The JABBERWOCKY: A semiautomated system for the transcription of verbal protocols*

RUVEN BROOKS and LEE ERMAN
Carnegie-Mellon University, Pittsburgh, Pennsylvania 15213

and

RICHARD NEELY
Xerox Palo Alto Research Center, Palo Alto, California

The JABBERWOCKY is a system that greatly simplifies the transcription of verbal protocols by acting as a "smart" tape recorder which plays exactly a phrase at a time. In addition, it provides timings accurate to a centisecond. The final result is available as a file on a computer system that then may be used for other editing or processing.

Verbal protocols of human problem solving have been shown to be a rich type of data on which to base models of human problem-solving behavior (Newell & Simon, 1972); unfortunately, transcription of protocols, especially those more than a few minutes in length, is an extremely tedious task. The JABBERWOCKY is a semiautomated system designed to make protocol transcription much easier. Use of the JABBERWOCKY makes the transcription and analysis of verbal protocols easier in three ways: It plays exactly a single phrase at a time for transcription by the user, thus eliminating the difficulties associated with running a tape recorder backward and forward. It also provides timings for phrases that are reliably accurate to a centisecond. Finally, the finished transcription exists as an ordinary text file on a computer system; the user may then use a wide range of computer editing and formatting programs to process the transcription. The JABBERWOCKY system operates on two types of information: a digitized version of the original speech input, called ADC information (for analog-to-digital conversion), and a set of parameters extracted from the original speech signal, called ZCC information (for zero crossing counter). The ADC information is used for playing back sound, while the ZCC information is used for finding the boundaries of the phrases to be played for the user.

HARDWARE

The system runs on the Carnegie-Mellon University Computer Science Department Digital Equipment Corporation PDP-10 computer under a time-sharing and multiprogramming operating system. Currently, the computer configuration has 192,000 36-bit words of core memory, 330,000 words of swapping drum storage, and more than 18 million words of disk storage. To provide the ADC information, a 9-bit analog-to-digital converter running at a 10-kHz sampling rate is used. The ZCC information is acquired through specially built hardware known as the zero crossing counter. This special hardware preprocessor filters the audio signal into six bands and produces for each band a count of the zero crossings and the maximum amplitude in each consecutive 10-msec sampling period. The phrase detection program uses only the zero crossing information for the 3,200- to 6,400-Hz band and the unfiltered amplitude. Audio output from the computer is provided by a D/A converter. All these devices are connected to the PDP-10 via the I/O bus.

(This hardware and most of the software were developed as part of a project on the computer recognition of speech. As such, it is much more powerful than that required for the transcription system alone. It is the opinion of these authors that sufficient power for the transcription system could be obtained from a minicomputer, such as the Digital Equipment Corporation PDP-11, equipped with an analog-to-digital converter, 32,000 memory locations, and sufficient disk space.)

Table 1
Most Commonly Used System Commands

Command Character	Function
<carriage return>	Typed by itself, it causes the current phrase to be played.
C	Plays the current phrase surrounded by context.
<alt-mode>,<esc>,$	Plays the phrase previous to the current one.
X	Used to enter a transcription for the current phrase or to extend an existing partial transcription.
A	This command permits within-line alterations of the transcription of a phrase without the necessity for retyping the entire phrase.
S	This command permits skipping over a phrase without transcribing it. It is useful for skipping over noise or unmeaningful sound.
O	This command outputs the current transcription and goes on to find the next phrase.
←<variable>=<value>	Sets the named variable qual to the given value.
=<variable>	Types out the current value of the variable.

*This work has been supported in part by the Advanced Research Projects Agency of the Secretary of Defense (F44620-70-C-0107) which is monitored by the Air Force Office of Scientific Research and in part by a grant from Xerox Corporation.

Table 2
An Example of the System in Use

In this example, audio output will be indicated by double quotation marks. Explanatory comments which would not appear in actual system use will be surrounded by parentheses. Input typed by users will be in lowercase. < cr > indicates a carriage return.

```
"This is a dem-"  (First, a phrase is found and played.)
*c                (Play with context.)
"This is a demonstration."
*minpausing=80    (Increase the minimum length of a pause.)
*r                (Try finding a phrase using the new value of MINPAUSLING.)
"This is a demonstration."
*x                (This command is used to record a transcription.)
This is a demonstration. (Types in transcription.)
*o                (This command is used to output a transcription.)
THIS IS A DEMONSTRATION. (System types transcription back out.)
CONFIRM(O):o      (asks for confirmation.)
"of the JABBERWOCKY system" (Another phrase is played.)
*<crlf>           (User plays phrase again.)
"of the JABBERWOCKY system"
*x                (User wants to make a transcription.)
of the JABBERWOCKY system.
*o
OF THE JABBERWOCKY SYSTEM.
CONFIRM(O):o      (User gives confirmation.)
*e
CONFIRM(E):       (Exits system.)
```

SOFTWARE

Software for the system is written in SAIL (Swinehart & Sproull, 1970), an ALGOL-like language which also has string processing and powerful, convenient I/O capabilities. The software consists of three programs: JAWS, CLAWS, and EYES. The JAWS program performs the speech digitization and parameter extraction and creates sets of disk files containing this information which are later used by the CLAWS program to find and play phrases. Having this operation performed by a separate program has several advantages: The user need not be concerned with mounting a tape of the recorded sound each time he wants to use the system. It makes it possible to play any portion of the recording without rewinding the tape, and it permits the user to defer those portions of the system which require a great deal of processor time to off hours while still allowing him to do transcriptions at any time.

The second part of the system, the CLAWS program, finds phrases of speech, plays them back to the user acoustically, and records the user's transcription of the phrases. The algorithm used to find the phrases is the following: Names given in capital letters are names of quantities that may be altered by the system user to alter the characteristics of phrase detection.

In order to find a phrase, the system begins by looking for pauses. A pause is any section of sound that meets three conditions: It must be longer than the value of MINPAUSLNG, it must have zero crossing counter values less than Z5MAX, and it must have an amplitude less than AUMAX. Once two such pauses have been found, the sound between them is considered to be a phrase if it is less than MAXPHRLNG; if it is longer, the system arbitrarily ends the phrase at MAXPHRLNG. If the phrase thus found is shorter in length than MINPHRSLNG, it is extended into the pauses on both sides until it is this length. The values of Z5MAX and AUMAX are usually set high enough to prevent random noise from being included in the phrase. This has the side effect of cutting off some of the beginning and end of the phrase; to correct for this, a certain length of sound, BSLACK, is added to the beginning of the phrase. Simiarly, an amount, ESLACK, is also added to the end of the phrase. The total phrase found after these corrections, which has minimum length MINPHRSLNG + BSLACK + ESLACK, is then played by the system.

Currently, CLAWS permits the user to play the current phrase simply by typing <carriage return> at his terminal. He can also play the phrase with a variable amount of surrounding context, and he can play the phrase previous to the current one. He can also examine and alter the variables which set the parameters for phrase detection and then cause the system to try again to find a phrase using the new values. He may also override the phrase detection entirely and set phrase boundaries manually. After hearing a phrase, he may either enter a transcription of it or choose to skip over it entirely. Once he has transcribed a phrase, he may then go back and edit it.

Final output of the CLAWS program is known as a phrase file; it consists of the transcription plus the acoustic information necessary for timing and for synchrony with the ADC file. To process this information into easily usable form, the third program, EYES, is run. Output from the EYES program consists of a file containing, for each phrase, the transcription of the phrase, the time in centiseconds from the beginning of the tape to the beginning of the phrase, and the length of the phrase in centiseconds. This file is then available for any other processing that the user desires to do.

REFERENCES

Newell, A., & Simon, H. A. *Human problem solving*. Englewood Cliffs, N.J: Prentice-Hall, 1972.
Swinehart, D., & Sproull, R. SAIL manual. Carnegie-Mellon University Computer Science Department, May 1970.

(Received for publication December 3, 1972;
revision received February 8, 1973.)