A Comparison of Speech and Typed Input

Alexander G. Hauptmann and Alexander I. Rudnicky

School of Computer Science
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Meaningful evaluation of spoken language interfaces must be based on detailed comparisons with an alternate, well-understood input modality, such as the keyboard. This paper presents an empirical study in which users were asked to enter digit strings into the computer by voice and by keyboard. Two different ways of verifying and correcting the spoken input were also examined using either voice or keyboard. Timing analyses were performed to determine which aspects of the interface were critical to speedy completion of the task. The results show that speech is preferable for strings that require more than a few keystrokes. The results emphasize the need for fast and accurate speech recognition, but also demonstrate how error correction and input validation are crucial components of a speech interface.

Although the performance of continuous speech recognizers has improved significantly in recent years [6], few application programs using such technology have been built. This discrepancy is based on the fallacy of equating speech recognition performance with the usability of a spoken language application. Clearly, the accuracy of the speech recognition component is a key factor in the usability of a spoken language system. However other factors come into play when we consider a recognition system in the context of live use.

For example, system response time has direct consequences for system usability. Various studies have shown that the amount of delay introduced by a system significantly affects the characteristics of a task (such as throughput) as well as human performance (such as choice of task strategy) [3, 15]. Less intuitive interface issues concern the control of the interaction. When does the system listen to the speaker and when should it ignore speech as extraneous? How can the system best signal to the speaker that it is ready to listen? How can a user verify that the system understood the utterance correctly? How does the user correct any recognition errors quickly and efficiently? These and other questions are currently unanswered.

While some researchers have found speech to be the best communication mode in human-human problem solving [1], results from evaluations of computer speech recognizers point in the opposite direction [10, 11, 16, 9], with the exception of a few, contrived, exceptions [14]. The community has become aware that speech applications need more than good recognition to function adequately [13, 4, 8], but no systematic solutions have been offered.

Our objectives in this paper are to clarify some of the tradeoffs involved when users are given the option of using either speech or typing as an input to an application program. We deliberately chose the simplest possible task to avoid confusing task-related cognitive factors with the inherent advantages and disadvantages of the interface modes.

Experimental Procedure

A study was conducted at Carnegie Mellon to contrast the input of numeric data through speech with data entry through a conventional keyboard. The study consisted of two essentially identical experiments and differed only in the method of stimulus presentation. Both experiments required the subjects to enter three lists of 66 digit strings into the computer, using three different data entry modes. In the first experiment, the digit strings were presented on the screen, two lines above the area where either the speech recognition result or the typed input was displayed. In the second experiment, the subjects had to read the digit strings from a sheet of paper placed next to the keyboard and monitor. We will refer to the first experiment as the **screen** experiment and to the second experiment as the **paper** experiment throughout this report.

There were 3 lists of 66 digit strings to be entered. Each data set contained exactly 11 randomly generated digit strings of length 1, 3, 5, 7, 9, and 11. The first six digit strings included one string of each length and were identical for all data sets. These first six digit strings were included for the purpose of familiarizing the subject with a particular condition and were consequently removed from the transcripts before data analysis. Three lists of randomized digit string were generated once at the start of the experiment and used throughout.

Three data entry modes were included in the experiment.

- In the first mode **voice only**, subjects could only use speech to enter a digit string. They read the digit string out loud into a head-mounted, close-talking microphone. The speech recognizer would then analyze the speech signal and display the recognition as a digit string. The subject was asked to verify the result of the recognition. If the result was not correct, the subject was instructed to repeat the digit string into the microphone. This procedure was repeated until the number displayed as the recognition was correct. If the displayed recognition result was correct, the subject would then say the word "OK" or "ENTER". The system running the experiment would then store the number that was entered and the subject could proceed to the next number on his/her list.

- In the **voice with keyboard correction** mode, the subject would again read the digit string into the system. If the recognition was not correct, the subject was instructed to use the keyboard to enter the correct digit string terminated by a carriage return. If the recognition string was correct, or after the keyboard correction was performed, the subject hit the enter key to store the number in the system.
- In the **keyboard only** mode, subjects typed in the digit string, which was then also displayed for confirmation and correction. If they had miskeyed the string they could correct it again using the keyboard. Once the correct digit string was displayed on the screen, subjects would hit the enter key to store the number in the system and proceed to the next number.

Each subject entered the different lists using each of the different input modes (voice only, voice with keyboard correction and keyboard only). Both experiments used replicated 3x3 Greco-Latin square designs [12].

Subjects

Eighteen (18) subjects were recruited at Carnegie Mellon for an experiment in speech recognition. All subjects claimed to be casual typists; examination of typing speeds indicated that this was true, with the exception of one fast touch typist. Nine subjects participated in the first experiment (on-screen presentation of each stimulus) and 9 subjects participated in the second experiment (where the list of digit strings was presented on paper).

Apparatus

Subjects were seated in front of a SUN-3/280 computer workstation with a high-resolution monitor. The operating system was MACH/UNIX. The keyboard for this workstation does not have a numeric keypad and all numbers had to be typed in using keys on the top row of the keyboard. The SPHINX system [6] was used to perform recognition. SPHINX is a large-vocabulary, speaker-independent, continuous speech recognition system developed at Carnegie Mellon. The speech recognition vocabulary consisted of the words ZERO through NINE, OH, ENTER and OK. The grammar allowed either an arbitrary length digit string to be spoken or the words OK or ENTER. When a spoken digit string was recognized, the system displayed the result as a single digit string (with appropriate conversions, i.e. ZERO, OH => 0; ONE => 1; TWO => 2; ... NINE => 9). Typed input was displayed without alteration on the same line as the spoken input.

To minimize variations in system response, the workstation was running a dedicated program to control the experiment. No other processes were running and the system was isolated from the department's network. The program controlling the experiment recorded a log consisting of time stamped inputs and corresponding recognitions. The actual utterances were also captured.

Results

This section presents the results of the experiment, covering recognition system performance and time to completion. All statistical analyses were performed using linear modeling, as implemented in the GLIM system [2]. All statistical comparisons reported at significant at p<0.01 or better.

Accuracy

Overall, typing accuracy was quite high, indicating that subjects performed the experimental task more or less diligently. For the Keyboard mode, digit accuracy was 97.1% for paper presentation and 98.7% for screen presentation. While subjects appear to have been able to type strings more accurately when these were shown on the screen, this difference is not statistically significant.

Recognition word accuracy was significantly higher for screen presentation (95.8%) than for paper presentation (87.6%). It is not clear whether this difference is due to the presentation mode or whether it reflects a sample difference between the two groups of subjects. Given the lack of other evidence, it is not possible to further interpret this difference. It should be also noted that, given the task, these word accuracies are rather low. This can be attributed to the lack of any attempt to tune models for a digit task and the absence of any strategies for dealing with extraneous acoustic events. Were such precautions to be taken, accuracy would be higher.

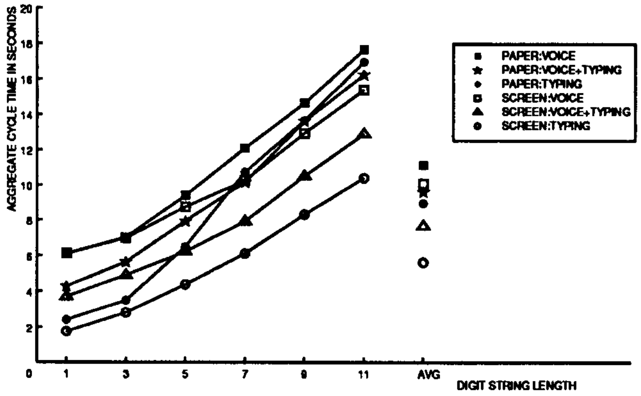
Aggregate cycle time

To determine the efficiency of data entry under the different conditions, we measured the total time a subject needed to enter a number correctly. This aggregate cycle time includes the time elapsed before the subject began speaking after the system had displayed its prompt, the time required to produce the utterance or to type the digit string and (for speech) any system recognition time until the recognition result was displayed. The number computed is the result of adding these times for each initial recognition attempt, each correction attempt and the final confirmation cycle. Thus this time reflects the average time to enter a digit string *correctly*, including all correction and verification time.

Table 1 shows the aggregate completion times for the different combinations of presentation and input mode. Figure 1 shows the aggregate completion times for different string lengths. The paper / screen difference is significant, F(1,3151) = 138, with paper taking longer to complete than screen. Aggregate time to completion for the different input modes is also significantly different, F(2,3151) = 127, as is the interaction between presentation and input mode, F(2,3151) = 3.08.

To better understand the effects of presentation and input mode, we analyzed aggregate cycle time in terms of its component times, factoring out the time for the initial attempt to enter the digit string, the time for the correction cycles and the time necessary for the confirmation.

Figure 1: The aggregate cycle time to input one number correctly for both presentations

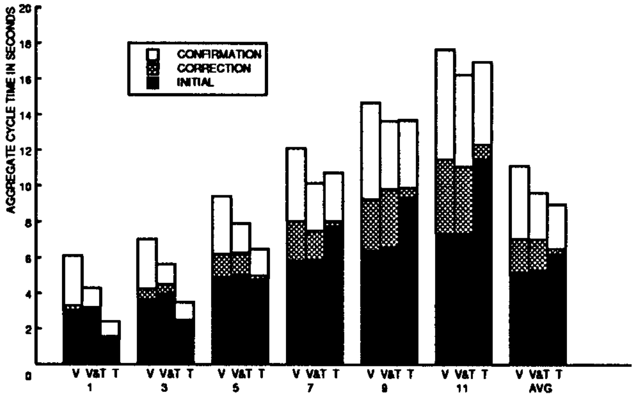


The aggregate cycle time is shown for each input condition (voice, voice with typing and typing) for both experiments. The plot includes each digit string length plotted separately; AVG denotes the overall average for a condition.

Table 1: Mean aggregate cycle times (in seconds)

Modality	paper	screen
Voice	12.5	10.1
Voice + Keyboard	9.6	7.7
Keyboard	8.8	5.6

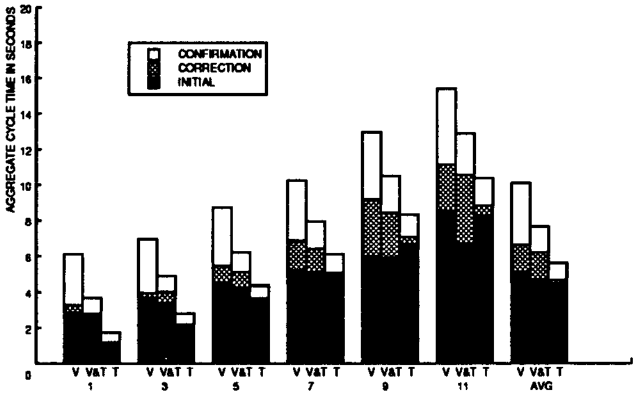
Figure 2: The aggregate cycle time broken down by components for the paper experiment



The components are composed of the initial attempt to enter a string, any corrections and the final confirmation that the string is correct. The modes are abbreviated as V=Voice, V&T=Voice with typing, T=Typing. Each digit string length is plotted separately; AVG denotes the overall averages for a condition.

Correction times are significantly longer for paper presentation, F(1,3151) = 21.9, and there is an interaction between presentation and input mode, F(2,3151) = 8.8. As might be expected, correction time increases significantly with string length, F(5,3151) = 30.0, but there is no interaction with either presentation or input mode. Verification times are also significantly longer for paper presentation, F(1,3151) = 487. Longer strings require longer verification time, though more time is required for this in the paper presentation, F(1,3151) = 80.2. In sum, it would appear that both correction and verification is more difficult with paper presentation, apparently due to the lesser accessibility of the reference materials in that condition.

Figure 3: The aggregate cycle time broken down by components for the screen experiment



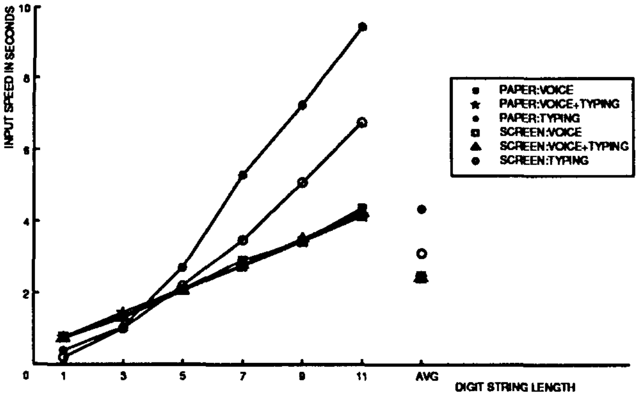
The components are composed of the initial attempt to enter a string, any corrections and the final confirmation that the string is correct. The modes are abbreviated as V=Voice, V&T=Voice with typing, T=Typing. Each digit string length is plotted separately; AVG denotes the overall averages for a condition.

Input Duration time

Since the aggregate cycle times reflect system delays that were much longer for the voice conditions than for the typing condition, we also measured simple input time. That is, the time required by the subjects to type in the digit string or to speak the utterance when they tried to enter a digit string for the first time. This time is a reflection of the actual typing speed or the speech rate, and ignores all influences of reaction time or system processing delays. It also ignores any correction time.

In the paper experiment, we found that speaking the average digit string time took 2.49 seconds in the voice condition, 2.45 seconds in the voice with typing condition while typing time was 4.35 seconds from the first to the last keystroke. In the screen experiment, the voice condition averaged 2.42 seconds from the beginning of the first word to the end of the utterance. The voice with keyboard correction mode required 2.44 seconds of speech, while typing lasted 3.12 seconds. Figure 4 shows the comparison between the typing and speaking rate for both experiments, plotted by string length. The interaction between presentation and length is significant, F(5,3151) = 8.7, as is the interaction between input mode and length, F(10,3151) = 89.9. The time it takes to type a string takes progressively longer than saying it the longer the string. This effect is more pronounced for paper presentation than for screen presentation. In contrast, the time to speak appears to be a linear function of string length.

Figure 4: The raw speech and typing rates for both experiments



The speech or typing rate measures only the time required to say or type the string, excluding all reaction time. The rates are plotted for all 3 conditions in both experiments. Data are plotted by string length as well as overall averages.

Discussion

We should note that the circumstances of this study were biased against speech recognition. One bias was introduced with speech recognition equipment that worked much slower than real time. When subjects had to wait several seconds for a response, their attention wandered and they were more likely to produce utterances that were not task related. We must also assume that their response-time profile is somewhat different, most likely slower than it would be otherwise. In future experiments, a speech interface with better hardware is likely to perform better than in these baseline comparisons.

Another bias came from the use of digits as the basic data unit of the task. Each digit is equivalent to one monosyllabic spoken word (except "seven" and "zero") or one typed character. In most tasks, except those concerned exclusively with alphabets and digits, we find that a monosyllabic word is more equivalent to four or five typed characters. Thus, in other kinds of tasks, the advantage of speech over typing may be more significant because of a greater typing effort involved (see [14]).

Utterance Accuracy

The utterance accuracy results show that speech requires many more interactions to complete the task than typing. This is in part due to the inadequate performance of the speech recognizer involved, which was not well suited to the digit recognition task. A better digit recognition system, properly tuned to this task, has been described by [7]. Speech had a strong disadvantage, especially for longer strings that needed many corrections. Even though it is not novel to assert the need for higher accuracy speech recognition, these numbers provide a reference for comparison with future, higher accuracy spoken language systems.

Aggregate Cycle Time

The basic comparisons in this study involves the time to enter a number correctly, including all corrections and confirmations that are required. This time was measured as wall clock time, which therefore also included system overhead time. System processing time was much longer for the speech conditions. The speech recognizer we used has reported recognition speeds of 1.5 times real time. Several hundred extra milliseconds per utterance were also needed to capture and store the speech signal and for reinitialization of the recognition hardware.

Our results show that speech is almost comparable to typing for the longer digit strings, but typing has a clear advantage for shorter digit strings. The cycle times for the screen experiment were quite a bit faster than those for the paper experiment. This can be attributed to the close proximity of the stimulus and the system display in the same area of the screen. In the paper experiment, the typing condition was slower than speaking, especially for longer digit strings. We attribute this effect to the need to look at the digit string on the paper and then looking back at the keyboard and monitor to type it in. The longer strings require more alternations of looking at the string and typing a part of it, then looking again, etc. Reading the strings was conceptually simpler. There was no need to change the eye position until the complete final result was displayed, which only occurred once after the complete digit string was read.

Considering the components of the aggregate cycle time in Figures 2 and 3, we find relatively fast initial entry times for voice, comparable to or better than the equivalent time for typing. The voice mode loses the race due to correction time. Typing accuracy avoids almost any correction, and speech loses most of its ground. In the confirmation transaction, typing is again very fast, but speech is about a constant amount slower. For paper presentation, confirmation times also increase with string length, indicating the extra effort involved to verify long strings. One lesson that becomes clear from these data is the need to obtain better accuracy and response time for speech input. We especially need to have faster correction mechanisms, and ideally, a better system would totally avoid the need for multiple corrections in the voice-only conditions.

Effective speech interface design requires that it be possible to correct or bypass the speech modality. The effectiveness of this is shown in the improved throughput observed for the voice + keyboard condition. More generally, appropriate error-correction facilities need to be provided.

Input Duration Time

The input duration times measure the typing speed and the speech rate. These times give a lower bound on what can be done by casual users. Note, however, that these times were obtained using a standard keyboard, not numeric keypads and that they are not characteristic for expert touch typists.

Speech input is fast. This is evident if we compare average speech rate, which is estimated at about 200 words per minute with typing; even good typists cannot normally achieve this rate of input [5]. Our data confirm these findings.

The input duration times in our experiments also show that real time response and accurate speech recognition are essential if a clear advantage is to be shown for speech. The average difference between pronouncing a digit string and typing one was less than 2 seconds in both experiments. Thus, if the speech recognizer has more than a 2 second delay or if the recognizer has a significant error rate (as it did in our experiments) or the interface introduces other artificial delays, speech would cease to be a desirable communication mode.

The results showed that the raw typing rate in the paper experiment was much slower than typing rate for the screen experiment. This difference can only be attributed to the extra load imposed on the users when they divide their attention between the keyboard, the screen and the paper containing the data to be entered. If a task has these characteristics, sometimes more vaguely described as 'eyes-busy', then speech would be a preferable input channel for data entry. In our experiment even a relatively small increase in the work load for the eyes substantially changed the performance in the typing rate. Other, more demanding tasks can be expected to degrade performance in the typing mode even more.

Summary

In this study we have examined how speech compares with typing for a digit entry task. We found that properties of the input, such as string length affect the relative advantages of each modality. System response cgharacteristics, however, ultimately dominate throughput. Based on our data, we believe that more complex materials, requiring more keystrokes per syllable, would demonstrate the superiority of speech.

Depending on the task, as demonstrated by our comparisons of screen vs paper presentation, speech can have tremendous advantages for casual users. The paper task required a certain visual effort, because the subject was glancing back and forth between the paper containing the input data, the keyboard and the screen result. The more a task requires visual monitoring of input (or most other kinds of cognitive distractions), the more preferable speech will become as an input medium. Of course, the vocabulary of the task must lie within the range of the speech recognizers that are available.

Screen presentation demonstrates that speech can provide an advantage despite adverse circumstances. Even when the subject has all relevant task information present in a small visual area of the screen, speech still helps out by eliminating the time spent locating keys on the keyboard. Speech allows the user achieve a cleaner separation of modalities and allows data input functions to be localized in a single channel, thus eliminating the interference produced by having to share the visual channel.

In tasks that require no visual monitoring, have very short words (e.g., digits) or when using skilled typists, speech will probably not demonstrate an advantage. This is particularly true when data is entered from specific, customized devices such as a numeric keypad or a specialized typewriter. We do not feel, however, that such situations are typical of the environments in which the availablility speech input will have its greatest impact.

The key to building improved spoken language applications lies in better speech recognition speed and accuracy, as well as effective strategies for correcting errors and confirming correct recognitions. Improving recognition accuracy and speed lies in the domain of chip designers and speech researchers. The challenge to spoken language interface builders is to find effective strategies for managing a communication channel that is prone to errors and requires ongoing validation of inputs.

References

1. Chapanis, A. Interactive Human Communication: Some lessons learned from laboratory experiments. In Shackel, B., Ed., *Man-Computer Interaction: Human Factors Aspects of Computers and People*, Sijthoff and Noordhoff, Rockville, Md, 1981, pp. 65-114.

2. Healy, M.J.R. *GLIM: An introduction.* Oxford University Press, New York, 1988.

3. Grossberg, M. and Wiesen, R.A. and Yntema, D.B. "An experiment on problem solving with delayed computer responses.". *IEEE Transactions on Systems, Man and Cybernetics SMC-6*, 3 (March 1976), 219-222.

4. Holmgren, J.E. "Toward Bell System Applications of Automatic Speech Recognition". *Bell System Technical Journal 62*, 6 (July - August 1983), 1865 - 1880.

5. Jusczyk, P. Speech Perception. In *Handbook of Perception and Human Performance*, Boff, K.R., Kaufman, L. and Thomas, J.P., Eds., Wiley, New York, 1986.

6. Lee, K.-F. *Automatic Speech Recognition: The Development of the SPHINX System.* Kluwer Academic Publishers, Boston, 1989.

7. Lee,C.-H., Juang,B.-H., Soong,F.K.. and Rabiner, L.R. Word recognition using whole word and subword models. ICASSP-89, Proceedings of the International Conference on Acoustics, Speech and Signal Processing, Glasgow, Scotland, May, 1989, pp. 683 - 686.

8. Leggett, J. and Williams, G. "An empirical investigation of voice as an input modality for computer programming". *International Journal of Man-Machine Studies 21*, 6 (December 1984), 493 - 520.

9. Martin, G.L. "The utility of speech input in user-computer interfaces". *International Journal of Man-Machine Studies 29* (1889), 355-376.

10. Martin, T.B. and Welch, J.R. Practical speech recognizers and some performance effectiveness parameters. In Lea, W.A., Ed., *Trends in speech recognition*, Prentice-Hall, Englewood Cliffs, 1980, pp. 24-38.

11. Morrison, D.L., Green, T.R.G., Shaw, A.C. and Payne, S.J. "Speech Controlled Text-editing: effects of input modality and of command structure". *International Journal of Man-Machine Studies 21*, 1 (June 1984), 49 - 64.

12. Myers, J.L. *Fundamentals of Experimental Design.* Allyn and Bacon, Boston, MA, 1972.

13. Nye, J.M. "Human Factors Analysis of Speech Recognition Systems". *Speech Technology 1*, 2 (April 1982), 50 - 57.

14. Poock, G.K. "Voice Recognition boosts Command Terminal Throughput". *Speech Technology 1*, 2 (April 1982), 36 - 39.

15. Rudnicky, A.I. System response delay and user strategy selection in a spreadsheet task. , April, 1990. CHI'90, invited poster.

16. Rudnicky, A.I., M. H. Sakamoto and J. H. Polifroni. "Spoken language interaction in a goal-directed task". *Proceedings of the ICASSP* (April 1990), 45-48.