AFIT/GIR/LAS/99D-1

IMPLEMENTATION OF SPEECH RECOGNITION SOFTWARE FOR TEXT PROCESSING: AN EXPLORATORY ANALYSIS

THESIS

Sean P. Abell, First Lieutenant, USAF

AFIT/GIR/LAS/99D-1

Approved for public release, distribution unlimited

20000307 063

DTIC QUALITY INSPECTED 3

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the US Government.

AFIT/GIR/LAS/99D-1

IMPLEMENTATION OF SPEECH RECOGNITION SOFTWARE FOR TEXT PROCESSING: AN EXPLORATORY ANALYSIS

THESIS

Presented to the Faculty of the Graduate School of Engineering and Management of the

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Information Resources Management

Sean P. Abell, B.A.

First Lieutenant, USAF

December 1999

Approved for public release, distribution unlimited

AFIT/GIR/LAS/99D-1

IMPLEMENTATION OF SPEECH RECOGNITION SOFTWARE FOR TEXT PROCESSING: AN EXPLORATORY ANALYSIS

Sean P. Abell
First Lieutenant, USAF

Approved:

Michael G. Morris, Major, USAF, Ph.D.	1 DEC 99
Advisor	date
David P. Biros, Major, USAF, Ph.D.	1 Dec 99
Reader	date

Acknowledgements

I would like to thank the members of my thesis committee, Major Mike Morris and Major Dave Biros. Their encouragement and humor have helped to make the thesis process enjoyable. This thesis would not have been possible without the help of Captain Alan Fiorello. My appreciation of his creativity and initiative throughout the experiment design and data collection phases of this thesis cannot be understated. Thanks, Al. I would also like to thank my wife, Michelle, and my son, Zachary for their unwavering support and love throughout the last eighteen months.

Table of Contents

Page

Acknowledgements....ii

List of Figures....v

List of Tables....vi

Abstract....vii

I. Introduction....1

Background....1
History of Speech Recognition....2
Specific Problem....2
Research Question....4
Justification for Research....4
Summary....5

II. Literature Review....7

Introduction....7
Speech Recognition Software (SRS)....8
Classes of Speech Recognition....11
Productivity Paradox....12
Productivity Measurement....14
Reactions to the Productivity Paradox....15
Previous Speech Recognition Productivity Research....16
Summary....17

III. Method....18

Research Approach....18
Technology....19
Procedure....20
Sample....19
Day One....21
Day Two....22
Statistical Analysis....25

IV. Results....27
Assumptions Tests....27
Results....30

Initial Text Entry Results.....31
Proofreading Results.....32
Total Text Processing Time Results.....33
Initial Text Entry Error Results.....34
Proofreading Error Results.....36
Summary.....38

V. Discussion.....39

General Discussion.....39
Implications.....41
Limitations.....42
Conclusion.....43

Appendix A. Survey.....44

Appendix B. Demo Task 1.....45

Appendix C. Training Aid.....46

Appendix D. Experiment Script.....47

Appendix E. Demo Task 2.....55

Appendix F. Demo Task 3.....56

Appendix G. Task 1.....57

Appendix H. Task 2.....60

Bibliography.....63

Vita 65

List of Figures

Figure **Page**

1. Day 1 Timeline....21

2. Day 2 For Odd Numbered Subjects....24

3. Day 2 For Even Numbered Subjects....24

4. Distribution Plot of Residuals for Initial Text Entry Times....28

5. X by Y Plot of Residuals for Initial Text Entry Times....29

6. Control Chart for Initial Text Entry Observations....30

7. Initial Text Entry Time Means Comparison....32

8. Proofreading Time Means Comparison....33

9. Total Text Processing Time Means Comparison....34

10. Initial Text Entry Error Means Comparison....35

11. Proofreading Error Means Comparison....37

List of Tables

Table	Page
1. Task Statistics	22
2. Subject Demographics	25
3. Two Latin Square Replications	26
4. Normality Assumption Test Statistics	28
5. Constant Variance Assumption Test Statistic	29
6. Independence Assumption Test Statistic	30
7. Initial Text Entry Time F-Test Results	32
8. Proofreading Time F-Test Results	33
9. Total Text Processing Time F-Test Results	34
10. Initial Text Entry Error F-Test Results	35
11. Proofreading Error F-Test Results	37

Abstract

In recent years, information technology has advanced at an incredible pace. One new technology that has recently become available to the average computer user is speech recognition software for text processing. The rationale behind implementing such new technologies is often to gain productivity improvements associated with the substitution of machinery for labor. However, the literature shows little direct evidence of a positive relationship between information technology investment and subsequent productivity benefits.

This thesis reports on the examination into the productivity implications of implementing speech recognition software in a text-processing environment. More specifically, research was conducted to compare text processing speeds and error rates using speech recognition software versus the keyboard and mouse. Of interest was the time required to input and proofread text processing tasks as well as the number of errors generated using both methods of text input.

The empirical data offer somewhat mixed results. While users initially entered text faster using speech recognition software ($p < .05$), they generated more errors and consequently performed proofreading and error corrections slower using speech. These results suggest that, in terms of accurate text processing, speech recognition software is still not a practical alternative to the keyboard. Therefore, implementation of speech recognition software is unlikely to result in any gains in productivity that would serve to justify its cost.

IMPLEMENTATION OF SPEECH RECOGNITION SOFTWARE FOR TEXT PROCESSING: AN EXPLORATORY ANALYSIS

I. Introduction

Background

Mention speech recognition today, and it is almost inevitable that someone will point to HAL, the computer from 2001: A Space Odyssey. This illustration of where the technology is headed has lulled many information technology (IT) managers into ignoring speech recognition because it is obvious that computers that can hold an intelligent conversation will remain science fiction for a long time. The fact is that practical, usable speech recognition products are here now. For example, several companies now sell continuous speech recognition applications that offer the everyday computer user the ability to input and format text in a word processing program with, the companies claim, the speed and accuracy that now rivals the traditional keyboard and mouse. Dragon Systems introduced the first general-purpose continuous-speech recognition program for the personal computer (PC) in June 1997. IBM Corp. followed soon after with the introduction of IBM ViaVoice. There is no doubt that speech recognition software has improved significantly in a short time. Most current speech recognition applications offer large active vocabularies. Also, the speech recognition engines have become robust. More importantly, a user can now dictate directly into most popular applications like Microsoft Word (Alwang, 1998).

History of Speech Recognition

In its most basic form, speech recognition involves the process of a computer matching an acoustic signal to some text. While this may sound relatively simple, speech recognition software development spans a huge range of scientific disciplines, from linguistics and biology to computer science and artificial intelligence. The ultimate aim of those working on speech recognition is to produce a system that enables humans to communicate with computers as they would with other humans i.e.. using natural speech (Rodman, 1999).

Voice and speech recognition have been around since the early 1970s, when research was conducted on these technologies at the Defense Advanced Research Projects Agency (DARPA) (Katz, 1993). While commercial applications existed in the '80s and early '90s, they were cost and technology prohibitive. Today's microprocessor technology, though, has brought voice and speech recognition out of the laboratories, and into the mainstream consumer market (Lorek, 1997).

Although attempts have been made by many companies over the last 15 years to introduce low cost products utilizing speech recognition, the products have been few and far between and market failures have been common. For example, the toy industry historically, has been plagued by poor speech recognition accuracy leading to a very high percentage of returns on products such as voice controlled cars (Markowitz, 1996).

Specific Problem

In the early 1950s, the rationale behind computerization was to gain productivity improvements associated with the substitution of machinery for labor. However, little

direct evidence is available of the relationship between information technology investment and performance. When a relationship is found, the results cannot be generalized beyond the particular industry study (Katz, 1993).

The obvious growth of computer information-processing industries since the 1950s might suggest that every expectation of productivity payoffs has been fulfilled. With quality-adjusted investment in new computer equipment near $500 billion during the 1990s, U.S. firms have clearly embraced the computer. The problem, however, is that economy-wide productivity growth remains well below historic averages (Brynjolfsson, 1993). The rise in computer investment coupled with slow growth in productivity is commonly referred to as the "Computer Productivity Paradox" (Brynjolfsson, 1993).

As with any computer technology, it is important to evaluate the level of productivity that can be realized by its implementation. Questions about the business value of Information Technology (IT) have perplexed managers and researchers for a number of years. Businesses continue to invest enormous sums of money in computers and related technologies, presumably expecting a substantial payoff. Yet two of studies present contradictory evidence as to whether these expected benefits have materialized (Brynjolfsson 1993a; Wilson 1993).

It is critical to answer these productivity questions because, from a managerial perspective, it is important to understand how investment in IT affects the bottom line. "Measuring IS effectiveness" is consistently reported in the top 20 on the list of most-important IS issues by the members of the Society for Information Management (SIM), an organization of IS executives (Ball & Harris, 1982; Brancheau & Wetherbe, 1987;

Dickson, Leitheiser, Nechis & Wetherbe, 1984; Niederman, Brancheau & Wetherbe, 1991).

Research Question

Information Systems (IS) managers are under increasing pressure to justify the value and contribution of IS expenditures to the productivity, quality, and competitiveness of the organization (Myers et al., 1997). In order to justify the procurement of a SRS system for an organization, it is important to define what specific productivity improvements are to be expected.

The problem considered in this research is: Can speech recognition software provide comparable or increased productivity when compared to conventional text input methods of keyboard and mouse? The productivity measures being specifically evaluated are those of time required to perform text processing tasks as well as an evaluation of text processing error-rates.

Justification for Research

It is reasonable to assume that speech recognition has the potential to quickly pervade conventional office automation environments, particularly in the financial and travel-services industries. But for most IT managers, it is another matter when considering desktop computer users. With the exponential growth of computing technology, the office automation environment in a constant state of flux (Markowitz, 1996). Despite this, migrating from conventional text processing modes, using exclusively the keyboard and mouse, to speech recognition software requires a significant

paradigm shift. After all, why change when the keyboard and mouse have done the job for years? Also, the issue of an increased and confusing noise level in cubicle-filled work environments must also be considered.

Some of the problems that plagued early pioneers attempting to enable consistent, reliable speech recognition still remain. For example, every person speaks differently, with various noises or disturbances in their speech. Pausing, clearing the throat, coughing or using sounds like "uh," "um" and "ah" all may send the "listening" computer into confusion (Rubio, et al, 1995). Fast talkers tend to run their words together even more than speakers with normal pacing. Quite often there is background noise that "pollutes" incoming voice signals, making it difficult for the computer to accurately identify relevant sounds. In addition, many words sound alike, putting the burden of understanding meaning into the computer. This is commonly referred to as natural language processing (NLP), where computers must not only recognize speech, but understand what the words mean. All of these challenges have been met to some extent, as an evaluation of current speech recognition products will verify (Markowitz, 1996).

Summary

This thesis reports on an empirical evaluation conducted to assess the productivity measures of speed and error rates of users applying speech recognition software to text processing tasks compared to conventional text input modes of keyboard and mouse. A review of the relevant literature of speech recognition software and information technology productivity measurement will be followed by a detailed presentation of the methodology used in this research. Finally, this thesis will present the statistical analyses

and discussion of the results and discuss those findings in the context of IT productivity in the business environment.

II Literature Review

Introduction

The primary intent of this literature review is to collect and examine the relevant knowledge and research stream related to the history and current state speech recognition software as well as issues related to the assessment of information technology productivity. In this literature review, the term IT is an umbrella term that includes the integrated user machine systems for providing information to support the operation, management, analysis and decision-making functions in an organization. The systems use computer hardware, software, and communications equipment; manual procedures, models for analysis, planning, control and decision making.

One new IT product that has recently become available for the typical computer user is speech recognition software (SRS). SRS allows a computer user to input information and execute commands by simply talking into a microphone connected to a computer. The software then converts these audio waves into digital instructions that a computer processor can understand and execute. Actually, SRS is being used every day by hundreds of thousands of people (Koerner, 1996). The telecommunications and banking industries currently use this software in their automated phone systems (Lorek, 1997).

There are two reasons why this technology is becoming more popular. The first is that computer hardware in now available to take advantage of the technology, and the second is that it is now affordable (Lorek, 1997). Previously, computer hardware strong enough to run this technology was too expensive for the general public to purchase. But

now, with the advancements in central processing units and digital signaling processors, SRS can be effectively utilized (Markowitz, 1996).

Speech Recognition Software (SRS)

Since the late 1950s, researchers have been developing voice-input interfaces to computers based on automatic speech recognition technology (Leface and Renato, 1992). Advances in the 1960s and 1970s in digital signal processing, pattern matching and classification algorithms, and computer hardware, has made the speech-based computer user interface a reality (Karl, Pettey, and Sheiderman, 1993:667). The first commercially available computer capable of identifying spoken words from limited vocabulary appeared in the early 1970s (Koerner, 1996).

Historically, hardware limitations have been one of the biggest barriers in speech recognition development (Rodman, 1999). Intel, a leading microprocessor manufacturer, has recently released a new generation of processors; the Pentium® III (PIII), which is a follow-on to Intel's MMX Technology. Among the most important new instructions in the PIII is a new memory-streaming architecture (Rodman, 1999). The new instructions enhance the performance of speech recognition applications. The new processor will be designed to better handle complex speech recognition algorithms used by speech recognition software developers (Lorek, 1997). In addition, Lorek (1997) suggests that the end result will be a reduced error rate and a shortened response time. This enhanced capability will likely result in speech recognition being integrated in a growing number of business and consumer applications.

Even though speech recognition technology has been an ongoing research topic since the 1950s, it is only now that a convergence of technology breakthroughs is making speech ready for broader use. Alwang, (1998) outlines the following breakthroughs:

- A steady increase in affordable computing power, most recently the Pentium® III processor which runs at speeds up to 550 MHz
- Development of the overall PC platform such as Universal Serial Bus (USB) and faster memory technologies
- Improvements in speech algorithms and advances in signal processing
- Broad industry support from a range of software developers pursuing various types of speech-enabled applications.

Today, the minimum computer hardware requirements to use SRS is a Pentium II-200 Mhz, IBM compatible PC with 48 MB of RAM, and 180 MB of free hard disk space (Alwang, 1998).

SRS operates on the principle referred to as *template matching* (TM) (Frankish et al, 1992:798). The basic functionality of TM is best described as the comparison of spoken input to stored speech patterns provided by individual users (Frankish et al, 1992:798). Bell Telephone Laboratory was issued the patent for TM in 1982. The following is the abstract for the TM patent:

> In a speaker recognition and verification arrangement, acoustic feature templates are stored for predetermined reference words. Each template is a standardized set of acoustic features for one word, formed for example by averaging the values of acoustic features from a plurality of speakers. Responsive to the utterances of identified speakers, a set of signals representative of the correspondence of the identified speaker's features with said feature templates of said reference words is generated. An utterance of an unknown speaker is analyzed and the reference word sequence of the utterance is identified. A set of signals

> representative of the correspondence of the unknown speaker's utterance features and the stored templates for the recognized words is generated. The unknown speaker is identified jointly responsive to the correspondence signals of the identified speakers and unknown speaker (Koerner, 1996, 240).

The process of inputting speech samples for the purpose of TM is refered to as training. As implied, a user, for all practical purposes, trains the software to recognize his or her own unique speech patterns. For each reference word sampled by the SRS during training, the SRS will generate a coded spectral representation which serves as the reference template (Frankish et al, 1992:798). Coding is achieved by first passing the speech signal though a bank of approximately 20 band-pass filters with center frequencies required for speech perception, approximately 300-5000 Hz. By successively sampling the outputs of these filters for time intervals of 10 to 20 milliseconds, feature vectors or frames are generated which are then combined to form an acoustic pattern for a completed word. Thus, speech templates are generated that consist of a set of values arranged in a two-dimensional matrix, *frequency and time* (FT) (Frankish et al, 1992:798).

When the training is complete, the SRS is enabled to match the incoming speech signal that was digitally encoded in the same manner as the templates by computing the degree of similarity between the incoming FT matrix to the various template FT matrices (Frankish et al, 1992:798). Similarity is expressed as a single numerical value refered to as distance score and the template which yields the smallest distance score is normally selected as the most probable match for the incoming speech (Leface and Renato, 1992).

Classes of Speech Recognition

Speaker-dependent speech recognition systems have been the standard for commercial applications (Leface and Renato, 1992). This type of system involves training through repetition to recognize a vocabulary of words from a particular user and is based on a template representation of speech. Users train speaker-dependent systems on their voice patterns by speaking voice samples or words that must be recognized. The computer then stores these templates of voiceprints in the system. Later, when speech recognition is enabled, the system compares the spoken commands with the stored voiceprints. When the voiceprints and the spoken commands match, the system instructs the computer to execute the command. Most of the currently available commercial software packages integrating speech recognition are based on speaker-dependent systems (Markowitz, 1996).

Speaker-independent speech recognition systems are more like what are typically envisioned in science-fiction works (Leface and Renato, 1992). These systems have the ability to recognize speech regardless of who it comes from. These types of systems were quite rare until recently and can still be difficult to create, as they must be able to accurately recognize words from any speaker (Markowitz, 1996). Speaker-dependent and the speaker-independent speech recognition systems and further be divided into two additional types of systems based on what type of speech signal can be input; discrete and continuous.

Discrete speech recognition systems make users separate each spoken word with a pause. This technique makes it easier for the system to recognize words, since each word has a distinct beginning and ending. It also requires users to speak each word

slowly and separately, resulting in consistent pronunciation. While discrete speech recognition systems result in more accurate translation, they can be awkward at times and results in user frustration (Frankish et al, 1992:798).

Continuous speech recognition systems eliminate the need for a user to pause between words. The systems, compared to discrete systems, are much more natural for humans to work with, but traditionally result in accuracy problems (Koerner, 1996). These accuracy problems result because speakers can often run words together, which presents the problem of recognizing where words start and stop. And, in continuous speech, it has been found that users tend to pronounce certain words and phrases differently than in discrete speech (Koerner, 1996). An example is the phrase "going to," which when speaking continuously, is sometimes pronounced "gonna." Fortunately, today's powerful computing environment makes the accuracy problem less of an issue as programmers have found ways to overcome continuous recognition issues using more complex algorithms (Rubio, 1995).

Productivity Paradox

Investment in IT products, like speech recognition software, can provide an organization with both tangible and intangible benefits. To prove the existence of these benefits, information systems managers must be able to evaluate IT's advantages in order to justify their costs. To that end, over the past 15 years, both academia and the business press have periodically investigated and reported on the so-called productivity paradox of computers (e.g., Dickson et al, 1984; Hartog and Herbert, 1986; Brancheau, and Weatherbe, 1991; Niederman, Scudder and Kucic, 1991; McLean and Kappelman, 1993;

Wilson, 1993; Brynjolfsson, 1993; Crowston & Treacy, 1996). The paradox is that, despite delivered computing-power in the U.S. having increased by more than two orders of magnitude since the early 1970s, productivity, especially in the service sector, seems to have stagnated (Brynjolfson and Yang, 1996). Evaluating productivity enhancements gained by the implementation of IT is so critical, Rochester & Douglass, (1991, 14) suggest, "Assessing the value of the IT infrastructure is perhaps the biggest single problem for the 90s - the information technology organization is running out of credibility and managers are no longer willing to give us the benefit of the doubt".

In fact, effectiveness of the IT function has proven practically impossible to define and measure (Niederman et al.). Crowston and Treacy (1996) describe many possible explanations for this. For example, the role of the IS function in business performance can be subtle and difficult to differentiate from other factors. Some companies use weak 'surrogate' measures of IS effectiveness that hide the true value of the IT function. Others depend mostly on qualitative rather than quantitative measures. (Hartog & Herbert, 1986; Marion, 1992; McLean, Kappelman & Thompson, 1993).

These issues are critical to organizations with large investments in IT. Companies have come to realize they are paying big money for technology that is not being used (King, 1991). Furthermore, a recent survey of senior executives from 220 Fortune 1000 firms found extremely low satisfaction with returns on corporate technology investments. Over 81 percent of those polled rated their organization's payback on technology spending as minimal or average (Maglitta, 1993).

Productivity Measurement

Some researchers believe that the lack of evidence of a payoff for the high investment in technology could be interpreted as reflecting serious measurement difficulties. These measurement difficulties are evident in the vast research on IT productivity (Schumann, 1989; Berndt & Morrison, 1991; Katz, 1993; Baatz, 1994; Brynjolfsson, 1993). For example, Hitt and Brynjolfsson, 1994 suggest three rather admittedly vague measures of IT productivity. Their research clarified the point that there are three related, but distinct dimensions to the question of IT productivity: the effect of computers on productivity, the effect of computers on business performance, and the effect of computers on consumer surplus.

While their research found evidence that IT may be increasing productivity and consumer surplus, but not necessarily business profits, it also showed that there is no inherent contradiction if computers create value but destroy profits. In other words, the research suggests that firms are making the necessary IT investments to maintain competitive parity, but are not able to gain competitive advantage. An analysis of investments in other capital resources is not so elusive. Schumann, (1989) suggests that in investments in manufacturing resources, for example, can produce a return on investment of 20% or higher. In most cases, the payback period for manufacturing applications is less than three years. Automated office systems, on the other hand, may yield less than 20% and may even result in negative returns on investment.

Hartog and Herbert (1986), as well as Marion (1992), and McLean, Kappelman and Thompson (1993) suggest the difficulty in evaluating productivity improvements resulting from the implementation of IT is compounded by the fact that those responsible

for the implementation are often not sensitive to the issue. Frequently, information technology is used without a full understanding of its applicability, effectiveness, or efficiency. Information systems managers often lack the tools they need to decide if they are accomplishing the right activities (Schumann, 1989). In addition, these managers often fail to learn if they are meeting the needs of their customers. The productivity of the information systems function from the perspective of the customer has also proven difficult to define and measure (Scudder & Kucic, 1991).

Reactions to the Productivity Paradox

In contrast to those researchers who subscribe to the basic notion of the productivity paradox, some in the IT research community point out flaws in the idea that IT productivity can be measured in the same manner as other capital expenditures. Yannis, (1995) suggests that it is inappropriate to blame computers for inadequate productivity growth in that period. Computer investments in the 1970s and 1980s pale in comparison to the trillions of dollars of machinery, buildings and other assets that firms had accumulated over several decades (Rochester and Douglas, 1995). In addition, Wilson, (1993) points out that it takes time for companies to assimilate information technology and reorganize to take advantage of it. And this restructuring, which is often painful, did not happen wholesale until the late 1980s.

Much of the supposed productivity shortfalls of the 1980s may have been misleading. Yannis, 1995 suggests that our tools for measuring productivity designed for counting bushels of wheat and Model Ts off Ford's assembly line are ineffective when used to measure the tremendous improvements in service, quality, convenience, variety

and timeliness that IT can provide. This is especially true in the service sector, where output data is unreliable and things that cannot be measured are assumed not to exist.

More recent studies, based on current data and sophisticated analysis, are closing in on the defining the value of information technology. For instance, research by Brynjolfsson and Hitt, (1996) evaluated the productivity in 380 large firms that generate yearly sales in excess of $1.8 billion. This research found that computers were far from unproductive: They were significantly more productive than any other type of investment these companies made. The gross return on investment averaged about 60% annually for computers, including supercomputers, mainframes, minis and micros. In addition, their research concluded that IT staffers were more than twice as productive as other workers.

The large number of companies studied is likely to average out any errors, and the multiyear data means this is likely not a statistical anomaly. These findings applied to manufacturing and service firms and have since been replicated by other researchers.

Previous speech recognition productivity research

To this date, relatively little research has been conducted with the goal of assessing the expected productivity improvements resulting from implementation of SRS in an office automation environment. The empirical studies focused primarily on determining performance differences when speech input replaced traditional keyboard input in restricted applications. Performance measures for these early studies were usually speed and error rates, and results were typically contradictory and inconclusive (Pettey & Shneiderman, 1993).

However, recent studies have been conducted to investigate the utility of SRS for word processing applications. Typical word processing tasks have three basic activities that involve direct interaction with the computer – text entry, command execution, and direct manipulation activities such as cursor positioning and text selection. These studies differed from the first studies in that SRS was used in concert with traditional input devices (mouse, keyboard) instead of simply replacing them.

In a study by Pettey & Shneiderman, (1993) they concluded that, using SRS in addition to the mouse and keyboard, is 18.7% faster than keyboard/mouse only, and error rates remained the same for both groups. The results suggest that speech input for command activation provides improved performance over mouse activation of commands in word processing applications, particularly for tasks that are command intensive or that require formatting of text as it is entered, as in scientific formula tasks and long typing tasks.

Summary

The difficulties and lingering questions stated in this literature review concerning the measure of IT's impact on organizational productivity are important considerations for an organization seeking to invest in new technologies such as speech recognition software. With this in mind, the research reported in this thesis will attempt to answer two critical productivity questions involving speech recognition software. First, how does speech recognition software compare to the keyboard in terms of task completion time? Second, how does speech recognition software compare to the keyboard in terms of accuracy?

III. Method

Research Approach

This chapter presents details of the experiment and data gathering conducted to evaluate productivity implications of using speech recognition software versus a keyboard and mouse for text processing. Research by Pettey and Schneiderman (1993) evaluated speech recognition software using a similar experimental approach.

The experiment was conducted in a laboratory setting at the Air Force Institute of Technology, Wright-Patterson Air Force Base, Dayton, Ohio. An industry-leading continuous speech recognition software package was used to input and process text into a popular word processing software package. The speech recognition software was evaluated in terms of task completion times and error rates using speech recognition software versus a keyboard and mouse to input text and correct errors. Task completion times and error rates have long been established as measures of software in software usability and productivity testing (Shneiderman, 1998).

For the initial text entry portion of both tasks, task completion times were measured starting with the initial spoken or typed word for each task. For the proofreading portion of the experiment, subjects were instructed to start at the beginning of the document and timing was begun. Timing was ended when the subjects indicated they were finished proofreading. Error rates were determined by evaluating the tasks performed by the subjects and determining any deviation from the original document. Errors included spacing, formatting, spelling, mis-recognition, and punctuation deviations.

Technology

The speech recognition software package used for this experiment was Dragon NaturallySpeaking version 3.01, Professional Edition. The word processing software package used for this experiment was Microsoft Word 97. Dragon NaturallySpeaking was selected as the speech recognition software package after studying several speech recognition software reviews. For example, PC Magazine, October 20, 1998 selected Dragon NaturallySpeaking 3.0, Preferred Edition as its "Editor's Choice" for continuous speech recognition software (Jecker, 1998). "The three most important features of speech recognition software are: accuracy, accuracy, accuracy. Dragon NaturallySpeaking includes many innovative features, but Dragon Systems has focused most of its efforts on developing the most accurate speech recognition engine. As our tests demonstrate, they've succeeded" (Jecker, 1998, 8). Microsoft Word was selected based primarily on its popularity and prevalent use throughout Air Force office automation environments.

Sample

Thirty-two subjects participated in this study. The number of subjects used provided sufficient statistical power for this experiment. The participants in the experiment were primarily active duty Air Force members whose ranks ranged from E-2 to O-4. The participants also included a small number of civilians. Before participating in the experiment, all subjects were asked to complete the survey (Appendix 1) describing their level of expertise with both software packages used in the experiment. While the participants' expertise using Microsoft Word was varied, their experience and

expertise with speech recognition software was none. Table 2 presents basic demographic statistics for the subjects.

Table 2. Subject Demographics

Sex	M	F				
	23	7				
Age	Average	Min	Max			
	31.5	22	67			
Rank	E-1-E-4	E-5-E-6	E-7-E-9	O-1-O-3	>O-3	Civilian
	2	3	1	19	1	4
Perceived Proficiency/Word	None	Novice	Intermediate	Advanced		
	0	4	21	5		
Hours/Word	None	1-5	5-10	10-20	>20	
	0	8	13	8	1	
Perceived Proficiency/SRS	None	Novice	Intermediate	Advanced		
	30	0	0	0		
Perceived Proficiency/Dragon	None	Novice	Intermediate	Advanced		
	30	0	0	0		

Procedure

After completing the initial survey, participants were briefed on the overall goals of the research, as well as their individual rights as an experiment subject. A detailed experiment script was used by the researcher as a means to ensure consistency for all subjects (Appendix D). For each individual, the experiment was conducted on two separate days.

Day One. On day one, the subjects completed the automated speech recognition software training procedures recommended by the software manufacturer. This training resulted in an individual user speech file for each subject. That user speech file was then used to complete the required tasks. The user speech file contains voice pattern templates unique to that specific user. This template is used to match subsequent voice commands and produce output text into the word processing software. Once the individual user speech file was created, the subjects were trained on the concepts behind the basic use of speech recognition software. Two practice tasks were used to familiarize the subjects with several commands that would be used for inputting and formatting text. The researcher, familiar with the software, was available to offer instruction, and answer any questions asked by the subjects.

The first practice task was an unformatted text selection of approximately 200 words (Appendix 2). As the subjects input the text using the speech recognition software, they were instructed on the methods and techniques used to correct mis-recognition errors. A mis-recognition error is an error that occurs when a user says one word and the software may interpret that word as a different, similar sounding word. For example, the user may say the word "speech", but the software mis-recognizes the word and enters that word as "peach", or "beach".

Once subjects completed entering the text for the first practice task, they were given a training aid with 20 basic commands used to format text using speech recognition software (Appendix 3). Using a script, (Appendix 4) the researcher proceeded with the subject through the use of all 20 commands. Each command was explained and practiced using the completed practice task until the subject felt comfortable with it.

Each subject then completed a second practice task that required him to use each of the 20 commands to complete the task that, unlike the first practice task, included various text, line, and paragraph formatting requirements (Appendix 5). This task was also not timed, and the researcher continued to be available for instruction and guidance.

Upon completion of the second task, the subjects were then given an opportunity to perform any additional practice they felt necessary. They were also offered the opportunity to discuss with the researcher any questions they had concerning the speech recognition software as well as the experiment in general. The subjects were then reminded of their scheduled time for the second day of experimentation. In addition, they were asked not to discuss the details of the experiment thus far with any other potential subjects. Once the subjects completed the practice tasks, they were debriefed and given an opportunity to ask questions and reminded not to discuss any details concerning the experiment to any potential subjects. Figure 1 presents the typical timeline for day one activities.

Task:	Intro	Speech file creation	Practice task #1	Practice task #2	Debrief
Minutes:	15	30	20	20	5

Figure 1: Day 1 Timeline

Day Two. In an effort to minimize any undesired memory effect between subjects, day two of experimentation was scheduled strictly from between 24 and 48 hours after day one. Additionally, in an effort to minimize any undesired effect that may have resulted from the order in which the subjects performed the tasks, each subject was randomly assigned an order in which to perform the tasks. Half of the subjects performed

the timed tasks, first using speech, then keyboard and mouse. The other half of the subjects performed the timed tasks, first using keyboard and mouse, then speech.

Upon entering the laboratory, the subjects were again reminded of their rights as a participant. In addition, they were also reminded that the goal of the research was to evaluate speech recognition software, not their individual ability to use the software. Before beginning the timed portion of the speech recognition task of the experiment, each subject was given a short task, similar to the second practice task from day one (Appendix 6). This task was completed in an effort to refresh the subjects' memory and further their understanding and expertise with speech recognition software. As on day one, the researcher was available for instruction and guidance for this refresher-training task.

Text selections for each of the timed tasks, while not identical, included the same number of words, grammatical structure and general reading level, as well as similar number of paragraph transitions and text formatting requirements (Appendices 7 and 8). Table 1 presents a comparison of the two tasks.

Table 1. Task Statistics

	Task 1	Task 2
Words	606	570
Characters	3156	3190
Paragraphs	12	13
Lines	54	59
Bold Functions	8	11
Underline Functions	8	7
Italic Functions	2	0
Hyphen functions	4	4
Hard Return Functions (equivalent to "enter-key")	16	16
Indent Functions (equivalent to " tab-key")	4	4

Regardless of the text input method order for the subjects, Task One was performed first, then Task Two. This ensured that, throughout the subject pool, each task was performed using both methods of text input. Before beginning each timed task, the subjects were given detailed instructions. They were then given the opportunity to ask questions and instructed that, once the task was begun and timing started, they would not be permitted to ask questions.

Once each timed task was complete and time was recorded, the file was saved in that subject's individual task directory. They were then given the opportunity, using the same text input mode (speech or keyboard) used to initially enter the text, to proofread the document and make any corrections required to make the document appear exactly as it did on the handout they were given. The proofreading and correction phase was also timed and recorded. Once the proofreading was complete, the document was saved as a different file name in the subject's individual task directory. The end result was a separate directory for each subject that included four files with .doc file extensions – two initial text entry files, and two proofread files. These files were closely compared to the original text handouts the subjects were given. Those subjects performed the first timed task using keyboard, and the second task using SRS. Once the subjects completed the tasks, they were debriefed and given an opportunity to ask questions and reminded not to discuss any details concerning the experiment to any potential subjects. Figure 2 presents the typical timeline for all odd numbered subjects. Those subjects performed the first timed task using SRS, and the second timed task using the keyboard. Figure 3 presents the typical timeline for all even numbered subjects.

Task:	SRS Refresher training	Timed task #1	Proof	Timed task #2	Proof	Debrief
Minutes:	15	10-20	5-10	10-20	5-10	5

Figure 2: Day 2 Timeline For Odd Numbered Subjects

Task:	Timed task #1	Proof	SRS Refresher training	Timed task #2	Proof	Debrief
Minutes:	10-20	5-10	15	10-20	5-10	5

Figure 3: Day 2 Timeline For Even Numbered Subjects

Statistical Analysis. The text entry, proofreading and error data were all collected using a replicated Latin Square design. Since each subject performed tasks using both methods of input, traditional analysis of variance (ANOVA) is inappropriate as the data collected is not completely random. That is, data for both treatments are being collected from each subject. In this experiment, for example, traditional ANOVA would only be appropriate if one treatment (keyboard or speech) was administered to each subject.

The Latin square design takes into consideration three possible experimental design effects. The first, and most obvious design effect is that of the actual experimental treatments "Keyboard" or "Speech". This is the design effect of real interest. The second design effect inherent in a Latin square design is the column effect. The column effect takes into account statistical differences caused by the time period for which each subject participated. For example, subject 30 may have been subject to biases or learning curve effects on the part of the researcher as a result of having already administered the experiment with 29 prior subjects. A statistically significant presence of this effect is undesired as it detracts from the primary focus of the treatment effects. The third

possible effect is the row effect. The row effect takes into account the order in which the subjects performed the tasks. For example, all odd numbered subjects performed speech tasks first, then keyboard tasks. While all even numbered subjects performed keyboard tasks first, then speech tasks. A statistically significant row effect is also undesired and may be explained by fatigue, attention span, and learning curve effects on the part of both the researcher and the subjects. Table 3 gives a representation of two replications of the experiment design. With 30 subjects, 15 replications of the design were analyzed.

Table 3. Two Latin Square Replications
(K=Keyboard, S=Speech)

Subj #	Treatment	
1	S	K
2	K	S
3	S	K
4	K	S

IV. Results

Assumptions Tests

As stated in Chapter 3, a replicated Latin Square design was used to collect the data in this experiment. Despite the use of a Latin Square design for data collection, the three standard assumptions for ANOVA hold true for the Latin Square design as well. The assumptions being independence, constant variance, and normality. These assumptions were tested using JMP, a popular statistics software package, with and without two apparent outliers, and were not deemed to be violated.

The normality assumption was tested by plotting residuals resulting from the ANOVA and generating a normality plot. In addition, the Shapiro-Wilks test statistic was used verify a normal distribution for all observations. Figure 4 presents a normality plot of the residuals for the initial text entry results. Normality plots were generated in this manner for all four observations (i.e. initial text entry times, proofreading times, initial errors, post-proofreading errors). Table 4 presents the Shapiro-Wilks test statistics for all observations (p-value (p) > .05).

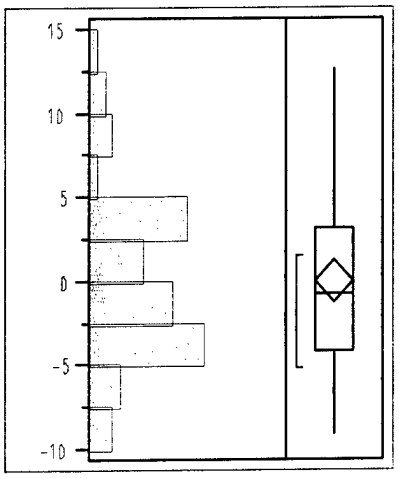

Figure 4: Distribution Plot of Residuals for Initial Text Entry Times

Table 4. Normality Assumption Test Statistic

Observation	Shapiro-Wilks Test Statistic ($p > .05$)
Initial Text Entry Times	.1039
Proofreading Times	.0997
Initial Text Entry Errors	.1342
Post-Proofreading Errors	.1098

To ensure that the data for all observations were equally varied for both treatments, the constant variance assumption was tested by plotting the observation residuals for each treatment (X by Y) and generating a test statistic. The Levene test statistic was used ($p > .05$) to determine equal variance. Figure 5 presents the X by Y residuals plot for initial text entry times. Table 5 presents the Levene test statistic ($p < .05$) for all observations.

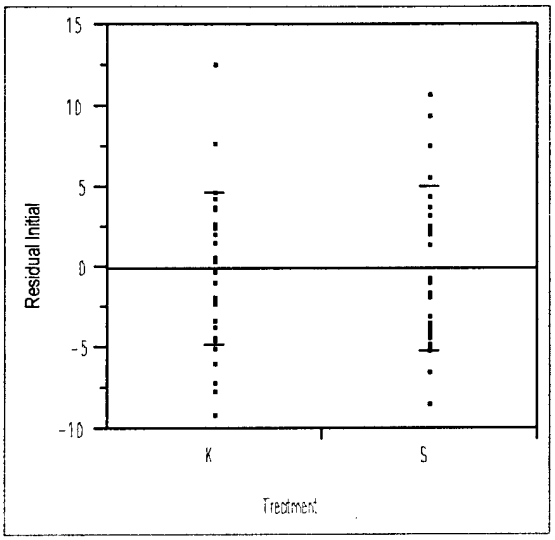

Figure 5: X by Y Plot of Residuals for Initial Text Entry Times

Table 5. Constant Variance Assumption Test Statistic

Observation	Levene Test Statistic ($p > .05$)
Initial Text Entry Times	.4523
Proofreading Times	.4561
Initial Text Entry Errors	.3318
Post-Proofreading Errors	.4021

To verify that the data were independent for all observation for both treatments, the data collected for each observation was plotted on a control chart. The charts were observed for evidence of correlation. In addition, the Durbin-Watson test statistic was used to determine whether or not the observations have first-order auto-correlation. Figure 6 presents a control chart for initial text entry times. Table 6 presents the Durbin-Watson test statistic ($p < .05$) for all observations.

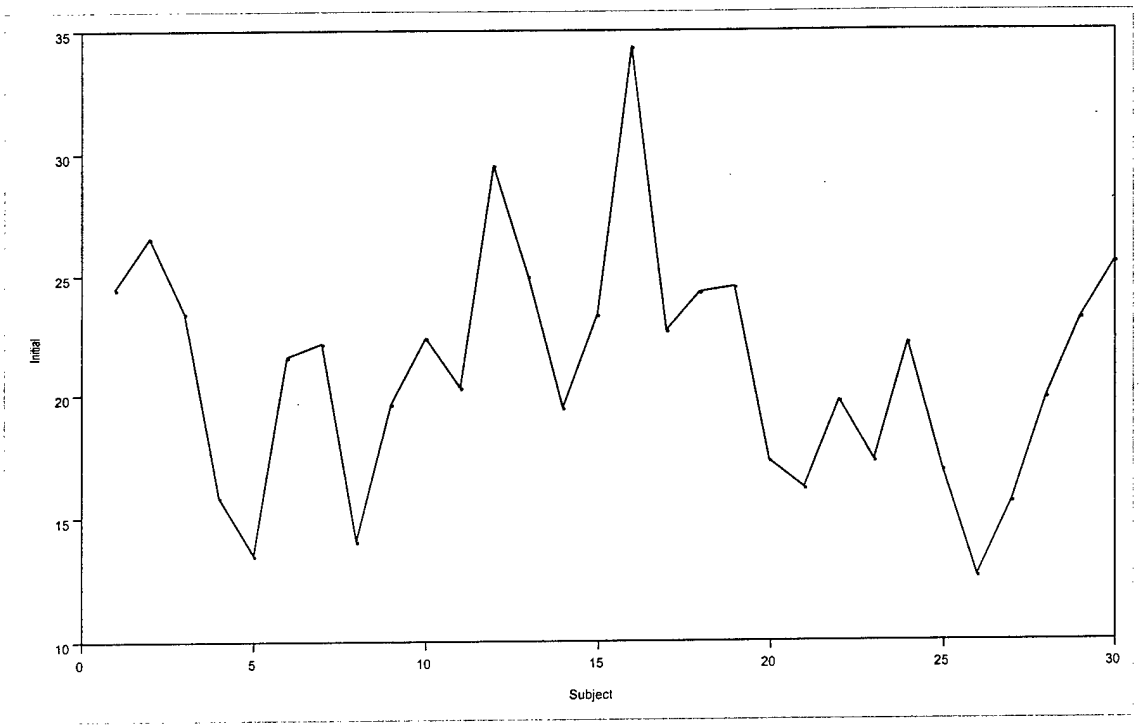

Figure 6: Control Chart for Initial Text Entry Observations

Table 6. Independence Assumption Test Statistic

Observation	Durbin-Watson Test Statistic ($p > .05$)
Initial Text Entry Times	.1165
Proofreading Times	.1047
Initial Text Entry Errors	.1003
Post-Proofreading Errors	.1184

Results

The results of the procedures described in the preceding chapter are reported here. The data were collected were reduced for analysis as appropriate for measurement. The data were separated and analyzed using ANOVA and results are reported in five categories:

1. Time for initial text entry
2. Time for proofreading

3. Total time for text processing
4. Errors after initial text entry
5. Errors after proofreading

For easier data analysis, all times were converted from MM:SS to minutes in decimal format.

Initial Text Entry Results

The primary motivation for this research is to determine the productivity implications of implementing speech recognition software in a text processing environment. Two of the primary components of text processing are text entry and proofreading/error correction. The primary question at hand for this portion of the experiment is: How much time is required complete a relatively simple text entry task using speech recognition software compared to a similar task using a keyboard? As indicated in Table 7, there is sufficient evidence to conclude that there is a statistically significant difference ($p < .05$) in the average time required to enter the text using the two experiment treatments, Speech and Keyboard. The column and row effects were not statistically significant. Furthermore, it took an average of 18.49 minutes to complete the initial text entry task using speech recognition software, compared to an average of 21.31 minutes using the keyboard. Figure 7 presents a pictorial representation of the means for the initial text entry treatments.

Table 7. Initial Text Entry Time F-Test Results

Source	F	P
Row	0.6224	0.4335
Column	0.0065	0.9362
Treatment (S and K)	4.0205	0.0498

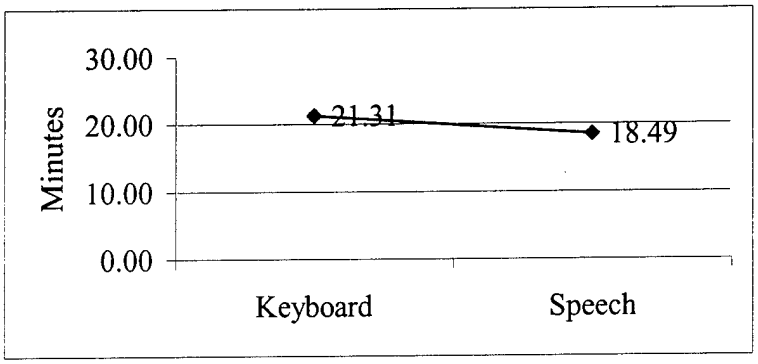

Figure 7. Initial Text Entry Time Means Comparison

Proofreading Results

The time required to initially input the text is an important aspect in determining the productivity implications of a text entry tool. However, since an error free text processing tool does not exist, document proofreading and error identification and correction is a key component of text processing. Therefore, it is equally important to measure the time required to identify and correct errors that will inevitably appear.

As indicated in Table 8, we can conclude that there is a statistically significant difference ($p < .05$) in the average time required to proofread the documents using the two experiment treatments, Speech and Keyboard. Again, the column and row effects were not statistically significant. Furthermore, it took an average of 9.31 minutes to proofread and correct the document using speech recognition software, compared to an

average of 4.46 minutes using the keyboard. Figure 8 presents a pictorial representation of the means for the treatments for document proofreading.

Table 8. Proofreading Time F-Test Results

Source	F	P
Row	0.4696	0.4960
Column	0.5297	0.4698
Treatment (S and K)	28.4953	<.0001

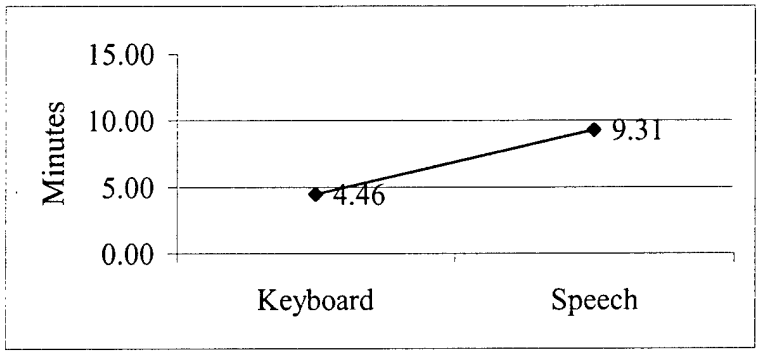

Figure 8. Proofreading Time Means Comparison

Total Text Processing Time Results

At this point, it is important to combine the two measures of time (initial text entry and proofreading/error correction) in an effort to measure the total time required to completely perform a simple text entry task. As indicated in Table 9, we can conclude that there is a statistically significant difference ($p < .05$) in the average total time required to enter text and proofread the documents using the two experiment treatments, Speech and Keyboard. In addition, the column and row effects were not statistically significant. Looking at the total time required to enter, proofread and correct the text in the document, it took an average of 27.80 minutes using speech recognition software, compared to an average of 25.39 minutes using the keyboard. Figure 9 presents a

pictorial representation of the means for the treatments for total document entry, proofreading and correction.

Table 9. Total Text Processing Time F-Test Results

Source	F	P
Row	0.8900	0.3495
Column	0.0993	0.7539
Treatment (S and K)	1.5685	0.2156

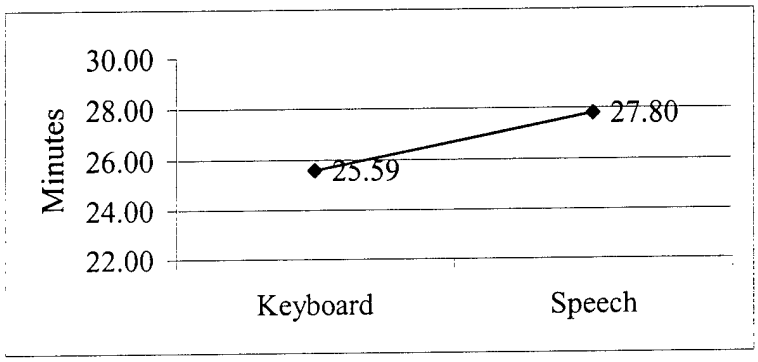

Figure 9. Total Text Processing Time Means Comparison

Initial Text Entry Error Results

Just as time required to perform a text entry task is a clear determinant of a text entry tool's productivity, so to is a count of the errors that result from its use. The question to be answered for this portion of the data analysis was: How many errors are produced using speech recognition software for initial text entry compared to a keyboard? As stated in Chapter 3, error rates were determined by evaluating the tasks performed by the subjects and determining any deviation from the original document. Errors included spacing, formatting, spelling, mis-recognition, and punctuation deviations.

As indicated in Table 10, we can conclude that there is a statistically significant difference ($p < .05$) in the average number of errors resulting from entering the text using

the two experiment treatments, Speech and Keyboard. The column and row effects were not statistically significant. Furthermore, when initially entering the text, subjects produced an average of 32.43 errors using speech recognition software compared to 13.90 errors using the keyboard. Figure 10 presents a pictorial representation of the error means for the treatments.

Table 10. Initial Text Entry Error F-Test Results

Source	F	P
Row	0.0004	0.9832
Column	1.8276	0.1818
Treatment (S and K)	34.4842	<.0001

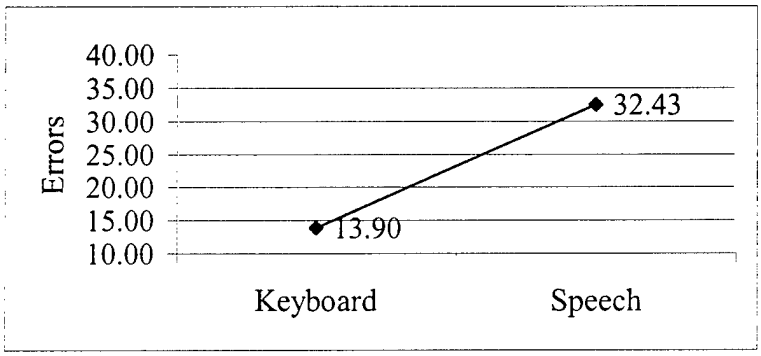

Figure 10. Initial Text Entry Error Means Comparison

Further analysis of the initial text entry errors reveals that, when dividing the number of accurate words produced by the subjects during the initial text entry portion of the experiment by the total number of accurate words in the original document they were given, an average accuracy rate of 98.39% was achieved when subjects used the keyboard. Similarly, a 96.28% accuracy rate was achieved when subjects used speech to enter text.

Proofreading Error Results

Just as the number of errors were counted after initial text entry was complete, so too was the number of errors that remained after proofreading. This is an important measure of a text processing tool's ability to allow the user identify and correct mistakes. As indicated in Table 10, there is sufficient evidence to conclude that there is a statistically significant difference ($p < .05$) in the average number of errors present after proofreading the documents using the two experiment treatments, Speech and Keyboard. Again, the column and row effects were not statistically significant. Finally, after proofreading the document, there was an average of 21.00 errors present when subjects used speech to identify and correct errors compared to an average of 8.37 errors when subjects used the keyboard. Figure 10 presents a pictorial representation of the means for the treatments.

Table 11. Proofreading Error F-Test Results

Source	F	P
Row	.0547	0.8160
Column	2.1272	0.1503
Treatment (S and K)	17.8051	<.0001

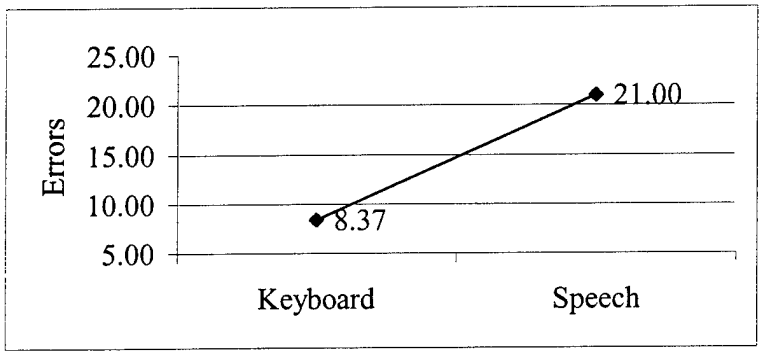

Figure 11. Proofreading Error Means Comparison

Further analysis of the post-proofreading errors reveals that, when dividing the number of accurate words produced by the subjects during the proofreading/error correction portion of the experiment by the total number of accurate words in the original document they were given, an average accuracy rate of 99.55% was achieved when subjects used the keyboard. Similarly, a 97.47% accuracy rate was achieved when subjects used speech recognition to enter text and correct errors. Additionally, further analysis reveals that, when dividing the number of errors present after proofreading by the number of errors remaining after proofreading, 48.96% of the initial errors were identified and successfully corrected when subjects used the keyboard. Similarly, when subjects used speech recognition software to correct errors, 37.76% of the initial errors were identified and successfully corrected.

Summary

Using these text processing productivity measurements, we can conclude that subjects completed the initial text entry portion of the task faster using speech recognition software compared to using the keyboard. However, when subjects used speech recognition software to enter text, significantly more errors were produced when compared to using the keyboard. Because of this, it took longer for subjects to complete the proofreading/error correction phase of the task using speech recognition software, resulting in a slightly longer time required to completely enter, proofread and correct the document using speech recognition software. Furthermore, fewer errors were identified and corrected using speech recognition software, resulting in a higher number of errors present after proofreading compared to the keyboard.

V. Discussion

General Discussion

This research was undertaken in an effort to assess productivity measures for text processing tasks involving conventional text input and correction methods using a keyboard and mouse compared to text input and correction using speech recognition software. To date, relatively little empirical research has been conducted comparing the productivity of speech recognition software with that of the keyboard and mouse. With continuing improvements in microprocessor technology and speech recognition algorithms, it is reasonable to assume that speech recognition software will continue to evolve in an attempt to provide a useful alternative to conventional text input devices.

The results of this research indicate that, for basic text entry tasks, a user with minimal training and experience with speech recognition software can perform a text entry task in a shorter time relative to a keyboard and mouse. An interesting finding in this research is that this can be achieved regardless of a user's general word processing proficiency or conventional typing speed. While users can achieve faster initial text entry speeds using speech recognition software, they may do so at a decreased accuracy level. In general, subjects using speech recognition software to input text and correct errors produced a higher level of errors than when using a keyboard and mouse. This higher error rate resulted in longer proofreading and error identification/correction times using speech recognition software. Despite an overall average accuracy rate of 97.47% after proofreading, this research demonstrated that more errors are produced and fewer errors are successfully identified and corrected using speech recognition software. It is

important to note that the subjects who participated in this experiment had no experience in the use of speech recognition software, yet still managed, for the most part, to realize similar initial text input speeds when compared to using a keyboard and mouse to input text.

The disparity in the subjects' training and experience with the two text input modes may help explain the greater number of errors produced when using speech recognition software. Based on responses to the subject questionnaire administered to each subject prior to their participation in this experiment (Appendix A), most of the subjects who participated in this study had extensive training and experience using the keyboard for text processing. In addition, most of the subjects used the keyboard in a text processing environment on a routine basis. On the other hand, the subjects had no experience whatsoever with speech recognition software for text processing. Their participation in this study was their first exposure to this method of text processing. As indicated in Chapter Three, they were given approximately 90 minutes of training using the speech recognition software, then were required to perform text processing tasks. This inexperience, relative with the keyboard, is most likely responsible for the higher error rates.

Another factor contributing to the greater number of errors using speech recognition software may be related to the subjects' user speech files. As discussed in Chapter Two, speech recognition systems are speaker-dependent. Each user of the software has a unique user speech file that was created on the first day of participation. Most speech recognition software packages, including the one used for this study, continue to update and refine these files after every use, thereby improving accuracy

(Rodman, 1999). Furthermore, the greater number of errors produced using speech recognition software for the initial text entry also accounts for the higher times required to proofread and correct the document. It is reasonable to assume that a mature user speech file, combined with increased user training, may result in an improved error rate as well as improved overall text processing times.

Implications

The fundamental problem approached by this research is that of determining what productivity benefits can be gained or lost by converting from a conventional text processing environment of keyboard and mouse to speech recognition software. Despite claims from speech recognition software developers that using speech recognition software will result in faster text processing times and increased accuracy rates relative to the use of a keyboard, to date no research has concluded that the implementation of speech software in a text processing environment will provide substantial productivity benefits, or lack thereof. The results of this research suggest mixed results toward determining any productivity benefits resulting from implementing speech recognition software for text processing.

The significance of potential productivity benefits resulting from the implementation of a new information technology is great. For practitioners, the findings in the area of error rates indicate that, prior to any implementation of speech recognition software in an office automation environment, a comprehensive training effort must be undertaken. This training would serve two purposes toward the ultimate goal of increased productivity. First, potential users of speech recognition software would have

the training and experience necessary to be proficient in its use. Second, increased exposure to and use of speech recognition software may result in a more mature user speech file, with increased capability to properly match output to user desires, thus reducing error rates.

Limitations

As with all research, there are a number of limitations inherent in this study. First, the majority of the participants in this study used for the study were primarily homogeneous with regard to their backgrounds and perspectives concerning office automation practices. Different results may be obtained with samples that have more varied backgrounds and perspectives. Also, the tasks performed in this experiment were basic in their content and format. More complex tasks involving figures, graphics and numbers may also provide different results. In addition, while this researcher has a general knowledge and understanding concerning the use of speech recognition software, a researcher with more extensive knowledge and expertise using speech recognition software may be able to develop a more comprehensive study that might provide different and more accurate results.

Suggestions for Future Research

Further research should be conducted to determine more long-range productivity impacts of implementing speech recognition software. With increased training and more experience, it is reasonable to assume that users of speech recognition software will become more proficient in its use, improving both the time required to perform text

processing tasks as well as accuracy rates. A longitudinal study to determine specific productivity implications for trained and experienced users of speech recognition software would go far in helping practitioners faced with the decision of implementing it. Results from this and future study involving speech recognition software can be used to compare the cost of purchasing the software and training workers to use it with the relative productivity gains in man hours they can expect to gain from its use.

Conclusion

The recent emergence and growing availability of speech recognition software for home and office automation environments is notable. However, it is clear that, prior to investing in a new information technology product, it is important to determine what productivity impacts will be realized from its implementation. This study indicates that, though more research is recommended to explore the long term productivity impacts of implementing speech recognition software, for relatively inexperienced users, speech recognition software currently provides no tangible productivity benefits over the traditional text processing modes of the keyboard and mouse.

Appendix A: Survey

Survey Day 1

Subject: _____

1. Name:_____

Daytime Phone:_____ E-Mial:_____

2. Sex (Circle 1): M or F

3. Age:_____

4. Rank (Check 1):
   - ☐ Airman (E1 to E4)
   - ☐ NCO (E5 and E6)
   - ☐ Sr. NCO (E7 to E9)
   - ☐ CGO (O1 to O3)
   - ☐ FGO (O4 to O6)
   - ☐ Civilian

5. Proficiency with MS Word (Check 1)
   - ☐ None
   - ☐ Novice
   - ☐ Intermediate
   - ☐ Advanced

6. Approximately, how many hours do you use MS Word per week? (Check 1):
   - ☐ None
   - ☐ 1 - 5 hours
   - ☐ 5 - 10 hours
   - ☐ 10 - 20 hours
   - ☐ More than 20 hours

7. Proficiency with speech recognition software (Check 1):
   - ☐ None
   - ☐ Novice
   - ☐ Intermediate
   - ☐ Advanced

8. Proficiency with Dragon NaturallySpeaking speech recognition software (Check 1):
   - ☐ None
   - ☐ Novice
   - ☐ Intermediate
   - ☐ Advanced

Appendix B: Demo Task 1

Recent advances in the area of low-cost speech recognition have moved the technology into everyday consumer products. With vast improvements in the quality and accuracy of cheap speech recognition systems, the value of adding speech recognition technology to everyday customer products is now being realized. As products become increasingly complicated and offer more functions, implementing speech recognition allows consumers to use products more intuitively while maximizing their functionality. Talking to our products and listening to what they say gives products a life of their own and significantly changes the way we can use them. One area which speech recognition will have a deep impact is voice dialing. This allows a consumer to dial a phone number simply by saying the name of the person they wish to call. Sensory Inc. first made its reputation as a company that made toys talk.

Appendix C: Training Aid

SPEECH RECOGNITION COMMANDS

SPEECH RECOGNITION COMMANDS

TYPE	#	*SPEECH COMMAND*	EXPLINATION OF COMMANDS
General Formatting	*1*	**New Line**	Adds one carriage return and does not automatically capitalized the next word you dictate.
	2	**New Paragraph**	Adds two carriage returns and capitalizes the next word you dictate.
	3	**Tab Key**	Moves the insertion point to the next tab stop.
	4	**Space Bar**	Adds one space where the cursor is positioned.
Text Editing	5	*Select* <text>	<text> means currently visible, contiguous words.
	6	*Select* <start-text> *Through* <end-text>	<start-text> means the first word you want to select. <end -text> means the last word you want to select.
	7	*Select* [*Character* or *Word* or *Line*]	Selects character, word, line or paragraph based on the position of the cursor.
	8	**Select Again**	Selects the next instance of the word currently selected.
	9	**Scratch That**	Deletes selected text or the last word you say. You can repeat *Scratch That* up to 10 times.
	10	*Delete That*	Deletes selected text, the next character after the cursor or the last word you said.
	1 1	*Delete* [*Next* or *Last*] # [*Character(s)* or *Word(s)*]	Deletes a specified number of characters or words forward from the cursor or back from the cursor.
	12	**Press Backspace**	Same functionality as pressing the "Backspace" the key.
	13	**Copy That**	Copies selected text.
	14	**Cut That**	Cuts selected text.
	15	**Paste That**	Pastes selected text.
	16	*Undo That*	Undoes last action.
Text Formatting	17	*Bold That*	Bolds selected text or the word you said last.
	18	*Underline That*	Underlines selected text or the word you said last.
	19	*Italicize That*	Italicizes selected text or the words you said last.
	20	*Cap That* or *No Caps That*	Capitalizes first letter of the text you selected or removes all capitalization of selected text.

Appendix D: Experiment Script

SPEECH RECOGNITION EXPERIMENT SCRIPT

DAY 1

Lab Set-up Check List Complete

PHASE I: INTRODUCTION

(Upon arrival, escort subject into SRS Lab, seat them at the computer and explain the experiment.)

Thank you for participating in this study. We are conducting research to evaluate how speech recognition software performs in comparison to conventional text input modes of keyboard and mouse. Before we get started, I'd like you to fill out this survey.

(After Subject completes survey)

As you know, you have been scheduled for two separate days of experimentation.

Today, day one of the experiment, you will train the speech recognition software to recognize your voice. Then, we will train you how to use the software.

On day two, we will review what we covered today, then you will perform timed tasks using both speech recognition software and keyboard and mouse. Keep in mind our purpose is to test the software, not your ability to use the software.

After the experiment on day two, you will also complete a survey asking about your perceptions of the speech recognition software.

Though your participation is greatly appreciated, you have the right to, at any time, terminate your participation in this experiment.

Do you have any medical conditions such as a head cold or sinus infection that might affect the way your voice normally sounds?

(Reschedule experiment time with subject if necessary)

Do you have any questions at this point?

(Answer any questions)

Now, go ahead and put on the headset and position microphone as outlined in the microphone positioning instructions on the screen.

The next few screens will calibrate the software volume to your voice.

Click "Next".

Read the instructions on the screen, then click "Start Test" and begin reading the text.

(After Beep)

Click "Next".

Click "Finish".

Click "Next".

Read the instructions on the screen, then click "Run Training Program".

Read the instructions on the screen, then click "Continue".

Read the instructions on the top of the active screen, then when you are ready to begin recording, click "Record" and begin dictating.

You will be dictating for about 30 minutes to train the software to learn your voice. If at anytime during training, you want to pause to ask questions, cough, drink water, or just take a break, click the pause button on the next screen.

Do you have any questions at this point?

(Answer any questions)

Select "Dogbert's Top Secret Management Handbook", then click "Train Now".

(After training is completed and window pops up)

Click "Finish".

(Ask if subject needs a break)

Now we will run an automated introductory training program in order to introduce you to the basic commands you will be using to operate the speech recognition software. Once this initial training is complete you will be given the opportunity to practice what you've learned.

Do you have any questions at this point?

(Answer any questions)

Click "View Quick Tour".

Maximize the screen by clicking the middle button on the top-right of the active window.

For the following several screens, simply read the instructions on the left of the screen. This will give you the basic idea of the topic being presented. Then click the "Play" button to get a demonstration of that topic. When complete, click the "Next" button.

When done reading click "Play". (Screen 2)

When complete, click the "Next" button twice.

When done reading click "Play". (Screen 4)

When complete, click the "Next" button three times.

Click "Play". (Screen 7)

Click "Next".

Click "Play". (Screen 8)

Click "Next" button three times.

Click "Play". (Screen 11)

Click "Next".

Click "Play". (Screen 12)

Click "Next".

Click "Play". (Screen 13)

Close the active window by clicking the "X" at the top right of the screen.

Click "Next".

Click "Finish".

We will now reboot the computer. Feel free to ask any questions and take a break if you like.

Reboot the Computer
Start Word
Start NaturalWord
Open appropriate user file.

PHASE II: HANDS ON TRAINING - SCRIPTED

You will now be given an opportunity to practice the principles of speech recognition software use that you just learned.

(Hand subject Demo Task 1)

Here is a short text selection. You can either use this document stand and position the document stand wherever it is most comfortable for you, or you can hold the document in your hand as you dictate.

Simply dictate this text into the document. You can correct any errors as you go along using the commands you learned during the tutorial. Remember, as you dictate, words may not appear on the screen right away. Just keep reading naturally, while maintaining an awareness of the words on the screen so you can notice any errors as they occur. Feel free to ask any questions, but before you do, be sure to turn off the microphone. Once you're finished you will be given a chance to proof read and correct this document.

Before you begin, remember a few basic concepts:

1. Remember to take your time and speak clearly, enunciating each word. For this portion of the experiment, you will not be timed.

2. Don't forget to dictate punctuation (DEMO – For example comma, don't forget to say "period" say the end of a sentence period.).

3. There may be occasions where, as you are dictating, the software enters a word you did not intend. For example, you may say the word "speech", but the computer might enter the word "peach". This is called a mis-recognition error. In this case you can correct this error by selecting the mis-recognized word using the "select" command, then dictating the correct word again.

4. Also, remember that if the incorrect words are entered, you can use the command "Scratch That" to remove the incorrect words. Then re-dictate the phrase, making sure to enunciate each word. You can keep repeating this command until all errors are corrected.

5 Lastly, the microphone may pick up my voice or background sound and write unwanted text on the screen. In the event that this happens, simply say "Scratch That" or select the unwanted word(s) with the mouse and say "Scratch That".

Once you start dictating, feel free to ask any questions. Be sure to turn off the microphone using the microphone icon on the tool bar.

Do you have any questions about specific commands at this point?

(Answer any questions)

Put back on the headset and turn on the microphone using the microphone icon on the tool bar, and begin dictating.

(When subject finishes)

Now you can check the document and correct any errors using voice commands. As you do this, we will coach you along. As we coach you, you do not need to turn off the microphone unless you need to ask a question. When correcting an error, there may be an occasion where the software just won't cooperate. When this happens, only try to correct the error three times. If the software fails to make the correction after three attempts, simply move on.

(Hand subject memory aid)

Here is a sheet containing some of the basic speech recognition commands you will be using for this experiment. The sheet is divided into four columns. The first two columns indicate the type and reference number of each command. The second column indicates the commands you actually say into microphone. Those commands are italicized. The commands in brackets you must say. The commands in parentheses are optional. You will perform each command until you are confident with them.

First we'll do some text formatting.

Using the mouse, position the cursor at the beginning of any sentence within the middle of the paragraph.

Insert a new line using command #1.

Position the cursor at the beginning of any sentence within the middle of the paragraph.

Insert a new paragraph using command #2. (After Subject performs the command) Notice the difference between command 1 and 2. Command 2 adds two carriage returns and command 1 only adds one carriage return.

Indent the first line of one of the paragraphs by using command #3. Position in the cursor at the beginning of any paragraph and say command #3.

Position the cursor between two words and practice command #4 a few times.

Now we will go through the text editing commands.

Select any single word in the document using command #5.
(Instruct the "two-word +" technique)

Now you can select a group of words using the "Select Through" command, #6.
Select the words "Talking, through Significantly". (Point the words out if necessary)

Now you can select a specific character, word, or line using command #7.
Position the cursor in the word "recognition". Now say "Select Character". Now say "Select Word". Now say "Select Line". (Quiz Subject if necessary)

Command #8 is Select Again. This command selects the next instance of the word currently selected. Select the word "recognition". Now select another instance of that word using command #8. Try this command two or three times. (Wait for Subject to complete the command two or three times) Notice how the software searches up from the bottom of the viewing area to select the next word.

Command #9 is Scratch That. We've practiced that already. Do you feel confident with that one?

Command #10, Delete That, is similar to scratch that. Simply select any word, or series of words, and say command #10, and it will be deleted. Try that now.

Command #10 is also good for the deleting extra spaces before or after words. Now create some extra spaces by using command #4 then use command #10 to delete the extra spaces you don't want.

Command #11 allows you do delete a number of characters or words in relation to the position to the cursor. Position the cursor somewhere within a paragraph, and delete the four previous words by saying "Delete last 4 words". Now delete the next 5 words by saying "Delete 5 Words". You can delete individual characters by saying "Characters" instead of "Words". Now try that.

Now, position the cursor somewhere between two words and try command #12. It's just like hitting the backspace key.

Commands #13, 14, and 15 allow you to copy, cut and paste text you selected. Go ahead and select any word on the document by using the select command. Say command #13 (copy). Position the cursor between any two words and say #15 (paste).

Now do the same procedure for command #14 (cut). Select any word on the document using the select command. Say command #14. Position the cursor between any two words and say command #15 (paste).

Command #16 is Undo That. That is just like hitting the undo button on the tool bar. It undoes the last command you dictated. Delete a word and then use command #16 to undo deletion.

Now we will move into text formatting commands.

Go ahead and select any word, or group of words, using the select command.

Now use commands #17, 18, and 19 to format them. (Coach Subject as necessary)

Once you've formatted a word, you can repeat the command to unformat the selected word. Try that.

These commands can also be used to format text before it is ever entered into the document. Using the mouse, position the cursor in the document immediately after the period at the end of any sentence. Insert a new line, then say command #17, notice how of the bold icon on the toolbar turns on. (Point to bold icon if necessary)

Now say "speech recognition". (Pause) now say command #17 again to turn the bold function off. Notice how the bold icon on the toolbar turns off.

Command #20 capitalizes the first letter of the selected word or words. Select a word and capitalize it by saying "Cap That". Try that.

To remove capitalization, select the words and say "No Caps That". Try that.

These commands can also be used to capitalize a phrase you just said. Now dictate the phrase "capitalize a phrase you just said". Now use command #20 to capitalize the first letter of every word in the phrase. Now turn capitalization off by saying "No Caps That".

Go-ahead and turn the microphone off.

Do you have any questions about specific commands at this point?

(Answer any questions)

PHASE II: HANDS ON TRAINING - COACHED

In this phase of the training, you will enter text exactly as it appears on the hand out that I will give you using your voice and mouse. Feel free to use the full functionality of the mouse to select words, phrases and navigate throughout the document; however, you are not allowed to use the keyboard at all. **Do not** use the mouse to perform any text editing or formatting commands, use speech commands instead. As you enter text, feel free to correct any errors as they occur. I will coach you on the use of the speech recognition commands you just learned as you progress through the hand out. Feel free to stop dictating and ask questions at any time, but remember to turn off the microphone.

Here is a copy of the text selection [HAND SUBJECT DEMO TASK 2] you will enter into the computer using Microsoft Word. Feel free to use the memory aid to help you recall the speech recognition commands you learned earlier.

Do you have any questions at this point?

(Answer any questions)

Remember there will be a slight delay from the time you begin speaking to the time the software starts writing the text to the screen, just keep talking naturally and clearly and the computer will catch up to you. Remember, when you are trying to correct a mis-recognition, only give the software three chances and move on.

When you have completed entering the text, turn off the microphone by clicking the microphone icon on the tool bar and say "finished".

Now go ahead and turn on the microphone using the microphone icon on the tool bar.

When you are ready, begin speaking.

(Coach subject as required through the task)

(When subject is finished)

Do you have any questions?

(Answer any questions)

This completes the formal part of the training session. You may now practice any of the things you learned today or, if you feel comfortable with the software, you may leave.

Do you wish to practice a little now or you feel comfortable with the software?

(Thank the subject and remind subject to refrain from discussing details of the experiment to any potential subjects.)

Appendix E: Demo Task 2

1.1.3.1 Microprocessors

The dramatic and continuing growth in the speed and power of microprocessors is a primary factor in the migration of **advanced speech** recognition technology from laboratories to real-world applications. Figure 1.1 provides a **dramatic example** of that growth for microprocessors. The advent of each new generation of chips has heralded the commercialization of a new, more advanced class of speech recognition systems and technology.

1.1.3.2 The Effects of Miniaturization

One measure of **progress** is the increasing number of components we can cram onto a silicon chip about the size of a fingernail. For signal-processing chips, the scale of integration is about 33 percent per year. At the same time, the speed of individual components is increasing about 20 percent each year.

Miniaturization of hardware is fostering the use of **speech** recognition in consumer products. As smaller systems become more powerful, they can support increasingly complex speech-recognition technology. Several small-vocabulary, chip based systems were introduced in the early 1990s.

Appendix F: Demo Task 3

1.1.3.3 Global Business

The world is growing smaller politically and economically. **International business ventures** that link professionals on opposite sides of the globe are becoming commonplace. This has spawned a need to establish 24-hour telecommunications capabilities. Some of these needs can be satisfied by hiring "*bilingual*" telephone operators and business professionals. That solution is not always necessary or affordable, and touch-tone technology is not widely available outside of **North America**.

1.2 Historical Overview

The first documented attempts to construct an automatic speech recognition system occurred long before the **digital computer** was invented. In the 1870s Alexander Graham Bell wanted to build a device that would make speech visible to hearing-impaired people. He ended up inventing the telephone. Many years later, a Hungarian scientist requested permission for a patent to develop an automatic transport system using the optical sound tracks of movie films. The soundtrack was to serve as a source of capturing the sound patterns of speech. The system would identify the sound sequences and print them out. The request for a patent was labeled unrealistic and denied.

Appendix G: Task 1

1.1.2 The Challenge

The success of *speech recognition* software resulted from 40 years of work that has produced technology capable of accurately processing spoken input containing sizable vocabularies. Despite those impressive achievements, *speech recognition* still has not reached its goal of developing systems capable of understanding virtually anything anyone says on any topic when they are speaking in a natural free-flowing style of speech and situated in almost any speaking environment, no matter how noisy, that is, to understand spoken language as well as humans can. This shortcoming may be surprising since, for many humans, understanding what other people say may seem to be a simple task. In fact, it is extremely complex and difficult. There are many reasons why.

1. Voluminous Data

Although it may seem as if we speak using a single tone, the quantity of data in the sound wave is overwhelming. Within the range of human hearing, speech sounds can span more than **20,000** frequencies.

- The time required to capture, digitize, and recognize frequency patterns for every fraction of a second of speech would overwhelm any PC on the market as well as most other computer systems.
- In order to recognize speech at a speed that is acceptable to users, the amount of data and the signal must be dramatically reduced. It is not necessary to manipulate all the data from the entire speech wave. Some excludable data are irrelevant to the recognition process while other pieces of data are redundant.
- The quantity of the data can be reduced further by taking samples from the signal rather than trying to process the entire waveform.

2. Sound Wave

The paucity of information in the speech sound wave may appear to contradict the preceding point, but it simply highlights the fact that speech is more than acoustic sound patterns. Spoken language interaction between people requires knowledge about **word meanings, communication patterns, and the world in general**. Words with widely different meanings and usage patterns may share the same sequence of sound patterns. These are sets of frequently occurring words that sound the same but are spelled differently. Using these systems, the meanings of words that affect the interpretation of utterances cannot be extracted from the sound stream alone.

Often, a grammar is required to assist in the process. A grammar that links appropriate words using distinctions would, for example, link words in a way that makes sense. Knowledge of the world would be needed to determine the correct meaning of each sentence. Similar examples requiring world knowledge that is currently unavailable to computers can be drawn from newspaper headlines. Fortunately, the inclusion of information beyond acoustic analysis of the sound stream is not needed for many simple applications. It is obvious, though, that the incorporation of such "**higher level**" knowledge into a speech recognition system would serve as a gateway to truly natural speech communication with machines.

3. Speech Flow

Since we speak in individual words and we "hear" what other people say as sequences of words, it seems reasonable to expect the speech sound wave to consist of words with clearly marked boundaries. Unfortunately, that is not at all the case. Speech is uttered as a continuous flow of sounds and even when words are spoken distinctly there are no inherent separations between them. This should not be surprising since we hear foreign languages as streams of sound unbroken by our recognition of distinct words. The same phenomenon occurs for unfamiliar

words and phrases in English. Once it moved beyond single word input, speech recognition was forced to address the problem of segmenting the speech stream into its component words.

Appendix H: Task 2

4. Variability

One person's voice and **speech patterns** can be entirely different from those of another person. Some elements of this diversity are physical. Each individual is unique, differing from others in the size and shape of their mouths, the length and width of their necks, and a range of other physical characteristics. Added to these anatomical variations are age, sex, regional dialect, health, and an individual's **personal style** of speech. Despite these differences, a recognition system must be able to accurately process the speech of anyone who is expected to use the speech system. The development of speaker modeling techniques has produced dramatic advances in handling inter-speaker variability. Technologies alone will not eliminate all of these issues. Resolution of a significant portion of speaker variability issues, including speaker training, vocabulary selection, and the human factors in application design, all affect the ability of a recognition system to handle inter-speaker variability. These concerns are the responsibility of application designers.

5. More Variability

Even a single speaker will exhibit variability. The sound pattern of a word changes when speakers whisper or shout, when they are angry or sad, and when they are tired or ill.

- Even when speaking **normally**, individual speakers rarely say a word the same way twice.

- In fact, variability is the basic characteristic of speech. When speaker variability is added to inter-speaker differences it becomes difficult to identify and extract critical, word identification information from the input.

- Speaker modeling techniques have been designed to extract common intra-speaker patterns of the variability and produce very high **speech** recognition accuracy.

6. Noise

Natural speaking environments bombard the speaker with sounds of varying wildness emanating from many sources. They include people speaking in the background, street sounds, the slam of a door, music, and the loud noise of machinery. Sometimes the noise in a speaking environment can be so great that people cannot understand each other. As speech recognition is embedded in more diverse products and systems, the spectrum of noises will also grow. **Unfortunately, the challenging speaking environments are the ones that most characterize our daily living: busy offices, factories, loading docks, airports, automobiles**, and even our own homes. Background noise is not the only intrusion speech recognition systems must combat.

They must handle noise produced by the input device, sounds made by the speaker, such as lip smacks, and non-communication vocal limitations made by the speaker. Speech recognition over telephones is becoming increasingly popular, but it is one of the most challenging of speaking channels. Even people have trouble with it. Voices can be faint or full of static, but when everything is functioning well, it may still be difficult to distinguish between similar sounding words and sounds. The success of speech

recognition over the telephone illustrates the recent progress that has been made in this area. As with other issues, the role of the application developer in addressing noise has a strong impact on the ultimate success of the speech recognition application. The **rapid technological** advances of the last 15 years have come far toward achieving those goals, but the challenge should not be underestimated.

1.1.3 Driving Forces

Speech recognition has only recently achieved a level of reliability and flexibility to attract the interest of business and consumers.

Its achievements are due, in part, to significant technological advances within the industry. **Equally important** are external factors that have functioned as driving forces for speech recognition.

BIBLIOGRAPHY

Ainsworth, William A. Speech Recognition by Machine. New York: INSPEC, Inc., (1988).

Baatz, E. B. "Altered Stats," CIO, 8(2), (1994).

Ball, L., & Harris, R. "SMIS Members: A membership analysis," MIS Quarterly, 6(1), (1982).

Bakos, J. Yannis. "Are computers boosting productivity?," ComputerWorld, (March 1995)

Berndt, E. R., & Morrison, C. J. "Computers Aren't Pulling Their Weight," Computerworld, (Februray 1991).

Brynjolfsson, E., "The Productivity Paradox of Information Technology," Communications of the ACM, 36(12), 1993.

Brynjolfsson, E., Hitt, Lorin, "Paradox Lost? Firm-level Evidence on the Returns to Information Systems Spending," Management Science, 42(4), 1996.

Brancheau, J. C., & Wetherbe, J. C. "Key Issues in Information Systems Management," MIS Quarterly, 11(1), 23-45 (1987).

Crowston, K., & Treacy, M. E. "Assessing the Impact of Information Technology on Enterprise Level Performance," Paper presented at the Seventh International Conference on Information Systems (ICIS), (1996).

Dickson, G. W., Leitheiser, R. L., Nechis, M., and Wetherbe, J. C. "Key Information System Issues for the 1980s," MIS Quarterly, 8(3), 1984.

Hartog, C., and Herbert, M. "Opinion Survey of MIS Managers: Key Issues," MIS Quarterly, 10(4), 1986.

Hitt, Lorin and Brynjolfsson, Erik. "Productivity, Business Profitability, and Consumer Surplus: Three Different Measures of Information Technology Value," MIS Quarterly, June 1996.

Katz, Rudolph. "Measuring Technology's Business Value," Information Systems Management, Winter 1993.

King, J. "Hello, Help Desk? Help!,". Computerworld, (November 1991).

Klevans, Richard L and Rodman, Robert D. Voice Recognition. New York: Artech House, Incorporated, 1997.

Koerner, Michael. Speech Recognition: The Future Now!. New York: Prentice Hall, 1996.

Laface, Pietro and De Mori, Renato. Speech Recognition and Understanding: Recent Advances, Trends, and Applications. New York: Springer-Varlag, 1992.

Lorek, L.A., "Speech-recognition programs becoming more reliable," Strategic Information Technology Management, 1997.

Maglitta, J. "Executives: Where's the IS payback?," Computerworld, (October 1993).

Marion, L. "The Trouble with Techies," CFO: The Magazine for Senior Financial Executives, 8(7) (1992).

Markowitz, Judith A. Using Speech Recognition: A Guide for Application Developers. New York: Prentice Hall, 1996.

McLean, E. R., Kappelman, L. A., & Thompson, J. P. "Converging End-User and Corporate Computing," Communications of the ACM, 36(12),, (1993).

Niederman, F. and Brancheau, J. C. "Information Systems Management Issues for the 1990s," MIS Quarterly, 15(4) (1991).

Pettey, M., and Shneiderman, B. "Speech Versus Mouse Commands for Word Processing: an Empirical Evaluation," International Journal of Man-Machine Studies, (1993).

Rodman, Robert D. Computer Speech Technology. New York: Artech House, 1999.

Rochester, J. B., and Douglass, D. P. (1991). "Critical Issues in Information Systems Management," U, 29(1): 1-16. 1995.

Rubio, Antonio J. and others. Speech Recognition and Coding. New York: Springer-Verlag, 1995.

Schumann, Matthias. "Methods of Quantifying the Value of Office Automation," MIS Economics, Fall 1989.

Scudder, R. A., and Kucic, A. R. Productivity Measures for Information Systems. Information & Management, 20(5) (1991).

Wilson, D. "Assessing the Impact of Information Technology on Organizational Performance," Strategic Information Technology Management, 1993.

Vita

Lieutenant Abell was born in Silver Springs Maryland on March 17, 1970. He graduated from Courtland High School in Fredericksburg Virginia in 1988. Upon completion of his Associate of Arts Degree from Lake-Sumter Community College in Leesburg Florida in 1990, he enlisted in the U.S. Air Force. He was first assigned as a Communications Security Accountant at the 74th Air Control Squadron, Langley Air Force Base, Virginia. There he completed his undergraduate studies in Business Administration at Saint Leo College. In 1993, Lieutenant Abell was reassigned as a systems analyst for the Defense Information Systems Agency at the Pentagon in Arlington Virginia. Lieutenant Abell received his commission in the U.S. Air Force upon completion of Officer Training School in 1996. Prior to his assignment to AFIT, Lieutenant Abell served as a Software Test Director for the 93rd Communications Systems Squadron at Robins Air Force Base. Upon graduation from AFIT, Lieutenant Abell will be assigned to the Air Intelligence Agency at Kelly Air Force Base in San Antonio Texas.

Lieutenant Abell is married to the former Michelle Gilbertson of Woodbridge Virginia. They have one five year old son, Zachary, and one on the way.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	December 1999	Masters Thesis

4. TITLE AND SUBTITLE
IMPLEMENTATION OF SPEECH RECOGNITION SOFTWARE FOR TEXT PROCESSING: AN EXPLORATORY ANALYSIS

5. FUNDING NUMBERS

6. AUTHOR(S)
Sean P. Abell, 1Lt, USAF

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Air Force Institute of Technology
2950 P. Street
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER
AFIT/GIR/LAS/99D-1

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
N/A

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
Michael G. Morris, Major, USAF, DSN 785-3636, ext 4578, michael.morris@afit.af.mil

12a. DISTRIBUTION AVAILABILITY STATEMENT
Approved for public release, distribution unlimited

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

The rationale behind implementing new information technologies is often to gain productivity improvements associated with the substitution of machinery for labor. However, the literature shows little direct evidence of a positive relationship between information technology investment and subsequent productivity benefits. This thesis reports on the examination into the productivity implications of implementing speech recognition software in a text-processing environment. More specifically, research was conducted to compare text processing speeds and error rates using speech recognition software versus the keyboard and mouse. Of interest was the time required to input and proofread text processing tasks as well as the number of errors generated using both methods of text input. The empirical data offer somewhat mixed results. While users initially entered text faster using speech recognition software ($p < .05$), they generated more errors and consequently performed proofreading and error corrections slower using speech. These results suggest that, in terms of accurate text processing, speech recognition software is still not a practical alternative to the keyboard. Therefore, implementation of speech recognition software is unlikely to result in any gains in productivity that would serve to justify its cost.

14. SUBJECT TERMS	15. NUMBER OF PAGES
Productivity, speech recognition, text processing, error rates.	74
	16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL