

Kansas State University Libraries

New Prairie Press

Conference on Applied Statistics in Agriculture

1996 - 8th Annual Conference Proceedings

DESIGNING SPEECH INTERFACE APPLICATIONS FOR ACQUISITION OF AGRICULTURAL INFORMATION

Jeffrey Willers

Susan Bridges

Xiaofeng Ma

James McKinion

Jean Liang

See next page for additional authors

Follow this and additional works at: <https://newprairiepress.org/agstatconference>



Part of the [Agriculture Commons](#), and the [Applied Statistics Commons](#)



This work is licensed under a [Creative Commons Attribution-Noncommercial-No Derivative Works 4.0 License](#).

Recommended Citation

Willers, Jeffrey; Bridges, Susan; Ma, Xiaofeng; McKinion, James; and Liang, Jean (1996). "DESIGNING SPEECH INTERFACE APPLICATIONS FOR ACQUISITION OF AGRICULTURAL INFORMATION," *Conference on Applied Statistics in Agriculture*. <https://doi.org/10.4148/2475-7772.1324>

This is brought to you for free and open access by the Conferences at New Prairie Press. It has been accepted for inclusion in Conference on Applied Statistics in Agriculture by an authorized administrator of New Prairie Press. For more information, please contact cads@k-state.edu.

Author Information

Jeffrey Willers, Susan Bridges, Xiaofeng Ma, James McKinion, and Jean Liang

DESIGNING SPEECH INTERFACE APPLICATIONS FOR ACQUISITION OF AGRICULTURAL INFORMATION

Jeffrey Willers¹, Susan Bridges², Xiaofeng Ma², James McKinion¹, and Jean Liang²

¹ USDA-ARS Crop Simulation Research Unit, Mississippi State

² Department of Computer Science, Mississippi State

Abstract

It will be argued that customary software design strategies, by themselves, fall short when designing speech recognition applications. Concepts of experimental design and analysis are also necessary for developing speech interface software. This study demonstrates that these tools can be advantageous to the software developer, especially if the prototype methodology model of software development is applied. A case study for the problem of developing a speech interface for collecting, or mapping, information on cotton plant growth is presented. The acquisition of cotton plant map data is a 'hands and eyes' busy task that requires considerable investment to record and convert hand-written data sheets into computer data files. The project goal is to develop software that converts spoken key words and phrases describing a cotton plant into text 'strings' that are subsequently manipulated into a computer ready data file.

Keywords: speech recognition, software testing, software design, experimental design.

1. Introduction

For many agricultural applications, it is necessary to collect data on the status of the crop so that alterations in management practices can occur. Much of this data is never recorded, and only a verbal report or short written summary is presented to the farm manager. On other occasions, however, these data are recorded on paper. Later, it may be necessary to convert these data from a paper format into a computer file in preparation for additional analyses. This change in format, from observation to paper to data file, can be time consuming, laborious, and at risk to coding or typographical errors. Recent advances in computer technology, involving both hardware and software, place many agricultural data collection tasks at the threshold of new opportunities. In particular, progress in speech recognition capabilities is making tremendous advances.

Presently, the intense information requirements of many computer-based systems is one obstacle to their wider use. For example, during the validation of the cotton insect management expert system, CIC-EM (Cotton Insect Consultant for Enhanced Management) (Bowden et al., 1992), one evaluator remarked that by the "... time I enter all the data, and wait for a response from the system, I could have already made a decision". While correct in the short run, this evaluator overlooked the potential of having access to a larger data base which could help him make a better decision. Similar experiences are a hindrance to many other software systems being built for cotton management (Lemmon, 1986; Olson et al., 1992). The problem of information management is not unique to cotton production. The "information acquisition

bottleneck” problem prevents many forms of agricultural data from being recorded, archived, or used for analyses.

Gauch (1993) has recently commented that before the advent of computers analysis, not data collection, was the more expensive endeavor of the two. But today, the expense of collecting data exceeds the cost of analysis. The implications of reversing this trend are profound. For example, on any tract of cropland, data are collected on several attributes several times a year; often, for many years. The potential value of capturing this data and using it to develop better management practices, especially for insects and other pests (Pedigo and Buntin, 1993; Willers et al., 1992) is immense. Meta-analysis (Hedges and Olkin, 1985) and other traditional analysis methods can be applied to data obtained from the same farm for several years or from several farms or regions to uncover important agroecosystem relationships to meet increased demands for food and fiber.

Currently, much of this data is lost forever simply because the time to capture and archive these data is too great. For example, one of us (JLW) was fortunate to obtain the insect scouting records and recommendations of one cotton farm during the 1995 production year. This set of data, numbering more than 500 pages, represents the time trend for several important insect pests of cotton in over 300 fields. If this data were available every year, along with each field’s management history, one could begin to glean trends to better manage crop pests. Unfortunately, additional records from other production years to complete such an analysis are unavailable. The information from prior years is already lost-the cost of storage, or conversion into computer files, was too high.

Speech recognition technology offers a potential solution to the data acquisition bottleneck. Speech enabled interfaces would allow the direct capture of observed data pertinent to “on farm” tasks in electronic form.

2. Materials and Methods

The ability of computers to recognize speech, while becoming advanced in capability, is still limited under today’s technology. Voice recognition software cannot guarantee that no errors will occur during the speech recognition process. While the accuracy of these applications can approach 95% or better, the designer of a speech interface should be aware of speech engine limitations. Therefore, a careful choice of an appropriate application for developing a speech enabled interface should be made. Specifically, with the engine used here, the vocabulary needed to complete the task should consist of fewer than 1,000 words (Anonymous, 1995). Even with careful consideration of the problem domain, the designer should still chose a vocabulary that avoids similar utterances or words that have similar sounds (e.g., “computer” and “commuter” (Anonymous, 1995)). Also, the speech enabled application should provide effective error recovery mechanisms. To explore the impact of these types of considerations, several experiments were developed. A brief history of the project follows.

Project Description. A speech interface for one portion of a crop management system was initially built using the IBM™ *Continuous Speech Series* (ICSS) speech recognition toolkit. The project continues its development under an ICSS upgrade named *Voice-Type Application Factory for Windows* (VTAF) (Anonymous, 1995). This toolkit allows the interface designer to

create a Windows 3.1™ environment interface tailored to the application using a programming language like Microsoft™ Visual C++ (ver. 2.0).

The purpose of the project is to explore the utility of speech input for time-consuming data entry tasks such as cotton plant mapping. The emphasis in this article is the need to utilize statistical concepts to better design speech recognition interfaces. Many issues related to usability were considered in the design and implementation of this speech interface, including learnability, efficiency, error prevention and detection, and the user's subjective sense of comfort. The software development procedure used was the 'prototype methodology' (Jalote, 1991) process. This procedure uses a sequence of intermediate prototypes to resolve many design issues.

A more detailed description of our initial application (called COTTON TALK) can be found in Liang et al. (1996). The development of COTTON TALK is motivated by a desire to improve the efficiency and speed of acquisition of plant growth data deemed necessary for optimal cotton crop management. Plant maps are used to monitor plant growth and the retention of buds at flowering sites. If growth or fruit retention are less than optimal various agronomic practices might be employed to correct the situation. One estimate, provided by a cotton producer is that it costs \$4 per acre per season to acquire plant mapping data. For this farm, where the total cotton acreage exceeds 8,000 acres, there exists considerable potential for cost savings if speech interfaces can be employed. Therefore the goal of the project is to build a speech interface that allows the user to enter plant mapping data both correctly and conveniently.

Presently, the system only provides a basic set of features for cotton plant mapping. The domain of application was restricted to help control the scope and size of the project. Our experience with this basic system has encouraged us to continue and broaden the functionality of the program for public use and to plan the development of other speech enabled applications.

Only a few of many experiments that helped guide interface development have been selected for this article. The results of several earlier experiments can be found in Liang (1995).

To assist the reader in understanding more about the nature of the experiments described below, several definitions relevant to the use of VTAF are provided. First, an *utterance* is a period of time where sound exceeds threshold values (Anonymous, 1995). A continuous period when the sound level does not exceed the threshold value is defined as *silence*. An input session ends whenever any period of time has a sound level below a *stop* threshold value and is longer than the *end silence period*. The end silence period is a control parameter that is used by VTAF to determine when an utterance has ended and the recognition process can continue to completion. Both *start* and *stop* thresholds define the beginning and ending of an utterance. It is important to understand that different thresholds are appropriate for different environments and different types of microphones. The VTAF documentation should be consulted for more details (Anonymous, 1995).

Description of Experiments. In the first experiment, two methods entering numeric data were compared. One method used the cardinal representation of the number (e.g., saying "eighty-five" for the number 85), whereas the second method used a digit-by-digit representation (e.g., saying "eight-five" for the number 85). The value zero is understood by both methods as either 'zero' or 'oh'. For example, with the digit-by-digit method a value like 'twenty' can be entered by saying 'two zero' or 'two oh'.

The expected result for the first experiment is that the digit-by-digit method will result in fewer recognition errors than the cardinal entry method. The digit-by-digit method should perform better because of its smaller vocabulary and the reduced possibility of having similar sounding utterances. On the other hand, the cardinal method uses a larger vocabulary with several similar utterances (e.g., “fifteen” and “fifty”), and would be expected to result in more recognition errors.

In this experiment, fourteen subjects were selected, seven males and seven females. All subjects were English speaking computer science students from diverse regions of the United States. Each subject was asked to input by voice a set of fifty randomly selected numbers between 0 and 99. Two different random orderings of the numbers were generated. One ordering was used by all subjects for cardinal input and the other ordering was used by all subjects for digit-by-digit input. The selection of which method a subject used first during their testing session was determined at random. The subjects had several minutes of practice time, using smaller, non-case lists before beginning the actual experiment. When the subject spoke the numbers from the lists the program stored into a data file the value that it recognized for each number. An error was recorded for each method of entry when the number stored in a data file did not match the corresponding element on the input list.

The response variable is the total number of errors for each method of entry for each subject. The data were first transformed, using ‘ $\ln(\text{error}+0.5)$ ’, before completing an analysis of variance. The gender and name of a subject are considered as factors, along with the method of entry. The name of the subject was nested within gender. All factors, except NAME, are treated as fixed effects. The reason for considering these factors to be fixed is that a deliberate effort was made to have the same number of males and females and to use only natural born, English speaking citizens of the United States. Also, the methods of entry were specifically chosen using identical lists, with identical numbers were in each list. The factor, NAME, is random in that the inference is made to a larger population than just the students of this department and due to the fact that the number of students available, matching the above criteria, is larger than the fourteen chosen. The experiment was analyzed as a model having nesting in the treatment and design structure by using the following SAS[®] statements:

```
PROC GLM;
CLASS NAME METHOD SEX;
MODEL LNERROR = SEX NAME(SEX) METHOD METHOD*SEX;
RANDOM NAME(SEX)/TEST;
TITLE 'NESTED GLM AOV';
RUN;
```

Based on the results of the above experiment, a second experiment was performed. This experiment explored the effect of selected system parameter settings upon the ability of the speech engine to correctly recognize input strings.

To test the effect of selected system settings on recognition ability, several decisions were made. First, we used an application supplied by VTAF, and not one which we had programmed. The selected VTAF application is a simple one in which users speak a phrase and the system displays what it understood on the computer screen. Second, we tested the same application for each run with the same subject, and the same hardware and software configuration. Also, we

selected and manipulated a subset of the total number of system control parameters.

There are about fourteen system parameters that can be modified (Anonymous, 1995); however, only five were selected, using two levels for each. The five selected were chosen because it was believed that these would have the most potential to influence recognition accuracy. The lower level of a factor was the software default, while the higher level was the value a parameter could assume to provide better accuracy, but at the expense of requiring more system resources. The minimum, default, and maximum values for these five system parameters are presented in Table 1. The parameters (see Table 1) that were evaluated for their effect on the performance of VTAF are (1) language weight, (2) insertion penalty, (3) noise (recognition) sensitivity, (4) beam width, and (5) minimum talk time. Language weight (factor L) determines the balance between acoustic evidence and a language model in pattern recognition. The insertion penalty (factor I) applies to a probability calculation when a new word is inserted into the search hypothesis. Noise sensitivity (factor N) adjusts for sensitivity to competing noises to the sound of the utterance. Beam width (factor B) determines how quickly pruning is done on the search space during recognition. Minimum talk time (factor T) defines the minimum duration of a regular utterance. For this set of five two-level factors, thirty-two runs or treatment combinations are possible.

Two factorial experiments examining these five parameters were completed. Each consisted of a single subject paired with one of two software and hardware configurations. The first configuration was an IBM 755CX Thinkpad equipped with an Andrea Electronics ANC-100 anti-noise canceling microphone. The second configuration was a Dell 450DE PC equipped with an Electro-Voice 257B Cardioid microphone. Each factorial experiment was not replicated; therefore, each one was analyzed using methods described by Milliken and Johnson (1989). Replication is not possible, in this experiment, since only one of each hardware and software configuration was available for each run. The use of more subjects was not considered in this preliminary experiment, and as the results will show (discussed below), the use of more subjects was not necessary. Also, having more subjects would not offer any benefit, because the method of entry experiment indicated that the factor, 'subjects', best serves as a blocking effect and, in our opinion, replication within a block is not possible.

The same three phrases, or utterances, were used for each run of these two factorial experiments. The input utterances are spoken according to the following template:

“Begin-speaking.

Word_1 Word_2 Word_3 Word_4 Word_5 Word_6

Stop-speaking.”

Each phrase has six positions and for each phrase the first, second, and third words are the same; specifically the word, “Testing”. The fourth, fifth, and sixth words are the numbers 1-9 taken in order three at a time. For example, the second phrase is “TESTING TESTING TESTING FOUR FIVE SIX”. An error was recorded whenever one of the displayed words was incorrect in any of the six positions. Therefore, for each phrase entered during a run the possible number of errors are from zero to six. The combined score used in a run would be some additive combination of the score for each phrase. For some runs we learned that the system responded with a message, “No recognized text”. This event was different than other errors; therefore, whenever this message applied to any phrase used in a run, the error score was coded as a seven, instead of six.

Thus, if all phrases in a run were scored as seven, the combined error score was twenty-one. The combined error score for each run is the response variable used in the analysis. The results of each subject/hardware combination was analyzed using the following SAS statements:

```
PROC ANOVA OUTSTAT=SSOUT;
CLASSES L I N B T;
MODEL SCORE = L|I|N|B|T;
RUN;
```

These last experiments utilized a non-replicated five-factor treatment structure where each factor had two levels (Daniel, 1976; Milliken and Johnson, 1989). Therefore, in order to interpret the results, half-normal plots were drawn using the calculated values for the single degree of freedom sums of squares for effects (Milliken and Johnson, 1989). On a half-normal plot, it is assumed that non-significant effect estimates represent random variation, normally distributed about a fixed mean. These values will lie on a straight line when plotted against the quantiles of a standard normal ($N(0,1)$) distribution. Plotted points not falling on a straight line correspond to effects that can be judged significant (Milliken and Johnson, 1989; Stevens et al., 1996).

It was decided, after obtaining the results of the second experiment, that it was not necessary to repeat the trials by crossing the subject/hardware combinations, nor would it be necessary to test additional subjects (see below for further discussion). Further experiments are being planned to provide more guidance on how to write better software using the speech engine in the VTAF toolkit (Anonymous, 1995).

3. Results

The set of experiments reported here have helped the software developers of COTTON TALK tremendously. The principal advantage of the experiments has been to learn more of the capabilities and limitations of the current speech engine. Insight into the performance of VTAF has been acquired with the result that better code for COTTON TALK can be written. A brief discussion of the results of these experiments should clearly indicate how statistical methods can support efforts to design speech recognition applications.

Histograms of the results for methods of numerical entry are presented in Fig. 1. Generally, these plots show that the cardinal method of entry resulted in more errors than the digit-by-digit method. The pattern of errors were similar for both genders within each method. For the digit-by-digit method three subjects had 12 or more errors out of fifty numbers, while the majority had two or fewer errors. Two subjects of each gender had fewer than two errors using the cardinal method. An examination of the original data (Table 2) show that the ordering of subjects (from a low to high occurrence of errors) is not identical between the two methods. Generally, individuals had greater occurrences of errors with the cardinal method as compared to the digit-by-digit method. But, a few individuals (Subjects 1 and 8) had fewer errors with the cardinal method than with the digit-by-digit method, but the difference between the two methods for these individuals was generally small. Four subjects (Subjects 6, 7, 13, and 14) scored high errors for their gender group. For this group of four, one subject (Subject 13) scored much better than the other three with the digit-by-digit method, two subjects (numbers 6 and 14) scored worse with the digit-by-digit method than the cardinal method, while the fourth subject (number

13) scored poorly with both methods, but somewhat better with the digit-by-digit method. For most subjects better results were obtained with the digit-by-digit method than with the cardinal method.

The results of an analysis of variance of the transformed data reflects these generalities. (It is believed that the data should be transformed because both methods had some subjects who performed well and others who performed poorly; see Fig. 1). Using a rejection level of 0.1 and examining the P-values associated with the type III sums-of-squares, gender ($P=0.84$) and the two way interaction ($P=0.34$) were non-significant. The non-significance of gender indicates that the person-to-person variability is not affected by the subject's gender. Tests of hypothesis about means need not be discussed because at this level of completion the analysis is very informative.

The broad conclusion from the results of the first experiment are that the VTAF system recognizes some people's speech much better than others and the design of the speech interface can have a major influence on the error rate of the system. The recognition accuracy for a particular interface design can potentially be improved by two different approaches. First, it may be possible to tune system parameters for a particular subject. Second, it may be possible to train the subject to speak in a manner that is more easily understood by the system. This latter alternative is not very attractive since one of the major reasons for using a speech interface is to allow more natural human-computer interaction.

The results of the two non-replicated 2^5 factorial experiments demonstrate the effects of both the hardware configuration and settings of system parameters. The first finding is that the results of the two subject/hardware combinations were essentially identical; that is, hardware is not influencing the results. Thus, the discussion which follows applies to the findings of each; however, the results (Fig. 2) of only one experiment are presented (i.e., the setup that used the Thinkpad 755CX and Andrea Electronics ANC-100 microphone). A half-normal plot (Milliken and Johnson, 1989) is presented in Fig. 2. Several sources of variation show the most potential for influencing the performance of the speech recognition engine: a main effect (I), a three way interaction ($I*B*T$) and a two way interaction ($B*T$) with single *df* sum-of-squares of 2520.50, 21.13, and 21.13, respectively. The two-way interaction is not labeled on Fig. 2 because it has the exact same plotting position as the three-way interaction. The effect estimates for the next points below the three- and two-way effects ($I*B*T$, $B*T$) have sums-of-squares of 6.125. Hence, due to scaling effects, the point for the three- and two-way interaction is not dramatic in its appearance of being off the line. One can draw the graph after deleting the main effect and then easily notice that these interactions are significant. In general, the half-normal plot is interpreted to indicate that insertion penalty (I) can be adjusted so severely that the performance of the speech engine is exceedingly poor. When values of I are not so extreme, insertion penalty (I), beam width (B) and minimum talk time (T) settings jointly determine performance. Once the COTTON TALK development team discovered that system settings can affect performance, inconsistencies observed in the past began to make sense.

The VTAF documentation does inform application developers that parameter settings can be adjusted to improve performance (Anonymous, 1995). However, our finding that system parameters can interact is not described in that documentation. This is an important omission. We suspect that not considering the influence of interactions during fine tuning efforts can result in many long hours in trial and error searches for the best combination. We can see a solution.

By using experimental design principles, one can avoid trial and error searches by utilizing an optimization experiment. The potential for improving performance by adjustment of system parameters now needs to be assessed by additional experimentation. Current efforts are directed at evaluating the feasibility of developing a VTAF utility to automate the selection of an optimal configuration of system parameters.

4. Discussion

The field of speech recognition is an active area of research (e.g., Baggia et al., 1992; Cole et al., 1995; Philip et al., 1991; Schmandt, 1994). In fact, one leading journal (Transactions on Computer-Human Interaction (TOCHI)), announced this year a call for papers on speech in interactive computing. We are aware of at least three companies that are releasing products this year that are a combination of portable hardware and software tools for speech recognition tasks in outdoor or factory and warehouse applications (Readers interested in the address of these companies are asked to contact JLW). Other companies are developing software for office based applications (Karney, 1995; Malloy, 1994). The VTAF speech engine is, so far, the only one of these products that we have used. Nevertheless, it is clear that applications of speech in many disciplines are increasing (Chen and Robinson, 1990; Cole et al., 1995; Philip et al., 1991).

The demand for computer assisted decision software systems (e.g., Acock et al., 1985; Stone et al., 1987; Baker et al., 1983; Landivar et al., 1991; Sterling et al., 1992) in agriculture is increasing. Currently, however, the use of these systems is limited, because most are data intensive and few users can provide enough time to meet their requirements for timely, reliable information. We initially considered scanner technology (Lindley, 1991) as a solution to this problem along with a neural net to recognize hand-written records. But, for capturing hand-written records of field data, several obstacles exist. The data sheets often become soiled, wrinkled, or characters change in appearance once fatigue occurs, or when the pencil (or pen) becomes dull (or low on ink). Therefore, after encountering several articles on the use of speech (Chen and Robinson, 1990; Karney, 1995; Malloy, 1994; Rash, 1994), we changed our focus to speech recognition interfaces. Someday, we anticipate users of computer based decision support systems will acquire most data in real-time, using only their voice, as they complete necessary tasks. This capability will expand the use of computer-support systems in agriculture and bring their full potential to bear on many production problems.

The most challenging programming task has been to prevent, detect, and process the occurrence of errors. Error control can be facilitated by using both a proper combination of hardware (e.g., a noise canceling microphone) and software (e.g., error trapping and warning of 'out of bound' conditions). Errors may come from extraneous noise (i.e., the shutting of a door), the system's erroneous recognition of the user's input, the user's erroneous input, or perhaps, from being uncomfortable with using the application. Many strategies have been employed to cope with these errors without interfering with user convenience. However, we have found that without the use of designed experiments, one can quickly become bogged down into a trial and error mentality and programming efforts can begin to run in circles. Therefore, it can be seen that the application of experimental design is helping to discover additional strategies to improve system performance and tune VTAF system parameters to improve performance.

There exist several software development models describing different approaches to creating a software product (Jalote, 1991; Nielsen, 1993). The principal model being used to develop COTTON TALK is a process known as prototyping. Prototyping means that in building a software program one creates transient, small scale programs that emulate tasks that the larger, more complete software product is expected to perform. These intermediate prototypes are presented to clients who exercise these programs and then provide feedback to the software developers. The developers learn from the performance of the prototype what design changes need to occur before the final product is completed. In doing the work described in this article, it has been found that principles of experimental design work extremely well with the prototyping development model. Therefore, it seems surprising that in reading the literature on software development so little of it stresses the usefulness of experimental design concepts and tools of analysis. However, there are some exceptions. The evaluation tools described by Harrison and Rainer (1996), Liggesmeyer (1995) and Yau and Luck (1995) use principles familiar to the statistics discipline.

5. Summary

In this article, we discussed the development of COTTON TALK as an example of how speech recognition capabilities, achievable within the foreseeable future, can be used to obtain agricultural data that involve 'eye and hands busy' tasks. Also discussed are the use of traditional statistical methods for helping to design the software that performs the speech recognition task.

The use of several experiments have been found necessary to identify strengths and weaknesses of sample prototypes designed for speech recognition tasks relevant to cotton plant mapping. The majority of experiments completed so far, including those presented in this article, are exploratory in nature. As our application becomes more refined, other statistical methods employed from a confirmatory perspective should also prove valuable.

It is expected that other applications involving the acquisition of agricultural information will have design problems similar to those described for plant mapping. Therefore, it is necessary to continue the use of prototypes and to evaluate their performance with clients familiar with specific tasks. Initially, these simple prototypes can be easily evaluated by watching how they perform, or by asking simple questions of the user. The information learned from a simple example can be employed to develop a more sophisticated and robust prototype. However, as the prototypes develop into a finished product, it becomes necessary to apply more rigorous criteria for assessing performance. Speech interface software applications can profit from employing traditional statistical methods during their development.

6. References

Acock, B., V. R. Reddy, F. D. Whisler, D. N. Baker, J. M. McKinion, H. Hodges & K. J. Boote. 1985. The soybean crop simulator GLYCIM: Model documentation. US Department of Agriculture, Washington, D.C. Publication No. PB85 171163/AS.

- Anonymous, 1995. VoiceType Application Factory for Windows, Version 1.1. Technical Reference, 2nd edition. International Business Machines Corporation.
- Baggia, P., L. Fissore, E. Gerbino, E. Giachin & C. Rullent. 1992. Improving speech understanding performance through feedback verification. *Speech Communication* 11: 289-297.
- Baker, D. N., J. M. McKinion & J. R. Lambert. 1983. GOSSYM: A simulator of cotton crop growth and yield. South Carolina Agricultural Experiment Station Technical Bulletin 1089.
- Bowden, R. O., R. G. Luttrell & W. S. Shin. 1992. Decision Aid for Managing Cotton Insects. *Interfaces* 22: 86-94.
- Chen, Y. R. & S. A. Robinson. 1990. Integrating a knowledge-based meat-grading system with a voice-input device. *Computers and Electronics in Agriculture* 4: 303-313.
- Cole, Ron, L. Hirschman, L. Atlas, M. Beckman, A. Biermann, M. Bush, M. Clements, J. Cohen, O. Garcia, B. Hanson, H. Hermansky, S. Levison, K. McKeown, N. Morgan, D. G. Novick, M. Ostendorf, S. Oviatt, P. Price, H. Silverman, J. Spitz, A. Waibel, C. Weinstein, S. Zahorian & V. Zue. 1995. The Challenge of Spoken Language Systems: Research Directions for the Nineties. *IEEE Trans. On Speech and Audio Processing* 3(1): 1 - 20.
- Daniel, C. 1976. *Applications of Statistics to Industrial Experimentation*. John Wiley and Sons, New York.
- Gauch, H. G., Jr. 1993. Prediction, Parsimony and Noise. *American Scientist* 81: 468-478.
- Harrison, A. W. & R. K. Rainer. 1996. A general measure of user computing satisfaction. *Computers in Human Behavior* 12(1): 79-92.
- Hedges, L.V. & I. Olkin. 1985. *Statistical Methods for Meta-Analysis*. Academic Press, Boston, Massachusetts.
- Jalote, P. 1991. *An Integrated Approach to Software Engineering*. Springer-Verlag, New York.
- Karney, J. 1995. Take a letter, HAL. *Computer Shopper*, February, pp. 558-560.
- Landivar, J. A., B. Eddelman, J. Benedict, D. J. Lawlor, D. Ring & D. T. Gardiner. 1991. ICEMM, an integrated crop ecosystem management model. Proceedings, Beltwide Cotton Conference, National Cotton Council, Memphis, Tennessee. pp. 453-458.
- Lemmon, H. E. 1986. COMAX: An expert system for Cotton Crop Management. *Science* 233: 29-33.

- Liang, C. 1995. Directed Project. Department of Computer Science. Mississippi State University, Mississippi State, MS.
- Liang, C., J. L. Willers, S. Bridges & J. M. McKinion. 1996. *Cotton Talk: A Speech Interface for Cotton Plant Mapping*. Proceedings, Beltwide Cotton Conference, National Cotton Council, Memphis, Tennessee. pp. 522-528.
- Liggesmeyer, P. 1995. A set of complexity metrics for guiding the software test process. *Software Qual. Journal* 4: 257-273.
- Lindley, C. A. 1991. *Practical Image Processing in C: Acquisition, Manipulation, and Storage*. John Wiley & Sons, New York.
- Malloy, R. 1994. The speech recognition revolution. *OS/2 Professional* 2(3): 26-32.
- Milliken, G. A. & D. E. Johnson. 1989. *Analysis of Messy Data, Vol. 2. Nonreplicated Experiments*. Van Nostrand, Reinhold. New York.
- Nielsen, J. 1993. *Usability Engineering*. AP Professional. Chestnut Hill, MA.
- Olson, R. L., T. L. Wagner, S. Yatham, M. R. Williams & J. L. Willers. 1992. Design and Implementation of rbWHIMS: An expert for cotton pest management. Proceedings, Beltwide Cotton Conference, National Cotton Council, Memphis, Tennessee. pp. 777- 782.
- Pedigo, L. P. & G. D. Buntin, eds. 1993. *CRC Handbook of Sampling Methods for Arthropods in Agriculture*. CRC Press, Boca Raton, Florida.
- Philip, G., B. F. Peters, F. J. Smith, D. Crookes & T. Rafferty. 1991. Design and evaluation of a speech interface for remote database searching. *J. Information Sci.* 17: 21-36.
- Rash, Jr. W. 1994. Talk Show. *PC Magazine*, Dec. 20: 203-219.
- Schmandt, C. 1994. *Voice Communications with Computers*. Van Nostrand, Reinhold. New York.
- Sterling, W. L., A.W. Hartstack & D. A. Dean. 1992. TEXCIM50: The Texas cotton-insect model. Texas Agricultural Experiment Station Misc. Publication MP-1646.
- Stevens, G. L., J. L. Willers, R. A. Sequeira & P. D. Gerard. 1996. Analysis of Deterministic Simulation Model Performance Using Non-Replicated Factorial Two-Level Experiments. *Agricultural Systems* 52(2): 293-315.

Stone, N. D., R. E. Frisbie, J. Richardson & C. Sansone. 1987. COTFLEX, a modular expert system that synthesizes biological and economic analysis: The pest management advisor as an example. Proceedings, Beltwide Cotton Conference, National Cotton Council, Memphis, Tennessee. pp. 194-197.

Willers, J. L., M. R. Williams, T. L. Wagner & R. L. Olson. 1992. What is the price of information? Proceedings, Beltwide Cotton Conference, National Cotton Council, Memphis, Tennessee. pp. 783- 786.

Yau, C. & Y. G. Luck. 1995. Comparing the top-down and bottom-up approaches of function point analysis: a case study. *Software Qual. Journal* 4: 175-187.

Table 1. The minimum, default, and maximum arguments for the five system parameters examined in the factorial experiment on performance of the speech recognition engine.

Parameter Name	Minimum Value	Default Value	Maximum Value
Language Weight (L)	0.0	6.0	10.0
Insertion Penalty (I)	-1.0	0.4	1.0
Noise Sensitivity (N)	-1.0	-0.6	1.0
Beam Width (B)	0.0000001	0.00005	0.1
Minimum Talk Time (T)	0.1	0.2	1.0

Table 2. The subject labels, gender (F=female, M=male), and error scores for the cardinal (C) or digit-by-digit (D) methods of numerical entry.

Subject_1	F	C	0	Subject_4	F	D	0
Subject_2	F	C	2	Subject_5	F	D	0
Subject_3	F	C	6	Subject_3	F	D	0
Subject_4	F	C	7	Subject_1	F	D	1
Subject_5	F	C	9	Subject_2	F	D	1
Subject_6	F	C	16	Subject_7	F	D	13
Subject_7	F	C	18	Subject_6	F	D	19
Subject_8	M	C	0	Subject_9	M	D	0
Subject_9	M	C	1	Subject_10	M	D	1
Subject_10	M	C	3	Subject_12	M	D	2
Subject_11	M	C	4	Subject_8	M	D	2
Subject_12	M	C	6	Subject_11	M	D	2
Subject_13	M	C	8	Subject_13	M	D	2
Subject_14	M	C	9	Subject_14	M	D	14

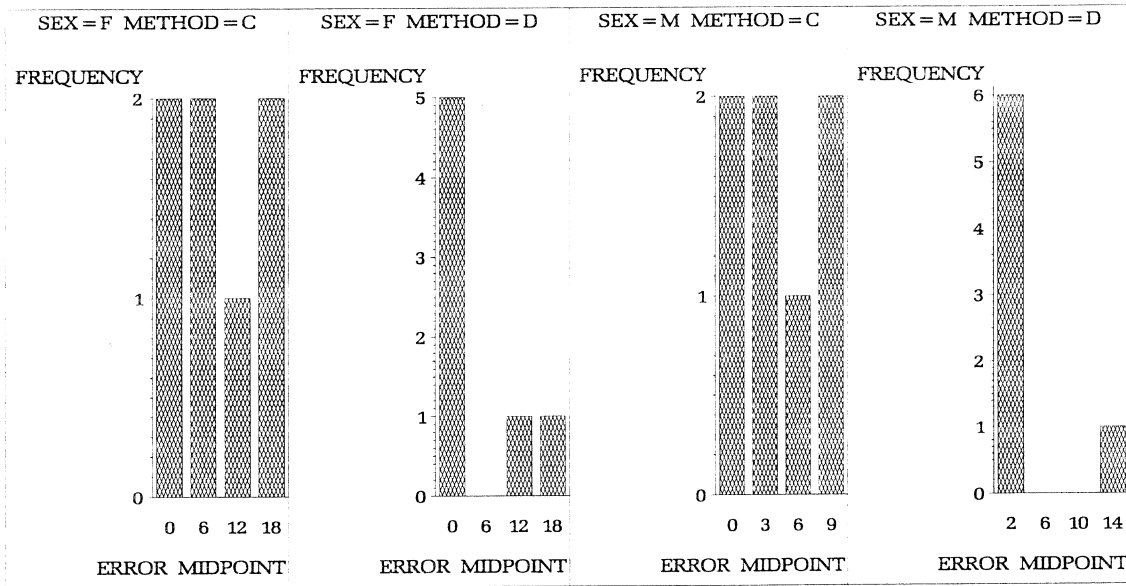


Figure 1. Histograms of error frequencies of interface prototypes for females (F) and males (M) for both the cardinal [C] and digit-by-digit [D] methods for numerical entry.

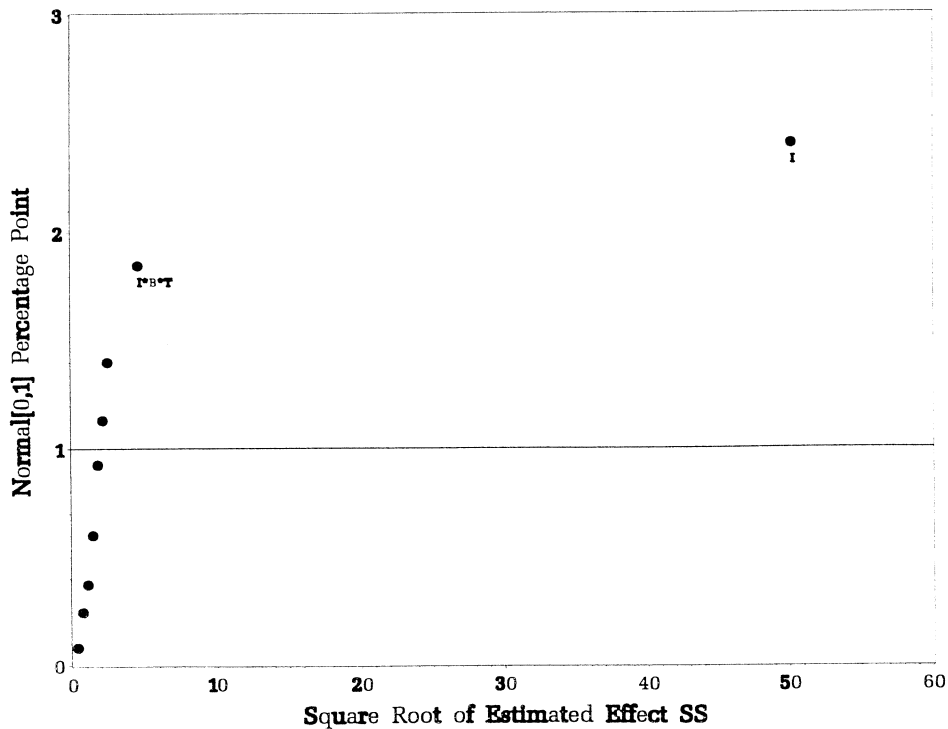


Figure 2. Half-normal plot of two levels of settings for five system parameters. Insertion penalty (I), beam width (B), and minimum talk time (T) interact, whilst insertion penalty is a dominating main effect.