A Prototype Personal Dictation System

Adam Janin
International Computer Science Institute
1947 Center St., Suite 600
Berkeley, CA 94704
janin@icsi.berkeley.edu

ABSTRACT

We describe a prototype personal dictation system. As a user speaks, the system produces a real-time audio transcript. The user can correct and annotate the transcript using a graphical user-interface (UI) running on a handheld computer. The speech recognition runs on a workstation. Although the two are currently connected via a wired network, a wireless connection is planned. The primary focus of this paper is the UI for correcting the transcript.

INTRODUCTION

We are in the process of developing Meeting Recorder, a portable device that records meetings in uninstrumented, natural environments. The Meeting Recorder will support multiple speakers, allow correction and annotation of the transcript, support indexing and searching of the audio record, and will be self-contained using Vector IRAM[1]. The full Meeting Recorder project is very ambitious, involving research in automatic speech recognition (ASR), speaker and topic tracking, information retrieval, collaboration, annotations, and small form-factor UIs. It also requires a chip that will not be available for another year or so.

To get a handle on some of the infrastructure and UI issues of the Meeting Recorder project, we developed an intermediate testbed, a Personal Dictation System. It allows a single user to dictate text in real time. The text appears on the screen of a Palm Pilot (a handheld computer). The user can then correct the transcript using a pen interface. Limited annotation is also allowed.

SYSTEM ARCHITECTURE

Since the Pilot is not yet capable of providing speech recognition, the ASR system runs on a workstation connected to the Pilot via a network. Also, we used a headset microphone to limit the problems associated with background noise and reverberation. Although both the headset microphone and the wired network require the user to be tethered to a workstation, we will soon lift this restriction by providing a wireless network and microphone.

The ICSI hybrid ASR system was used to perform the speech recognition[2]. Although accuracy and throughput of the ICSI system are very good, it was not designed to be interactive. This caused some problems with user interaction. In addition, the ICSI system was trained on television and radio news broadcasts. Therefore, the system does much better with a news-like vocabulary and speaking style.

The UI runs on the Pilot. It allows the user to correct the transcripts and create new text. The components running on the Pilot communicate with the components running on the workstation using TCP/IP. In fact, three workstations were used. One captured the audio signal and performed signal processing, the second ran the ASR algorithms and the correction server (see below), and another was used to connect the Pilot to the network though the Pilot's cradle.

CORRECTING AND ANNOTATING

We distinguish annotation from correction. Correcting is the process of informing the recognizer that it has made an error. For example, if you say “a record day” and the transcript reads “the records pay”, you may want to inform the system that it is “day” rather than “pay”. Annotation, on the other hand, allows the user to change or add to the transcript. Even if the recognition is perfect, the user may want to modify the results or add additional marks. For the purposes of the Personal Dictation System, non-textual annotations (e.g. circling, underlining) were not supported.

Correction is useful for two reasons. First, if the recognizer has a good idea of the possible alternatives, it may be much faster to select one of these rather than deleting the incorrect text and then entering the correct text. Secondly, the recognizer can adapt to the user more efficiently if the correct transcript is provided. Additionally, the correction mechanism allows out-of-vocabulary words to be added.

The system must allow the user to specify a portion of the transcript to be corrected; generate alternatives to the selection; and update the transcript. The following paragraphs detail some of the proposed solutions, and the results of (very) informal user studies.

The first attempted method of specifying an incorrect portion of the transcript involved selecting the incorrect text using a standard press-and-drag method followed by tapping on a “Correct It” button at the bottom of the screen. This method was the easiest to implement, as the Pilot directly supports it. However, the interaction was slow, especially for inaccurate sections of the transcript. Also, the "Correct It" button takes up valuable screen real estate.

Next, we tried a method inspired by the Pilot's popup triggers. When the user presses and holds on a word, a popup list appears with alternatives for the word. A drawback of this paradigm is that it requires separate "Annotate" vs. "Correct" modes. Another drawback is that only a word can be specified, rather than entire phrases. Regardless, this interaction was preferred over the previous method.

The press-and-hold method had another problem as well. Frequently, a user would tap on a word rather than press-and-hold. Therefore, the final method chosen for specifying a recognition error was to tap on a word.

The system for generating alternatives and updating the transcript requires some understanding of how transcripts are produced. As the ASR system looks for possible matches to what the user said, it generates a lattice of words. Figure 1 shows an example excerpt from a lattice in which the user said "a record day". Each path from the root node to a leaf represents a hypothesized utterance. Each hypothesis receives a score. When ranked in order, the hypotheses produce an "N-best" list of possible utterances. The transcript is generated by selecting the hypothesis with the highest score (the first element of the N-best list). In this case, the transcript might read "the records pay".

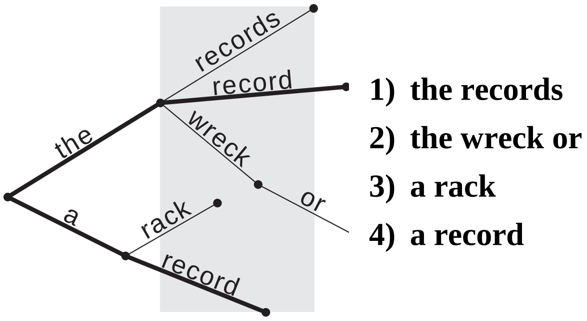
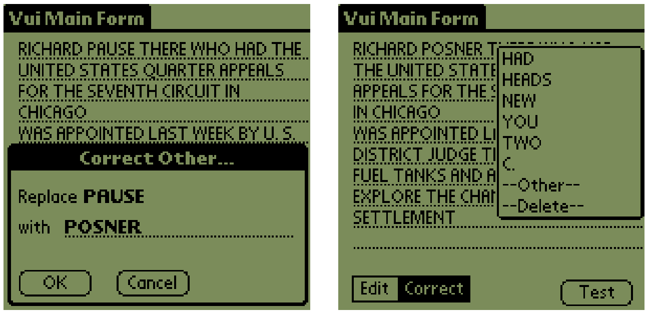


Figure 1 -- A Word Lattice and its N-best list.

When the user taps on the word "records", indicating that the system has made a recognition error, the correction server must generate all the possible alternatives for "records". Currently, the system picks all words that overlap in time. In Figure 1, this corresponds to all the words that overlap with the gray rectangle. The system pops up a box with the words ordered from most likely to least. In this example, the list would contain "records, record, rack, wreck, or". Full overlap is probably a poor heuristic, however. For example, the word "or" only overlaps by a small amount, and should probably be excluded from the list.

Once the user selects an alternative, the system must respond. The first method we tried selected the paths in the lattice containing the correct word as specified by the user. In Figure 1, if the user corrects "records" to "record", we select only those paths that contain "record" (highlighted in Figure 1). This corresponds to selecting those entries of the N-best list that contain "record". In this case, the best entry containing "record" is "a record", so we replace "*the* records" with "*a* record". Although this is, in fact, the correct hypothesis according to what the user said, it was highly unexpected behavior. The user tapped on "records", selected "record", and the word "the" changed to "a"! It appears to be better to change only the word the user specifies, even if the system is sure that additional changes are beneficial.



RICHARD PAUSE THERE WHO HAD THE
UNITED STATES QUARTER APPEALS
FOR THE SEVENTH CIRCUIT IN
CHICAGO
WAS APPOINTED LAST WEEK BY U. S.

Replace **PAUSE**
with **POSNER**

RICHARD POSNER T
THE UNITED STATE
APPEALS FOR THE S
IN CHICAGO
WAS APPOINTED L
DISTRICT JUDGE T
FUEL TANKS AND A
EXPLORE THE CHA
SETTLEMENT

Figure 2 – Screen shots of correcting the transcript. The user said "Richard Posner, who heads the United States Court of Appeals for the seventh circuit in Chicago..."

EXAMPLE SCREEN SHOTS

On the left of Figure 2, the user tapped on "PAUSE". However, since "Posner" wasn't in the dictionary, the user selected "—Other—" from the popup list. He then entered "POSNER". On the right, the user tapped on "HAD". Notice that "HEADS" is the second element in the popup list, allowing very easy correction.

The "Edit/Correct" toggle at the bottom of the screen allows easy and obvious mode switching between correcting the transcript and editing (annotating). Even with this screen element, some users became confused as to which mode they were in. Using a gesture to correct a word could provide a non-modal way of distinguishing between annotating and correcting. The "Test" button is a leftover debugging button, and is not part of the interface.

CONCLUSIONS

We presented a prototype personal dictation system. It provides a testbed for UI and infrastructure issues of the Meeting Recorder project. Design decisions and informal user tests for some correction mechanisms were presented.

NOTES

1 A chip being developed by the University of California, Berkeley. See http://iram.cs.berkeley.edu for details.

2 See http://www.icsi.berkeley.edu for information regarding ICSI's hybrid ASR system.