates [1,7,11,13,16]. Speech also has general benefits over text, being both expressive and easy to produce [3,7]. Speech archives are now becoming increasingly prevalent, but until recently it was hard to exploit these archives because of a shortage of effective tools for accessing and manipulating speech data. Unlike text, speech is a serial medium that does not naturally support textual access techniques such as search, visual scanning or key word spotting [6,8].

Nevertheless, important progress has recently been made in developing new tools for accessing speech. Browsers have been developed that extract and represent different types of structural indices, allowing users to access speech by: speaker [4,7,9,10,17], emphasis [1,11], external events such as user note-taking behaviors [7,11,12,16], or accompanying visual events [6,9]. Signal processing techniques allow speech to be played back at several times its normal rate, retaining comprehensibility [1]. And content-based search can be applied to transcripts generated by automatic speech recognition (ASR) [2,8,9]. These transcripts are also highly effective as an interface to support browsing [10,17,19].

But these techniques have mainly been focused on browsing and search. Instead we address the problem of *speech editing*. The sequential nature of speech makes it laborious to access whole archives, placing extra value on editing. Effective editing can extract and summarize the main points of a speech record, allowing others to access key information without having to listen to all of it.

Most current speech editors rely on an *acoustic* representation. To edit speech, users listen to the underlying speech and then manipulate the acoustic representation. This is a laborious process that involves multiple editing actions, repeatedly sampling the speech to precisely identify the beginning and end of regions of interest.

Instead we developed a *semantic* editor. This reduces the need for extensive speech sampling by providing access to *meaning*. The editor is based around a time-aligned transcript produced by applying ASR to the original speech. Users visually scan the transcript to identify important phrases. They edit it using standard ‘cut and paste’ operations, which extract the corresponding underlying

speech. Even when the transcript contains errors, people should still be able to use it as a guide to direct their play and edit operations. We carried out an experiment comparing the semantic editor to a state-of-the-art acoustic editor. The experiment showed that even when there are multiple transcript errors, the semantic representation still allows users to target what they play, reducing the need for extensive sampling, and improving editing efficiency.

The paper structure is the following. We describe the semantic editor, contrasting it with state-of-the-art acoustic editing techniques. We then present a laboratory evaluation comparing semantic and acoustic editors for speech editing tasks.

SEMANTIC VERSUS ACOUSTIC EDITING

Acoustic Editing

Figure 1 shows a state-of-the-art speech editor Cool Edit 2000, one of the most frequently used programs of its type. Cool Edit 2000 was ranked most popular audio download by sites such as CNET, winshareware.com, and PC-World. Its interface is similar to other popular speech editors, such as Goldwave [5].



Figure 1 – Cool Edit 2000 – An Acoustic Editing Interface

The main interface representation in Cool Edit shows an acoustic signal. The interface can show two separate channels. In each, the y-axis shows amplitude and the x-axis time. The representation allows users to infer when the signal contains speech and when there is silence - indicated by a signal of zero amplitude.

The user clicks and sweeps over the waveform to select different parts of the underlying speech, which are highlighted in white (see Figure 1). The middle center of the figure also shows a time value (0:24.614) in minutes, seconds and fractions of a second. This indicates the beginning of the currently selected region. Under the waveform to the right is a table showing the absolute times for the beginning, end and length of the selected region.

People use standard edit commands such as cut or copy on a selected region of speech, available from the file menu. They can then open a new file in a separate window and copy edited regions using standard paste commands.

A left mouse click plays a selected region - with the cursor tracking to indicate exactly what is currently being played. Under the waveform to the left are various controls for other play operations, e.g. stop, play, pause, fast forward, fast rewind, move to beginning or end of file.

There are numerous problems with acoustic editing stemming from the indirect relation between the acoustic representation and the underlying speech. The lack of correspondence between waveform and meaning makes it difficult to precisely identify the beginning and end of relevant phrases. Users often have to listen to the entire speech record, with edits then requiring multiple play operations to locate precise phrases or words.

Semantic Editing

Figure 2 shows the semantic editor (with identifying information removed). The most obvious contrast with the acoustic editor is the nature of the speech representation. Speech is represented as text, segmented into paragraphs, generated by applying ASR to the original speech signal. The transcript representation allows one to see the gist of the speech (in this case a voicemail message) at a glance: *“please call 886 7888 to obtain your personal information...”*. Above the transcript to the right is a player with a timeline representation of the speech and simple play commands. We first outline the ASR transcription method, and then describe how the semantic editor works.

Speech Recognition and Transcript Generation

We generate transcripts by first segmenting the speech into “paragraphs”, using acoustic information, classifying the recording conditions for every audio paragraph. We then apply relevant acoustic and language models to each, as described in [2]. We concatenate ASR results for each audio paragraph so that for every “speech document” we have a corresponding ASR transcript.

It is important to note that ASR transcripts contain errors, and word error rates averaged 28% in our working system. The errors made by the recognizer are deletions, insertions and substitutions of the recognizer’s vocabulary. So, if the target speech contains words that are not in the recognizer’s vocabulary, this leads to word substitution errors. In addition, recognition errors can cascade: the underlying language model explicitly models inter-word relationships, so that one misrecognition may lead to others.

Semantic Editing Operations

Figure 2 shows how speech is represented by the errorful transcript, which is aligned with the underlying speech. Users visually scan the transcript to identify relevant words and phrases for editing. Edits to the speech are carried out directly on the transcript using standard text ‘cut and paste’

editing operations; users select relevant paragraphs, words and phrases, and delete less relevant ones. Figure 2 shows that the user has highlighted the phrase “*received by eleven fifty nine pm eastern time on October twenty ninth two thousand two*”. Cutting and pasting the phrase removes the

text from the transcript, adding it to another file. More importantly, edits to the transcript have the effect of editing the underlying time aligned speech, in this case, removing the corresponding speech from the archive, and moving it to a new location.

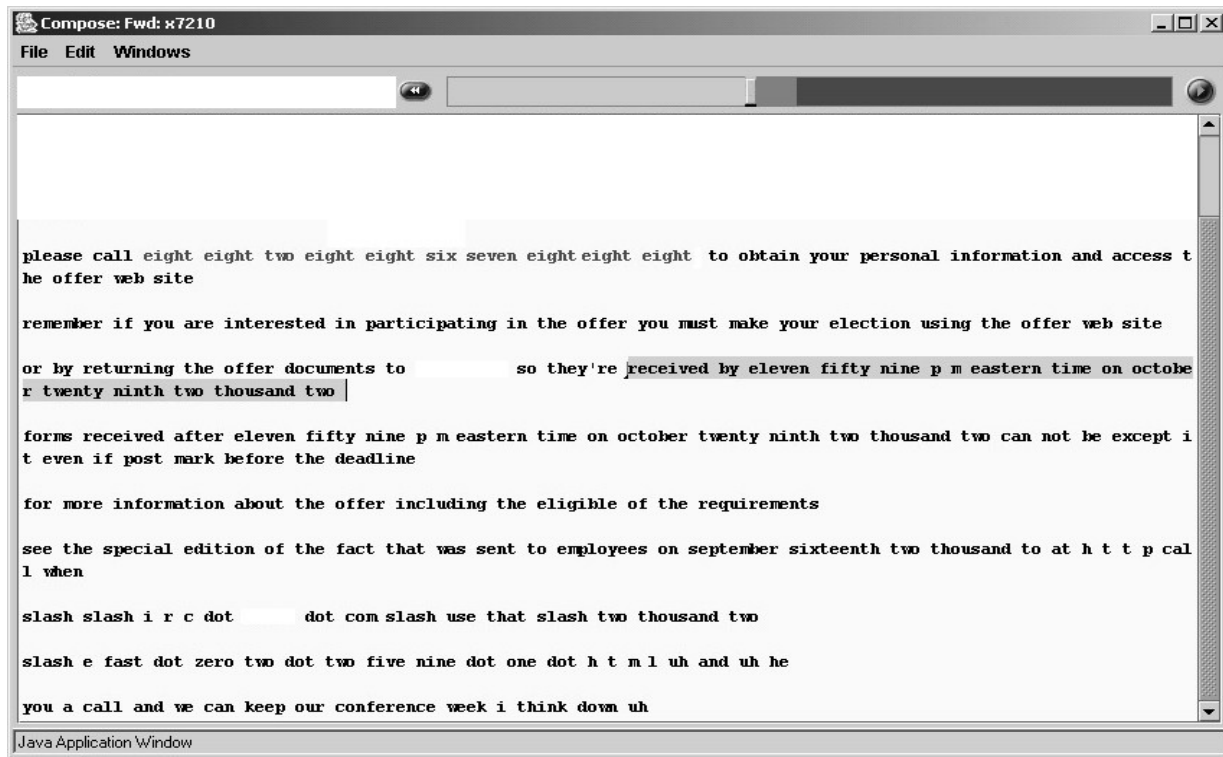


Figure 2 – The Semantic Speech Editor

One obvious limitation of this technique is that the transcript often contains errors. ASR errors mean that before editing, users must check what they read in the transcript by playing the original speech. Despite the presence of errors, the semantic representation usually provides enough information to determine the *gist* of what was said. Users can therefore use *gist* to navigate to relevant regions of the transcript, checking the transcript in just these regions by playing the speech. By using the transcript as a partial guide to focus on relevant regions, users reduce the need for extensive sampling of the underlying speech. Previous work on speech access has shown that errorful transcripts can be used in a similar way to successfully browse and search complex speech data [13,15].

Furthermore, it’s often possible to determine when the transcript can’t be trusted, as ASR errors are often signaled by the transcript failing to make sense. For example, the fifth paragraph in Figure 2 “*for more information about the offer including the eligible of the requirements*” is only partially comprehensible. If this information is critical to the editing task, users should find out what was actually said before editing. To play this material, users highlight that region of the transcript and

then hit the play control in the player. If the material is relevant it can be directly edited, as described above.

In sum, semantic editing represents speech textually. Unlike acoustic editing it provides (approximate) access to meaning. This allows users to more directly navigate and manipulate the underlying speech, reducing the amount of speech that they have to play. Even when the transcript contains errors, people can still use it as a partial guide to better direct play and edit operations.

EXPERIMENT COMPARING ACOUSTIC AND SEMANTIC EDITING

We compared acoustic and semantic editing in a laboratory experiment. We wanted to determine whether semantic editing was indeed more effective than a state-of-the-art acoustic editor. We also wanted to probe a potential weakness of semantic editing. We were concerned that semantic editing would be effective only when transcripts were accurate, but that inaccurate transcripts might mislead users into selecting irrelevant information.

We asked users to edit a series of speech recordings, identifying and extracting specific parts of voicemail messages. We chose voicemail editing as our target task,

as voicemail is an important and prevalent speech resource in the workplace. Users often have to extract important information from voicemail, but its utility is restricted by the fact that it is very difficult to edit, access and forward to others [12,13,14].

Procedure

Introduction and Tutorials. The experiment was run using a series of webpages. Users were first given the following general instructions:

“Imagine that one aspect of your job is to identify specific information in voicemail, and communicate it to busy colleagues. For each of the following four messages, we will specify two pieces of information in the form of specific phrases that we want you to extract from the message in order to forward to others. For the sake of the task, please imagine that you want to be as accurate as possible with your edits. You want the extracted information to contain no extraneous information.”

We then gave users a brief web based tutorial explaining both editors. They carried out a practice task twice, once with each editor. The task was similar to those used in the actual experiment. Users were allowed as long as they liked to complete these tasks, until they felt comfortable with each editor, and the procedure. They had to complete the practice task successfully with each editor before they were allowed to move on to the experimental tasks.

Experimental Tasks. The experimental tasks involved editing 4 voicemail messages extracted from a naturally occurring corpus. All 4 messages were for general distribution within the company in which all the users worked. This meant that they could be easily understood by the users; they did not contain private or unfamiliar information. None of the messages were replies to earlier messages so they did not require prior voicemail context for their interpretation.

For each task we presented users with a voicemail message and asked them to edit it with both editors, extracting specific phrases as accurately as possible. To control for potential learning and task sequence effects, we varied both the order in which users received the different tasks and also which editor (semantic or acoustic) they used to carry out the task first.

One (anonymized) message was the following:

“This is an important message for [] employees. []’s offer to exchange outstanding options for restricted stock units and cash is scheduled to expire on October 29th 2002. If you are an eligible employee and have not received your pin information from [] stock plan incorporated, please call 888 828 8678 to obtain your personal information and access the offer website. Remember if you are interested in participating in the offer you must make your election using the offer website or by returning the offer

documents to [] so they are received by 11:59 pm Eastern time on October 29th 2002. Forms received after 11:59 Eastern time on October 29th 2002 cannot be accepted even if postmarked before the deadline. For more information about the offer including the eligibility requirements see the special edition of ESAP that was sent to employees on September 16th 2002 at [http://irc.\[\] .com/esap/2002/esap.2.259.html](http://irc.[] .com/esap/2002/esap.2.259.html).”

We asked users to edit this message to extract two phrases containing: (1) the first mention of exactly when (date, time, time zone) offer documents must be returned to the company; and (2) the URL containing the relevant information. As far as possible, we controlled across messages the location and complexity of the information to be extracted. Users edited the message and we saved the results for analysis.

After completing each task, we gave users a brief web survey, asking them to compare (a) how easy the tools made editing, and (b) how likely they would be to select each for a future similar task. Responses were generated as 5-point Likert scales, by asking users to state their degree of agreement with assertions such as “*Semantic edits made it easy to edit the voicemail message, compared with acoustic edits*”. We also administered a final survey after all tasks: users compared the editors’ support for (a) identifying relevant regions of speech for editing, (b) extracting regions of speech from the message. In addition, we asked two open-ended questions about what users perceived to be the main differences between the editors, and why they preferred one editor to the other. We also recorded spontaneous comments made during the experiment about the editors or the tasks.

Variables and Measures

Editor type. We compared the semantic editor with Cooledit. Users carried out each editing task twice – once with each editor.

ASR accuracy. We also examined the effects of transcript accuracy on editing. Semantic editing performance is clearly dependent on the quality of the transcript. When ASR accuracy is high, we expect semantic editing to be fast and accurate; the transcript is a reliable representation of the underlying speech, making users less dependent on playing. When ASR accuracy is low, however, users may be misled by inaccurate transcripts. By relying on the transcript, they may edit faster than with an acoustic interface but make more errors as a result. We therefore selected two voicemail messages where transcript word accuracy was high (mean 80.8% for these two high accuracy transcript tasks) and two where it was low (mean 36.5% for two low accuracy transcript tasks). These values were chosen as they were one standard deviation above and below the mean for the voicemail corpus. Note that testing the effects of having ASR

accuracy at 36.5% represents a stringent test of semantic editing, as roughly 2 out every 3 words are incorrect.

Message length. We also varied message length, expecting that users would perform better on shorter messages, which should reduce the problem of audio navigation. We therefore selected two short and two long messages. Message lengths were 70s. for the two shorter messages and 93s. for the two longer messages.

Measures. To determine how well each editor performed, we measured (a) the *time* to complete each editing task; and (b) the *quality* of the edits carried out. Quality was scored as follows. A coder blind to the experimental condition listened to each edit to determine whether it was a verbatim match to the two target phrases for that task. Criteria for accurate edits were strict, with accuracy being determined at the syllable level. If an edited phrase contained an extraneous or a missing syllable it was scored as only partially correct. Each message edit was evaluated against the following scoring scheme. Each of the two parts of the edit for the message was scored out of 2, with a maximum of 4 for each message. Correct edits for each part containing no extraneous information were scored as 2, and edits containing no correct information as 0. Partially correct edits for each part (score = 1), were defined as either containing (a) all target information but with extraneous information, or (b) a subset of the target information. Scores therefore ranged from 0-4 for each message. Edits that contained all the relevant information and no extraneous information for both parts of the message were scored as correct ($2 \times 2 = 4$). Those containing either incomplete information or complete information with extraneous syllables were scored as (1-3), depending on how much correct non-extraneous information they contained. Those containing none of the target information were scored as incorrect (0).

Users

Sixteen users (6 women and 10 men, ranging in age from 25 to 57) took part. Users were volunteers consisting of researchers, administrative staff and marketers at a large corporate research lab. They had no prior knowledge of the project or experimental hypotheses, but all had more than five years experience of voicemail and standard PC software. The entire procedure took about forty minutes and users were given a small food reward for participating.

Hypotheses

Our hypotheses were the following:

- H1: *Editor type.* Users should generate faster, higher quality edits with the semantic, compared with the acoustic editor. We also expected users to rate the semantic editor higher in the subjective assessments.

- H2: *ASR Accuracy and Semantic Edits.* Edits with the semantic editor should be faster and higher quality with accurate, compared with inaccurate transcripts.
- H3: *Acoustic Versus Semantic Edits for Inaccurate ASR.* We expected that users would be misled by inaccurate transcripts; although these should be edited more quickly with the semantic than the acoustic editor, users should make more editing errors overall. We also expected users to rate the semantic editor as better.
- H4: *Message Length.* Shorter messages should produce faster, higher quality edits, regardless of editor. Ratings should also be better for shorter messages.

Analysis Methods. We tested our hypotheses using analysis of variance (ANOVA). To investigate editor type and length effects, we conducted 4 ANOVAs with editor type, message length and interface order (acoustic or semantic editor first on a given task) as independent variables. The dependent variables were (a) response time, (b) quality of solution, (c) judgments about utility of the tool for the editing task, (d) subjective predictions about whether they would use the tool again for similar editing tasks.

Editor	Time (secs.)	Quality (max=4)	Tool Effectiveness (max=5)	Expected Future Use (max=5)
Acoustic	176.4	2.7	2.0	2.0
Semantic	120.3	3.2	4.1	4.1
Prediction Confirmed?	Yes	Yes	Yes	Yes

Table 1: Overall Comparison of Two Editors Showing Score Means

To investigate ASR accuracy effects, we conducted ANOVAs for the semantic editor only, with time and quality as dependent variables. We excluded acoustic editor data, because ASR accuracy is irrelevant for acoustic editing. We also did not analyze ratings data, as this required comparisons between acoustic and semantic editors. The independent variables in this analysis were ASR accuracy, length and order.

To investigate the effects of editor type for inaccurate ASR we used time, quality and ratings as dependent variables. We excluded data from messages where ASR accuracy was high, comparing semantic editing for inaccurate ASR with matched tasks for the acoustic editor. Independent variables were editor type, length and order.

Results

Editor type. Table 1 shows our hypotheses about the general superiority of semantic editing were confirmed. Semantic edits were faster, higher quality and more

highly rated than acoustic edits. Specific results were as follows:

- *Time* Users carried out edits faster overall with the semantic compared with the acoustic editor ($F(1,120)=31.9, p<0.0001$).
- *Edit Quality* Users generated higher quality edits with the semantic, compared with the acoustic editor ($F(1,120)=10.9, p<0.001$).
- *Ratings* Users rated the semantic editor as a better tool for carrying out the editing task ($F(1,120)=135.3, p<0.0001$), and said they would be more likely to use it than the acoustic editor for similar future editing tasks ($F(1,120)=111.2, p<0.0001$).

ASR Accuracy	Time (secs.)	Quality (max=4)
Low	143.7	2.8
High	97.0	3.5
Prediction Confirmed?	Yes	Yes

Table 2: Effects of ASR Quality on Semantic Editing

ASR Accuracy. Table 2 shows that, as we expected, transcript accuracy was important when using the semantic editor. Accurate transcripts produced higher quality, faster edits.

- *Time* Users carried out edits faster with accurate than inaccurate ASR ($F(1,56)=30.1, p<0.0001$).
- *Edit Quality* Users generated higher quality edits with accurate ASR compared with inaccurate ASR ($F(1,56)=12.7, p<0.0001$).

Editor	Time (secs.)	Quality (max=4)	Tool Effectiveness (max=5)	Expected Future Use (max=5)
Acoustic	179.3	2.8	1.8	1.7
Semantic	143.7	2.8	3.8	3.7
Prediction Confirmed?	Yes	No	Yes	Yes

Table 3: Comparison of Editors for Tasks Where ASR Was Inaccurate

Acoustic versus Semantic Edits for Low Quality ASR. Table 3 shows our concerns about over-reliance on low accuracy transcripts were not borne out. Users were not seduced by the transcript into making fast but inaccurate edits. Low accuracy transcripts were edited more quickly using the semantic editor, but counter to our predictions, there was no evidence that these edits were lower quality than acoustic edits. Ratings for the semantic editor were higher for tool choice and for expected future use.

- *Time* Users carried out edits faster with the semantic than the acoustic editor ($F(1,56)=5.8, p<0.02$).

- *Edit Quality* Edit quality was equivalent for the two editors ($F(1,56)=1.4, p>0.05$).
- *Ratings* Users rated the semantic editor as a better tool for carrying out the editing task ($F(1,56)=28.2, p<0.0001$), and said they would be more likely to use it than the acoustic editor for similar future editing tasks ($F(1,56)=19.7, p<0.0001$).

Message Length. Table 4 shows that, as we expected, shorter messages are edited more quickly, although these edits are not higher quality, nor are ratings always higher. This indicates partial confirmation of our predictions.

- *Time* People were quicker to edit short than long messages ($F(1,120)=11.2, p<0.001$).
- *Edit Quality* Length had no effect on editing quality ($F(1,120)=2.0, p>0.05$).
- *Ratings* Length affected judgments of usefulness as a tool ($F(1,120)=10.7, p<0.001$), but not likelihood of future use ($F(1,120)=0.5, p>0.05$).

Transcript Length	Time (secs.)	Quality (max=4)	Tool Effectiveness (max=5)	Expected Future Use (max=5)
Long	165.0	3.0	2.8	3.1
Short	131.7	2.8	3.1	3.0
Prediction Confirmed?	Yes	No	Yes	No

Table 4: Comparison of Long and Short Messages

Finally, we looked at interface order effects across the analyses. We found that people were faster overall if they first edited a message using the semantic editor ($F(1,120)=30.1, p<0.0001$). This may be because it exposes them to more detailed information about the message, making their subsequent acoustic editing task more straightforward. There were no other order effects or interactions.

User Comments and Strategies

We analyzed users' responses to the open-ended survey question about the main differences between the editors, along with the spontaneous comments they made during the experiment. This gave us important information about the different strategies used with each editor.

Identifying Relevant Parts of the Message. One set of comments detailed how the semantic editor helped identify relevant regions of the message, which were then played to determine exactly what was said, before editing. There were two main strategies for finding relevant regions in the transcript. These were scanning the message for *gist*, or looking for *key words*.

"[semantic editing] is useful, because I can look at the message to get the gist and then just play very specific parts to hear what's said."

"I look for keywords with the [semantic editor]. These allow me to pinpoint where the material is so I can play it."

At the same time, users were aware that there were errors in the transcripts making them only an approximation to what was actually said.

"in [semantic editing] you can't go by exactly what's there, but it's still better than [acoustic editing]. You can get the gist with [the semantic editor], even when there are errors."

By using the transcript to identify specific regions of the transcript for detailed attention users were able to play much less speech than with an acoustic editor, making editing more efficient.

"there's no structure with [the acoustic editor], so I have to listen to the whole thing."

"it's easier to just listen to the entire message the first time with [the acoustic editor], that way I'm sure I haven't missed anything and won't need to go back"

These comments about semantic editing providing navigational information were also supported by survey ratings. Users generally agreed with the statement *"[semantic editing] made it easy to find where the relevant information is located in the voicemail message, compared with [acoustic editing]."* Their mean Likert rating was 4.1 (where 5 is totally agree), which is significant on a one sample t test ($t(15)=7.3, p<0.001$).

Editing the Relevant Material. Once people had located specific relevant material, they thought it was much easier to extract this with semantic editing. On some occasions this was done directly, without listening, although in general users were highly attentive to errors in the transcript.

"Sometimes I could edit the transcript directly without listening. But even when the [semantic editor] made errors, I still had a sense of where [the material] was in the text. With the waveform, I had no idea."

"When [the semantic editor] gets the transcript mostly correct, the task is trivial."

In contrast, acoustic editing is laborious because it's extremely painstaking to determine exactly where relevant material is located.

"even when I know roughly where it is, it's still annoying trying to locate the exact start and end points. You need to get the exact time code, but it's hard to listen and remember [the time code] at the same time."

Again, these comments were supported by users' survey responses. They overwhelmingly agreed with the statement *"[the semantic editor] made it easy to extract the relevant information from the voicemail message,*

compared with [acoustic editor]." Their mean Likert rating was 4.3 (where 5 is totally agree), which is significant on a one sample t test ($t(15)=8.2, p<0.001$).

User Strategies. By watching users carrying out the task, we also noticed different strategies emerging for each interface. In the semantic editor, users typically began by listening and reading along with the message. During the course of the tasks users refined their extraction strategies to listen only to the necessary pieces of the message, marking broad clip boundaries using the text and refining them by listening to the underlying speech. For the acoustic editor, a number of initial strategies were used. All were unsuccessful attempts to minimize the amount of speech that needed to be played. One user played from the middle and browsed forward and backward to locate the desired passages. Others simply sampled the audio and attempted to mark regions that had a high probability of containing the passage. No matter what their initial strategy, in the end, the majority of acoustic editor users abandoned all optimization strategies and just listened to the entire message. They quickly realized that any other strategy hindered their performance and they might just as well spend the time listening to the message.

New Editor Features

Several users also commented about whether we might be able to automatically provide information about transcript accuracy in the interface, allowing them to identify reliable transcripts at a glance. Other comments concerned new features. Users wanted to be able to correct errors they found in the transcript; they pointed out that there was little sense in forwarding known transcription errors to others. Second, they wanted to annotate the edited transcripts to provide explanations and context for the speech that they have edited, in a way that is similar to current practices for complex email responding and forwarding behavior (e.g. using >> or other quoting behavior).

CONCLUSIONS

We have developed a novel technique for editing speech data, based on a semantic rather than an acoustic representation. Our experiment showed, as we expected, that people edit faster using a semantic editor than a state-of-the-art acoustic editor. User comments suggest that the semantic editor allowed them to visually scan the transcript to identify relevant regions and then play these to identify precisely where to edit. By doing this, they were able to restrict their attention to relevant regions and carry out their task more quickly.

One of our initial concerns was that over-reliance on inaccurate transcripts would compromise the quality of user edits. However, peoples' comments show that they are aware of transcript imperfections. Although semantic edits of inaccurate transcripts are slower and lower quality

than edits of accurate transcripts, the more important comparison is with acoustic editing. Here we found that even when transcript accuracy was as low as 36.5%, semantic editing was faster, and as accurate, as acoustic edits.

One important design implication is that we need to move away from general-purpose acoustic tools for processing speech. Acoustic editors are designed to deal with all forms of audio data, but speech editing has specific demands, that are not well met by such general tools. By building tools that are specifically tailored to represent *meaning*, we can provide more effective ways to process speech.

Further design implications arise from user comments about the semantic editor. One challenge is to indicate to users that a transcript is inaccurate. One possibility is that we might use confidence information from the speech recognizer to signal this [13]. Regions of low ASR confidence could be grayed in the transcript to alert users to areas of potentially poor quality.

Users also wanted to be able to correct transcripts and comment on their edits. We have therefore extended our semantic editor to: (a) allow users to correct original transcription errors; (b) combine edited transcripts with explanatory user textual comments. There are some complex design issues to be explored here. Corrections and added comments must be clearly visually distinguished from the original transcript so they cannot be confused with it. We also need to determine both how and whether selecting corrections might lead parts of the original message to be played. If we allow corrections to trigger playing, there are potentially complex problems in aligning them with the underlying speech. And corrections need to be distinguished from other orienting comments that users wanted to add to explain their edits. Together these features lead to a novel type of multimedia object which mixes textual and spoken data, with the text serving as an explanation and index into the underlying speech.

In conclusion, semantic editing is better and faster for accurate ASR and more efficient than acoustic editing even when transcription is poor. These results are highly promising, suggesting that semantic editing may remove a major barrier to making speech into useful data.

ACKNOWLEDGMENTS

We thank our experimental subjects for their time and patience and for all the researchers who contributed ideas at various stages of this project.

REFERENCES

1. Arons, B. SpeechSkimmer: A system for interactively skimming recorded speech. *ACM Transactions on Human Computer Interaction*, 4(1), 38, 1997.

2. Bacchiani, M., Hirschberg, J., Rosenberg, A., Whittaker, S., Hindle, D., Isenhour, P., Jones, M., Stark, L., and Zamchick, G. SCANMail: Audio Navigation in the Voicemail Domain. In *Proc. of the Workshop on Human Language Technology*, 2001.
3. Chalfonte, B., Fish, R., and Kraut, R. Expressive richness. In *Proc. CHI91*, 21-26, 1991.
4. Degen, L., Mander, R., and Salomon, G. Working with audio. In *Proc. CHI92*, 413-418, 1992.
5. Goldwave Digital Audio Editor. <http://www.goldwave.com/>
6. Hauptmann, A. and Witbrock, M. Informedia: News-on-Demand Multimedia Information Acquisition and Retrieval, In M. Maybury (Ed.), *Intelligent Multimedia Information Retrieval*, AAAI Press, 213-239, 1997.
7. Hindus, D., Schmandt, C., and Horner, C. Capturing, structuring and representing ubiquitous audio. *ACM Transactions on Information Systems*, 11, 1993.
8. Jones, G., Foote, J., Spärck Jones, K., and Young, S. Retrieving Spoken Documents by Combining Multiple Index Sources, In *Proc. SIGIR*, 30-38, 1996.
9. Kazman, R., Al-Halimi, R., Hunt, W., and Mantei, M. Four paradigms for indexing videoconferences. In *IEEE Multimedia*, 3(1), 63-73, 1996.
10. Schmandt, C. The Intelligent Ear: A Graphical Interface to Digital Audio, *Proceedings, IEEE International Conference on Cybernetics and Society*, IEEE, Atlanta, GA, 1981.
11. Stifelman, L., Arons, B., and Schmandt, C. The audio notebook: paper and pen interaction with structured speech. In *Proc. CHI2001*, 182-189, 2001.
12. Whittaker, S., Davies, R., Hirschberg, J., and Muller, U. Jotmail: a voicemail interface that enables you to see what was said. In *Proceedings of CHI2000 Conference on Human Computer Interaction*, 89-96. New York: ACM Press, 2000.
13. Whittaker, S., Hirschberg, J., Amento, B., Stark, L., Bacchiani, M., Isenhour, P., Stead, L., Zamchick G., & Rosenberg, A. SCANMail: a voicemail interface that makes speech browsable, readable and searchable. In *Proceedings of CHI2002*, New York: ACM Press, 275-282, 2002.
14. Whittaker, S., Hirschberg, J., and Nakatani, C. H. All talk and all action: strategies for managing voicemail messages. In *Proceedings of CHI98 Conference on Computer Human Interaction*, New York: ACM Press, 1998.
15. Whittaker, S., Hirschberg, J., Choi, J., Hindle, D., Pereira, F., and Singhal, A. SCAN: designing and evaluating user interfaces to support retrieval from speech archives. In *Proc. of SIGIR99*, 26-33, New York: ACM Press, 1998.
16. Whittaker, S., Hyland, P., and Wiley, M. Filochat: handwritten notes provide access to recorded conversations. In *Proc. of CHI94 Conference on Computer Human Interaction*, 271-277. New York: ACM Press, 1994.
17. Wilcox, L. Chen, F., Kimber D. and Balasubramanian, V. Segmentation of Speech Using Speaker Identification. *Proc. International Conference on Acoustic Speech and Signal Processing*, 1994.