S2.2

SPOKEN LANGUAGE INTERACTION IN A GOAL-DIRECTED TASK

Alexander I. Rudnicky Michelle Sakamoto Joseph H. Polifroni

School of Computer Science Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

ABSTRACT

To study the spoken language interface in the context of a complex problem-solving task, a group of users were asked to perform a spreadsheet task, alternating voice and keyboard input. A total of 40 tasks were performed by each participant, the first thirty in a group (over several days), the remaining ones a month later. The voice spreadsheet program used in this study was extensively instrumented to provide detailed information about the components of the interaction. These data, as well as analysis of the participants's utterances and recognizer output, provide a fairly detailed picture of spoken language interaction.

Although task completion by voice took longer than by keyboard, analysis shows that users would be able to perform the spreadsheet task *faster* by voice, if two key criteria could be met: recognition occurs in real-time, and the error rate is sufficiently low. This initial experience with a spoken language system also allows us to identify several metrics, beyond those traditionally associated with speech recognition, that can be used to characterize system performance.

INTRODUCTION

The ability to communicate by speech is known to enhance the quality of communication, as reflected in shorter problem-solving times and general user satisfaction [1]. Recent advances in speech recognition technology [2] have made it possible to build "spoken language" systems that create the opportunity for interacting naturally with computers. Spoken language systems combine a number of desirable properties. Recognition of *continuous speech* allows users to use a natural speech style. *Speaker independence* allows casual users to easily use the system and eliminates training as well as its associated problems (such as drift). *Large vocabularies* make it possible to create habitable languages for complex applications. Finally, a *natural language* processing capability allows the user to express him or herself using familiar locutions.

While the recognition technology base that makes spoken language systems possible is rapidly maturing, there is no corresponding understanding of how such systems should be designed or what capabilities users will expect to have available. It is intuitively apparent that speech will be suited for some functions (e.g., data entry) but unsuited for others (e.g., drawing). We would also expect that users will be willing to tolerate some level of recognition error, but do not know what this is or how it would be affected by the nature of the task being performed or by the error recovery facilities provided by the system.

Meaningful exploration of such issues is difficult without some baseline understanding of how humans interact with a spoken language system. To provide such a baseline, we implemented a spoken language system using currently available technology and used it to study humans performing a series of simple tasks. We chose to work with a spreadsheet program since the spreadsheet supports a wide range of activities, from simple data entry to complex problem solving. It is also a widely used program, with a large experienced user population to draw on. We chose to examine performance over an extended series of tasks because we believe that regular use will be characteristic of spoken language applications.

THE VOICE SPREADSHEET SYSTEM

The voice spreadsheet (henceforth "VSC") consists of the UNIX-based spreadsheet program SC interfaced to a recognizer embodying the SPHINX technology described in [2]. Additional description of VSC is available elsewhere [4], as is a description of the spreadsheet language [6].

VSC is a speaker-independent continuous speech recognition system. To train the phonetic models used in the recognizer, we combined several different databases, all recorded at Carnegie Mellon using the same microphone as used for the spreadsheet study (a close-talking Sennheiser HMD-414). The training speech consisted of: calculator sentences (1997 utterances), a (general) spreadsheet database (1819 utterances), and a task-specific database for financial data (196 utterances). A total of 4012 utterances was thus included in the training set. Table 1 provides some performance data that characterize system performance.

Table 1: Comparison of recognizer performance for live and read speech

Test Set	utts	words	utterances correct	words accuracy
Reference (read)	99	491	72.7	93.7
Live (complete)	406	1486	78.9	92.7
Live (clean)	366	1389	85.5	94.9
Live (read clean)	366	1389	82.8	94.0

The basic recognition performance ("Reference"), as tested on speech collected at the same time as the training data, is about what might be expected given the known performance characteristics of the SPHINX system (specifically, 94% word accuracy for the perplexity 60 version of the Resource Management task).

The Table also presents recognition performance for speech collected in the user study described below ("Live Session"). The "complete" version shows system performance over 4 sessions representing 4 different talkers and chosen from about the mid-point of the initial 30 task series (as detailed below). Note that this set includes utterances that contain various spontaneous speech phenomena that cannot be handled correctly by the current system. The "clean speech" set includes only those utterances that both contain no interjected material (e.g., audible non-speech) and that are grammatical. Performance on this set is quite good, and there is no evidence that mere "spontaneity" leads to poorer recognition performance. We can verify this equivalence more concretely by comparing read and spontaneous speech produced by the same talkers. To do this, we asked the four participants whose speech comprised the spontaneous test sets to return and record read versions of their spontaneous utterances, using scripts taken from our transcriptions. As can be seen in the Table, performance is comparable for read and live speech[1].

[1]The slightly better performance with Live speech might seem counter-intuitive. Examination of specific errors in the Read version indicates that one of the speakers read her material at a distinctly slower pace than she spoke it spontaneously (we estimate 34% slower). The bulk of the excess errors can be accounted for by this interpretation. For example, many of the errors are *splits*, characteristic of slow speech.

CH2847-2/90/0000-0045 $1.00 © 1990 IEEE

Given that this pattern of results can be shown to generalize to other tasks (and there is no reason to believe that they would not), the implications of this experiment are highly significant: A system trained on read speech will not substantially degrade in accuracy when presented with spontaneous speech provided that certain other characteristics, such as speech rate, are comparable. Note that this only applies to those utterances that are comparable to read speech insofar as they are grammatical and contain no extraneous acoustic events. The system will still need to deal with these phenomena. If these problems can be solved in a satisfactory manner, then we can comfortably expect spontaneous spoken language system performance to be comparable to system performance evaluated on read speech.

A STUDY OF SPOKEN LANGUAGE SYSTEM USAGE

To understand how users approach a voice-driven system and how they develop strategies for dealing with this type of interface, we had a group of users perform a series of more or less comparable task over an extended period of time and monitored various aspects of system and user performance over this period.

METHOD

The task chosen for this study was the entry of personal financial data from written descriptions of various items in a fictitious individual's monthly finances. An attempt was made to make each version of the task comparable in the amount of information it contained and in the variety of arithmetic operations required. On the average, each task required entering 38 pieces of financial information, an average of 6 of these entries required arithmetic operations such as addition and multiplication. Movement within the worksheet, although generally following a top to bottom order, skipped around, forcing the user to make arbitrary movements, including off-screen movements. Users were presented with preformatted worksheets containing appropriate headings for each of the items they would have to enter. In addition, each relevant cell location was given a label that would allow the user to access it using symbolic movement instructions (as defined in [6]).

The information to be entered was presented on separate sheets of paper, one entry to a sheet, contained in a binder positioned to the side of the workstation. This was done to insure that all users dealt with the information in a sequential manner and would follow a predetermined movement sequence within the worksheet. To aid the user, the bottom of each sheet gave the category heading for the information to be entered and, if existing, a symbolic label for the cell into which the information was to be entered.

PROCEDURE AND DESIGN. All participants performed 40 tasks. The first 30 tasks were completed in a block, over several days. The last ten were completed after an interval of about one month. The purpose of the latter was to determine the extent to which users remembered their initial extended experience with the voice spreadsheet and to what degree this retest would reflect the performance gains realized over the course of the original block of sessions. Since we were interested in studying a spoken language system in an environment that realistically reflects the settings in which such a system might eventually be used, we made no special attempt to locate the experiment in a benign environment or to control the existing one. The workstation was located in an open laboratory and was not surrounded by any special enclosure.

At the beginning of each session, each participant was given a standard-format typing test to determine their facility with the keyboard. The typing test revealed two categories of participant, touch typists (3 people) with a mean typing rate of 63 words per minute (wpm) and "hunt and peck" typists (5 people), with a mean typing rate of 31 wpm. Task modality (whether speech or typing) alternated over the course of the experiment, each successive task being carried out in a different modality. To control for order and task-version effects the initial modality and the sequence of tasks (first-to-last *vs* last-to-first) was varied to produce all possible combinations (four). Two people were assigned to each combination.

The participants were informally solicited from the university community through personal contact and bulletin board announcements. There were 3 women and 5 men, ranging in age from 18 to 26 (mean of 22). With the exception of one person who was of English/Korean origin, all participants were native speakers of English. All had previous experience with spreadsheets, an average of 2.3 years (range 0.75 to 5), though current usage ranged from daily to "several times a year". None of the participants reported any previous experience with speech recognition systems (though one had previously seen a SPHINX demonstration).

RESULTS

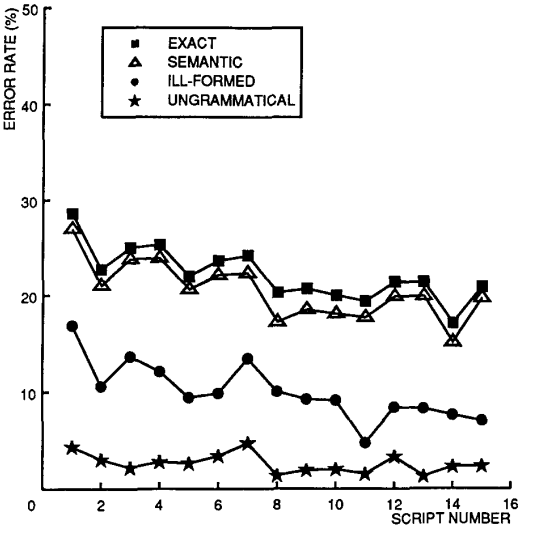
The data collected in this study consisted of detailed timings of the various stages of interaction as well as the actual speech uttered over the course of system interaction.

RECOGNITION PERFORMANCE AND LANGUAGE HABITABILITY

To analyze recognizer performance we captured and stored each utterance spoken as well as the corresponding recognition string produced by the system. All utterances were listened to and an exact lexical transcription produced. The transcription conventions are described more fully in [5], but suffice it to note that in addition to task-relevant speech, we coded a variety of spontaneous speech phenomena, including speech and non-speech interjections, as well as interrupted words and similar phenomena.

The analyses reported here are based on a total of 12507 recorded and transcribed utterances, comprising 43901 tokens. We can use these data to answer a variety of questions about speech produced in a complex problem-solving environment. Recognition performance data are presented in Figure 1. The values plotted represent the error rate averaged across all eight subjects.

Figure 1: Mean utterance accuracy across tasks



The top line in Figure 1 shows ***exact*** utterance accuracy, calculated over all utterances in the corpus, including system firings for extraneous noise and abandoned (i.e., user interrupted) utterances. It does not include begin-end detector failures (which produce a zero-length utterance), of which there were on the average 10% per session. Exact accuracy corresponds to utterance accuracy as conventionally reported for speech recognition systems using the NBS scoring algorithm [3]. The general trend of recognition performance over time is improvement, though the improvement appears to be fairly gradual. The improvement indicates that users are sufficiently aware of what might improve system performance to modify their behavior accordingly. On the other hand, the amount of control they have over it appears to be very limited.

The next line down shows ***semantic*** accuracy, calculated by determining, for each utterance, no matter what its content, whether the correct action was taken by the system[2]. Semantic accuracy, relative to exact accuracy, represents the added performance that can be realized by the parsing and understanding components of an SLS. In the present case, the added performance results from the 'silent' influence of the word-pair grammar which is part of the recognizer. Thus, grammatical constraints are enforced not through, say, explicit identification and reanalysis of out-of-language utterances, but implicitly, through the word-pair grammar. The *spread* between semantic and exact accuracy defines the contribution of higher-level process and is a parameter that can be used to track the performance of "higher-level" components of a spoken language system.

The line at the bottom of the graph shows ***grammaticality*** error. Grammaticality is determined by first eliminating all non-speech events from the *transcribed* corpus then passing these filtered utterances through the parsing component of the spreadsheet system. Grammaticality provides a dynamic measure of the *coverage* provided by the system task language (on the assumption that the user's task language evolves with experience) and is one indicator of whether the language is sufficient for carrying out the task in question.

The grammaticality function can be used to track a number of system attributes. For example, its value over the period that covers the user's initial experience with a system indicate the degree to which the implemented language covers utterances produced by the inexperienced user and provides one measure of how successfully the system designers have anticipated the speech language that users intuitively select for the task. Examined over time, the grammaticality function indicates the speed with which users modify their speech language for the task to reflect the constraints imposed by the implementation and how well they manage to stay within it. Measurement of grammaticality after some time away from the system indicates how well the task language can be retained and is an indication of its appropriateness for the task. We believe that grammaticality is an important component of a composite metric for the *language habitability* of an SLS and can provide a meaningful basis for comparing different SLS interfaces to a particular application[3].

Examining the curves for the present system we find, unsurprisingly, that VSC is rather primitive in its ability to compensate for poor recognition performance, as evidenced by how close the semantic accuracy line is to the exact accuracy line. On the other hand, it appears to cover user language quite well, with only an average of 2.9% grammaticality error[4]. In all likelihood, this indicates that users found it quite easy to stay within the confines of the task, which in turn may not be surprising given its simplicity.

SPONTANEOUS SPEECH PHENOMENA. When a spoken language system is exposed to speech generated in a natural setting a variety of acoustic events appear that contribute to performance degradation. Spontaneous speech events can be placed into one of three categories: ***lexical***, ***extra-lexical***, and ***non-lexical***, depending on whether the item is part of the system lexicon, a recognizable word that is not part of the lexicon, or some other event, such as a breath noise. These categories, as well as the procedure for their transcription, are described in greater detail in [5]. Table 2 lists the most common non-lexical events encountered in our corpus. The number of events is given, as well as their incidence in terms of words in the corpus. Given the nature of the task, it is not surprising to find, for example, that a large number of paper rustles intrudes into the speech stream. Non-lexical events were transcribed in 893 of the 12507 utterances used for this analysis (7.14% of all utterances).

The ***ill-formed*** curve in Figure 1 shows the proportion of transcribed utterances that contain extraneous material (such as the items in Table 2). This function was generated by calculating grammaticality with both non-lexical and extra-lexical tokens included in the transcription. As is apparent, the incidence of extraneous events steadily decreases over sessions.

[2]For example, the user might say "LET'S GO DOWN FIVE", which lies outside the system language. Nevertheless, because of grammatical constraints, the system might force this utterance into "DOWN FIVE", which happens to be grammatically acceptable and which also happens to carry out the desired action. From the task point of view, this recognition is correct; from the recognition point of view it is, of course, wrong.

Table 2: Incidence of (some) non-lexical spontaneous speech tokens.

585	++RUSTLE+	4	++PHONE-RING+
206	++BREATH+	4	++NOISE+
43	++MUMBLE+	4	++DOOR-SLAM+
18	++SNIFF+	4	++CLEARING-THROAT+
13	++BACKGROUND-NOISE+	4	++BACKGROUND-VOICES+
11	++MOUTH-NOISE+	2	++SNEEZE+
10	++COUGH+	1	++SIGH+
6	++YAWN+	1	++PING+
5	++GIGGLE+	1	++BACKGROUND-LAUGH+

Note: The first column given the percentage and the second column the actual number of tokens for the given non-lexical token. There are 43,901 tokens in the corpus.

Users apparently realize the harmful effects of such events and work to eliminate them (conversely, the user does not appear to have absolute control over such events, otherwise the decrease would have been much steeper).

While existing statistical modeling techniques can be used to deal with the most common events (such as paper rustles) in a satisfactory manner (as shown by [7]), more general techniques will need to be developed to account for low-frequency or otherwise unexpected events. A spoken language system should be capable of accurately identifying novel events and dispose of them in appropriate ways.

THE TIME IT TAKES TO DO THINGS

Of particular interest in the evaluation of a speech interface is the potential advantages that speech offers over alternate input modalities, in particular the keyboard. On the simplest terms, a demonstration that a given modality provides a time advantage is a strong *a priori* argument that this modality is more desirable than another.

The total time it takes to perform a task is a good indication of how effectively it can be carried out in a particular fashion. Figure 2 shows the mean total time it took users to perform the spreadsheet tasks. As can be seen, keyboard entry is faster. Moreover, the time taken to perform a task by keyboard improves steadily over time. The comparable speech time, while improving for a time, seems to asymptote a level above that of keyboard input. Since the tasks being performed are essentially (and over individuals, exactly) the same, we must infer that the lack of improvement is due in some fashion to the nature of the speech interface.

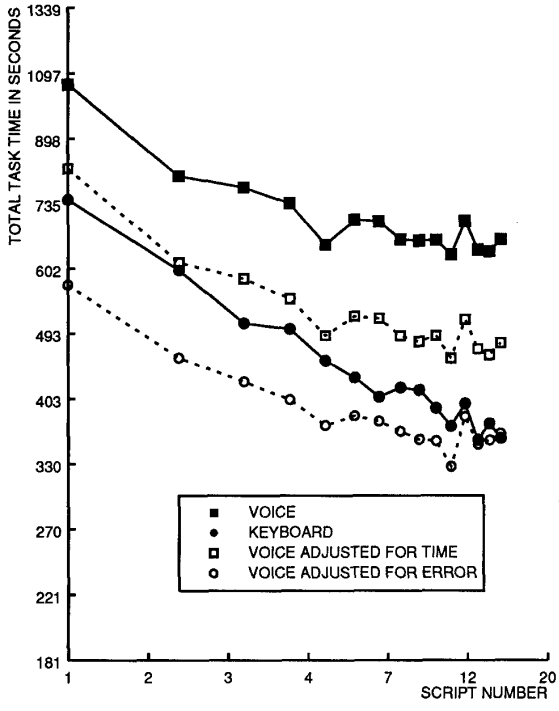
The reasons for this become clearer if we examine in greater detail where the time goes. The present implementation incurs substantial amounts of system overhead that at least in principle could be eliminated through suitable modifications. Currently, sizable delays are introduced by the need to initialize the recognizer (about 200 ms), to log experimental data (about 600 ms), and by the two times real-time performance of the recognizer. What would happen if we eliminate this overhead?

If we replot the data by subtracting these times, but retaining the time taken to speak an utterance, we find that the difference between speech and keyboard is reduced, though not eliminated (see Figure 2). This result underlines the probable importance of designing tightly-coupled spoken language systems for which the excess time necessary for entering information by speech has been reduced to a value comparable to that found for keyboard input. In a personal workstation environment this would essentially have to be nil, and we believe represents a *minimum* requirement for successful speech-based applications that support goal-directed behavior.

[3]*System habitability*, on the other hand, has to be based on a combination of language habitability, robustness with respect to spontaneous speech phenomena, and system responsiveness.

[4]Bear in mind that this percentage includes *intentional agrammaticality* with respect to the task, such as expressions of annoyance or interaction with other humans.

Figure 2: Total task completion time



There is an additional penalty imposed on speech in the current system—recognition error. In terms of the task, the only valid inputs are those for which the utterance is correctly recognized. If an input is incorrect, it has to be repeated. We can get an idea of how fast the task could actually be performed if we discount the total task time by the error rate. That is, if a task is presently carried out in 10 min, but exhibits a 25% utterance error, then the task could actually have been carried in 7.5 min, had we been using a system capable of providing 100% utterance recognition. Figure 2 compares total task time corrected by this procedure. If we do this, we find that the amount of time taken to carry out the task by voice is actually faster than by keyboard.

Finally, we can ask what level of recognition performance is necessary for speech to equal keyboard input. Given that the mean task time over 15 sessions for keyboard is 448 ms and that the mean task time for the "real-time" adjustment is 528 ms, then we can estimate that a 15% error rate (a halving of the current rate) will produce equivalent task completion times for speech and keyboard. We believe that this goal is achievable in the near term.

The above speculations are, of course, exercises in arithmetic and cannot take the place of an actual demonstration. We are currently working towards the goals of creating a true real-time implementation of our system and on improving system accuracy.

DISCUSSION

The results obtained in this study provide a valuable insight into the potential advantages of spoken languages systems and allow us to identify those aspects of system design whose improvement is critical to the usability of such systems. Furthermore, this study lays out a framework for the evaluation of SLS performance, identifying a number of useful diagnostic metrics.

Although we found that total task time was greater for speech input than for keyboard, this was not due to any intrinsic deficit for voice input. In fact, if we examine the component actions performed by the user, we find that they could be completed faster by voice than by typing. The failure of the speech mode to achieve greater throughput can be attributed to two shortcomings of our spoken language system.

A *time penalty* is imposed by our current implementation, which processes speech at about 2 times real-time and incorporates a substantial overhead. The penalty is reflected not only in longer task times, but also in changes to user strategies. Fortunately, real-time performance can be achieved with a suitable implementation and sufficient hardware resources. We are currently reimplementing our system on a multi-processor computer and expect to achieve sub-real-time performance in the near future.

While speed is a tractable problem, *low accuracy* is less so. We can expect to improve utterance recognition on the order of 10% if we properly model extraneous events, but even if we do so, recognition performance may still be at a level that significantly interferes with task performance. Judging from Figure 2, it may be sufficient to provide a moderate improvement in recognition accuracy, which together with real-time recognition would be sufficient to allow a spoken language system to perform at a level equivalent to a keyboard system.

The metrics presented above can be used to describe system performance in ways that are useful for understanding the characteristics of a particular spoken language system. As such, they would be of limited interest to those not directly involved in spoken language research. In a larger arena, SLSs will be competing with other interface technologies and the bases for comparison will be universally applicable metrics, such as task completion time and ease of use. The challenge is to build systems that can compete successfully on those terms.

ACKNOWLEDGMENTS

A number of people have contributed to the work described in this paper. We would like to thank Robert Brennan who did the initial implementation of the voice spreadsheet program and Takima Hoy who produced the bulk of the transcriptions used in our performance analyses.

The research described in this paper was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 5167, monitored by SPAWAR under contract N00039-85-C-0163. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

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

3. Pallett, D.S. Benchmark tests for DARPA Resource management database performance evaluations. In *Proceedings of ICASSP*, IEEE, 1989, pp. 536-539.

4. Rudnicky, A.I. The design of voice-driven interfaces. In *Proceedings of the February DARPA Speech and Natural Language Workshop*, Morgan Kaufman, 1989, pp. 120-124.

5. Rudnicky, A.I. and Sakamoto, M.H. Transcription conventions for spoken language research. Tech. Rept. CMU-CS-89-194, Carnegie Mellon University School of Computer Science, October, 1989.

6. Rudnicky, A.I., Polifroni, J.H., Thayer, E.H., and Brennan, R.A. "Interactive problem solving with speech". *Journal of the Acoustical Society of America 84* (1988), S213(A).

7. Ward, W.H. Modelling Non-Verbal Sounds for Speech Recognition. In *Proceedings of the October DARPA Speech and Natural Language Workshop*, Morgan Kaufman, 1989.