Multimodal Interactive Error Recovery for Non-Conversational Speech User Interfaces

Zur Erlangung des akademischen Grades eines
Doktors der Ingenieurwissenschaften
der Fakultät für Informatik
an der Universität Karlsruhe (Technische Hochschule)
vorgelegte

Dissertation

von

Bernhard Suhm

aus Gengenbach

Tag der mündlichen Prüfung: 17. Juli, 1998
Referent: Prof. Dr. A. Waibel
Koreferent: Prof. Dr. Brad Myers

Acknowledgments

My thanks extend to the two universities at which this dissertation work was performed: Karlsruhe University (Germany) and Carnegie Mellon University (Pittsburgh, USA). Being able to pool expertise from faculty specialized in related fields, and get advice when needed, makes the difference in world-class research environments, and are prerequisites for interdisciplinary research such as this dissertation.

I want to thank my advisor Alex Waibel for years of support and guidance. In spite of all the pressures that leading two research groups on two different continents entail, he provided a challenging work environment, precious time for discussions, and exceptional opportunities for my professional and personal development. I am sincerely thankful to my co-advisor Brad Myers. He provided a valuable alternative perspective and support. I would like to thank also Herb Simon and Bonnie John for introducing me to the foundations of Human Computer Interaction (HCI); and Mark Fichman, Bernd Bruegge, Bonnie John, Al Corbett and Suresh Bhavnani for useful advice and discussions on experimental design and statistical analyses.

Numerous colleagues at the Interactive Systems Laboratories, both at CMU and at Karlsruhe University, provided valuable help and facilitated the day-to-day work in many ways: by being good team members and friends. The following people have made this dissertation work possible by providing the necessary recognition components, and by advising me how to use and modify them: Herman Hild, Monika Woszczyna, Michael Finke, Ivica Rogina, Klaus Ries, Stefan Manke. I further extend my thanks to Arthur McNair for being a knowledgeable and always friendly system adminstrator, and for providing important ground-work of my dissertation; Torsten Zeppenfeld, Markus Bauer, Weyi Yang, and Frank Dreilich for their knowledge, help, friendliness as system administrators and co-workers. I thank the administrative staff both at CMU and Karlsruhe University for effectively keeping my head virtually free of administrative problems. In particular, I thank Silke Dannenmeier, Debbie Clement, Sharon Burks, Radha Rao, and Betsy Herk. Special thanks to Theona Stefanis, Lin Chase, Julia Deems, and Pauletta Pan for carefully proofreading my thesis. (Any remaining mistakes are, of course, my sole responsibility.) My gratitude also extends to many other co-workers and friends who provided help, support, and friendship: Jie Yang, Rob Malkin, Ralph Gross, Lin Chase, Mathias Denecke, Klaus Riess, Tanja Schultz, Martin Westphal, Ravishankar Mosur, and Kristie Seymore. Further thanks to all the participants in my user evaluations for their patience.

My profound thanks and appreciation go to the most influential person throughout my years at Carnegie Mellon University Suresh Bhavnani. A significant influence and sounding board for my work as well my personal development, Suresh is to me both a model for me for Andrew Carnegie's famous motto "my heart is in the work", and he was a true friend. I wouldn't have completed this work without the support of him, and a wonderful circle of other friends. I especially miss Theona Stefanis and Dirk Langer, and my friends from the church of Ascension who became like a second family for me.

Lastly, I am deeply grateful to my parents for all their support and love.

This research was sponsored by the Defense Advanced Research Projects Agency (DARPA) under Contract No. F33615-93-1-1330. The United States Government is authorized to reproduce and distribute reprints for Government purposes, notwithstanding any copyright notation thereon. The views and conclusions contained in this document are those of the author(s)and should not be interpreted as necessarily representing the officialpolicies or endorsements, either express or implied, of DARPA or the USGovernment.

Abstract

Performance of automatic speech recognition has significantly improved in recent years. Much of this due to significant progress made in speech recognition algorithms. Despite these algorithmic improvements, automatic speech recognition continues to be error-prone: non-interactive (algorithmic) approaches are unlikely to eliminate all recognition errors. Currently available correction methods for speech recognition - such as correction by respeaking, choosing from alternative words, or typing - are either inefficient or they require keyboard input. The theoretic upper bound on throughput of an automatic dictation system is equivalent to the average speaking rate of 150 wpm (words per minute). However, if correction is limited to current keyboard-free correction methods (i.e., respeaking and choosing from alternative words), this throughput is drastically reduced to an estimated 20 wpm. The possible productivity gain by using speech recognition technology is thus lost due to inefficient error correction.

This dissertation presents an interactive *multimodal* approach for efficient error correction without keyboard input in non-conversational speech recognition applications that employ graphic user interfaces. The approach presented in this dissertation improves efficiency of correction in two ways: First, by switching input modalities for correction and, second, by correlating correction input with the context of a repair. For example, on text and data input tasks, the user switches modality from continuous speech to spelling or handwriting, and performs simple editing tasks, such as deleting items and positioning the cursor, using intuitive gestures drawn on a touch-sensitive display. Correlating correction input with repair context increases correction accuracies (success rate) from 60-70% to 80-90%, compensating losses in accuracy due to the difficulty of recognizing correction input.

As a first step towards formalizing multimodal recognition-based interaction, this dissertation presents a performance model of multimodal human-computer interaction. The model predicts interaction throughput based on standard performance parameters of recognition tech-

nology and modalities, abstracting from current recognition technology and interface implementation. Applied to interactive error correction, it predicts correction speeds as a function of modality and recognition performance. For example, correction by handwriting will always be slower than keyboard input, unless application constraints slow down keyboard input (e.g., on small hand-held devices), or unless partial input is sufficient to convey the necessary information (e.g., disambiguating lists of alternative words based on the first one or two letters). Furthermore, the model predicts that speech correction would be faster than correction by fast unskilled typing on text input tasks, if speech corrections were recognized with approximately 70% accuracy (across multiple correction attempts when necessary).

To empirically evaluate the efficiency of multimodal correction in a potentially useful application, a prototype multimodal dictation system was built, which integrates interactive multimodal correction with a state-of-the-art, large-vocabulary continuous speech recognizer. User studies compared multimodal methods with conventional keyboard- and mouse-based correction methods on a dictation task. Results show that without use of a keyboard, text input rates of more than 40 wpm are feasible, assuming 90% accurate recognition of dictation input in realtime. This rate compares favorably to fast, non-secretarial typing[1]. This research thus confirms the hypothesis that switching modality speeds up error correction in speech recognition applications. However, typing remains the most efficient correction method in text input tasks for users with good typing skills (i.e., more than 40 wpm typing speed). Furthermore, the study shows that correction accuracy determines user preferences between correction modality. Although users initially prefer speech for corrections, they learn with experience to prefer the most accurate modality.

In summary, this dissertation shows how multimodal error correction can solve the problem of recognition errors in non-conversational speech recognition applications, and takes a first step towards a framework for multimodal recognition-based interaction.

1. Much higher input rates claimed by some vendors of commercial dictation systems either do not include the time necessary for corrections, or are based on optimistic performance assumptions.

Kurzfassung (Summary in German)

Die Erkennungsleistung automatischer Spracherkennungssysteme konnte in den vergangenen Jahren vornehmlich durch algorithmische Verbesserungen erheblich gesteigert werden. Aber trotz aller Verbesserungen, auch zukünftiger, können Erkennungsfehler nicht ausgeschlossen werden. Die vorliegende Dissertation untersucht interaktive multimodale Korrekturverfahren, mit denen ein Benutzer Erkennungsfehler in sprachgesteuerten Benutzeroberflächen schnell und elegant ohne Tastatureingabe korrigieren kann. Bisherige Verfahren zur interaktiven Fehlerkorrektur beschränken sich auf gesprochene Wiederholung, Auswahl aus einer Liste von Wort-Alternativen, und Korrektur durch Tastatureingabe. Unsere Benutzerstudien zeigen, daß die ersten beiden Verfahren in Anwendungen mit kontinuierlicher Sprache ineffizient sind. Mit Korrektur per Tastatureingabe fällt man auf die Modalität zurück, die man mit Spracherkennungstechnologie eigentlich ablösen wollte. Obwohl die durchschnittliche Sprechrate von 150 Wörtern pro Minute eine sehr hohe obere Schranke für die Produktivität eines automatischen Diktiersystems darstellt, messen wir nur ca. 20 Wörter pro Minute Durchsatz mit herkömmlichen tastaturlosen Korrekturmethoden (d.h. gesprochene Wiederholung oder Auswahl aus Alternativen). Der mögliche Produktivitätsgewinn durch den Einsatz von Spracherkennung ging also durch ineffiziente Fehlerkorrektur verloren.

Diese Dissertation stellt *multimodale* Fehlerkorrektur vor. Der multimodale Ansatz macht tastaturlose Korrektur effizient, weil die Eingabemodalität für Korrekturen gewechselt werden kann, und weil Algorithmen entwickelt wurden, die die Korrekturgenauigkeit durch Korrelieren der Korrektureingabe mit Kontextinformation signifikant erhöhen. Zum Beispiel kann die Modalität von kontinuierlicher Sprache zu Buchstabieren oder Handschrift gewechselt werden, sowie zu intuitiven graphischen Zeichen für einfache Editieraufgaben, wie z.B. Löschen von Wörtern oder Positionieren des Cursors. Durch Korrelieren der Korrektureingabe mit dem Kontext werden Korrekturgenauigkeiten von 80-90% erreicht, obwohl das Erkennen von Korrektureingaben deutlich schwieriger ist als in der Literatur bekannte Bench-

marks.

Zur Formalisierung multimodaler, automatisch verarbeiteter Eingabe wurde ein Performanzmodell multimodaler Mensch-Maschine Kommunikation entwickelt. Mit diesem Modell können Fehlerkorrekturgeschwindigkeiten vorhergesagt werden, in Abhangigkeit von der Performanz momentan verfügbarer automatischer Erkennungssysteme. Dieses Modell sagt vorher, daß gesprochene Korrekturen mindestens 70% genau erkannt werden müssen, um schneller zu sein als Korrektur durch Tastatureingabe für Benutzer mit hohen Tippgeschwindigkeiten. Handschriftliche Korrekturen durch Wiederholung des ganzen Wortes sind langsamer als Korrektur durch Tastatureingabe, selbst wenn perfekte Erkennungsgenauigkeit von 100% möglich wäre.

Um die Effizienz multimodaler Korrektur empirisch evaluieren zu können, wurde ein prototypischer multimodaler Texteditor mit Hilfe eines Spracherkenners für große Vokabularien entwickelt. Benutzerstudien wurden durchgeführt, die tastaturlose multimodale Korrektur mit herkömmlichen Korrekturmethoden vergleichen. Die Ergebnisse bestätigen die von anderen Forschern aufgestellte (aber nie überprüfte) Hypothese, daß multimodale Fehlerkorretur effizienter als unimodale Korrektur (durch gesprochene Wiederholung) ist. Tastatureingabe bleibt die schnellste Korrekturmethode für Benutzer, die gut tippen können (d.h., mehr als 40 Wörter pro Minute). Mit multimodaler Fehlerkorrektur und einer automatischer Erkennung von diktierten Eingaben mit 90% Wortakkuratheit in Echtzeit ist Texteingabe ohne Tippen mit Eingabegeschindigkeiten von 40-50 Wörtern pro Minute möglich, einschließlich der zur Fehlerkorrektur benötigten Zeit. Diese Eingabegeschindigkeit ist mit schnellem Tippen vergleichbar. Ferner zeigen unsere Studien, daß die Erkennungsgenauigkeit entscheidend die Wahl der Eingabemodalität beeinflußt - obwohl die meisten Benutzer Sprache bevorzugen, wenn sie genauso zuverlässig wie andere Modalitäten erkannt werden könnte. Zusammenfassend kann man sagen, diese Dissertation zeigt auf, wie mit multimodalen Eingabetechniken Tastatureingabe als bevorzugtes Eingabemedium in nicht dialog-orientierten Spracherkennungsanwendungen ablöst werden kann.

Table of Contents

List of Figures

List of Tables

1. Introduction

Speech recognition is an orthographic transcription of digitally recorded spoken utterances. In simpler terms, it is the process of converting an acoustic waveform into a sequence of hypothesized words [Tucker 1997]

Why has speech recognition technology captured the attention of computer engineers for more than 30 years? Some researchers view speech recognition as technology that enables computers to recognize existing forms of human expression (e.g., [Trubitt 1990]). A more pragmatic view of speech recognition is as an alternative to traditional computer input modalities, such as typing and pointing devices. Additionally, speech recognition can provide access to powerful technologies for people who are unable to use traditional input devices, and it can lead to new applications and uses which otherwise may prove to be infeasible.

Is the technology ready to live up to these promises? Over the last decade, there has been tremendous progress in speech recognition technology, speech recognition appeared to be on the verge of significant commercial breakthroughs more than once. Recently, more and more successful products have been introduced (e.g., automatic dictation systems by Dragon Inc. and IBM, and automated call centers by Nuance Communications, Inc. and SpeechWorks International). Nevertheless, speech recognition applications are yet to become common use among the general public. An examination of the reasons why speech recognition technology's failed to attract the public's attention reveals, among other things, persistent inadequate performance of many of its envisioned applications.

This dissertation addresses the problem of error correction in speech recognition applications. As a framework for discussing this problem across different applications, Section 1.1 intro-

duces a taxonomy of speech recognition applications and an overview of multimodal interfaces that integrate speech recognition with other forms (modalities) of communication. Section 1.2 outlines some of the limitations of current speech recognition technology, such as the problem of repair and error correction. Section 1.3 provides a broad overview of the subject, laying out design dimensions for speech recognition applications, and discussing design options for error correction across different task categories. Non-conversational applications with graphic user interfaces are introduced as the main focus of this dissertation work.

Drawbacks of current non-conversational repair methods are discussed in Section 1.4. There are three main non-conversational repair methods: repeating input using continuous speech (in this dissertation frequently called *respeaking)*, choosing from a list of alternative words, and using the keyboard and mouse. It is argued that respeaking and choosing from alternatives are ineffective in continuous speech applications, and that typing defeats the purpose of using speech input as an alternative to keyboard input.

The research question of this dissertation is presented in Section 1.5: given the low reliability of current speech recognition technology, how can users' efforts to recover from interpretation errors be minimized? Section 1.6 discusses the research question in the context of dictation systems, the example application chosen for this dissertation. Section 1.7 concludes the chapter with a summary of the dissertation.

1.1 Overview of Speech Recognition Applications

Before discussing error correction in speech recognition applications, one must first understand how speech recognition technology can be used. This section proposes a taxonomy of speech recognition applications as a framework for application-oriented issues such as error correction.

The second part of this section introduces the notion of multimodal interfaces and discusses the role of speech recognition within multimodal systems. The rationale for a multimodal

interface is based on the fact that people communicate with each other using many different modalities. Assuming that the approximation of normal human communication facilitates human-computer interaction, the decision to incorporate different modalities in a speech interface is logical. Therefore, many of the applications mentioned in the taxonomy are like-wise multimodal.

1.1.1 A Task-Oriented Taxonomy of Speech Recognition Applications

Even though the range of possible applications of speech recognition technology is still being explored, a preliminary taxonomy of speech recognition applications is useful as a conceptual framework to discuss application-oriented issues. The taxonomy focuses on tasks (i.e., jobs that users want to get done) that can be supported by speech recognition technology, rather than applications themselves. An application can involve more than one task. For example, an automatic service to rent a car involves both natural spoken dialogue (to lead the general dialogue with the customer) and form filling (to obtain the information necessary to complete a rental). Different published speech recognition applications illustrate to the different task categories. To avoid ambiguities that arise where an application involves more than one task, the main task determines an application's category in Figure 1-1 below.

The taxonomy divides tasks on the top-level of speech recognition applications into interactive and non-interactive tasks. In *non-interactive tasks*, the speech input does not originate in direct user interaction is processed. In this case, the speaker does not intend his or her speech input to trigger an action by the computer system within the application. Examples for such tasks include automatic transcription of speech (e.g., in a courtroom) and automatic indexing of speech data (e.g., of radio and TV broadcasts).

By contrast, in *interactive tasks,* speech originates in direct user interaction. In this case, the speaker expects the application to trigger some action as a result of the speech act. An example is automated directory assistance.

Interactive tasks can be further subdivided into *mediation of human-human communication* (e.g., a translation aid for foreign travel, teleconferencing tools, support for collaborative work) and *human-computer interaction* (e.g., to access a service or some functionality offered by a computer system). A user can pursue many goals in interacting with a computer, for example, entertainment, system task performance (shown in the figure as "command & control"), transaction or information retrieval ("transactions&queries"), data entry and manipulation ("data entry & manipulation"), and other tasks.

Examples for *entertainment* include new interactive games, computer animation (e.g., the recent popular movie "Toy Story"), and interactive TV. In command and control tasks, the user wants to initiate some action or control some process. In *command tasks*, the user issues concise commands to the system, typically single words or short phrases. Examples include controlling a robot via voice, or applications offering voice equivalents to menu and button interactions. As an example for *control tasks*, novel security systems may control the access to buildings or services using multiple channels. In *transaction and query tasks*, the user engages in a spoken natural language dialogue with a dedicated device to access some service. For instance, standard telephone services such as directory assistance and call routing are increasingly automated using speech recognition technology, as are call centers of many companies. Another important future application domain in this category includes services related to travel, such as scheduling inquiries for different means of transportation (rental cars, buses, trains, flights), booking of accommodations, and navigational support in foreign locales. In *data-entry and manipulation tasks*, the user creates and manipulates data that is stored in machine-readable form. According to the complexity of the data, two subcategories are defined. *Simple data entry* deals with isolated words, digits, or short phrases (as in form filling, personal assistants for addresses, and note-taking). *Text and Multimedia entry* tasks support the production (or composition) of text, and multimedia in general. Dictation systems fall into this category, as do Web authoring tools, and, in a more general sense, user interface design tools. Other tasks where multimodal interfaces are actively researched include smart

rooms (e.g., ALIVE [Casey, Gardner et al. 1995] at MIT's media lab), education (e.g., the Listen project [Mostow, Roth et al. 1994] at Carnegie Mellon University), and wearable computing (e.g., [Rudnicky, Reed et al. 1996]).

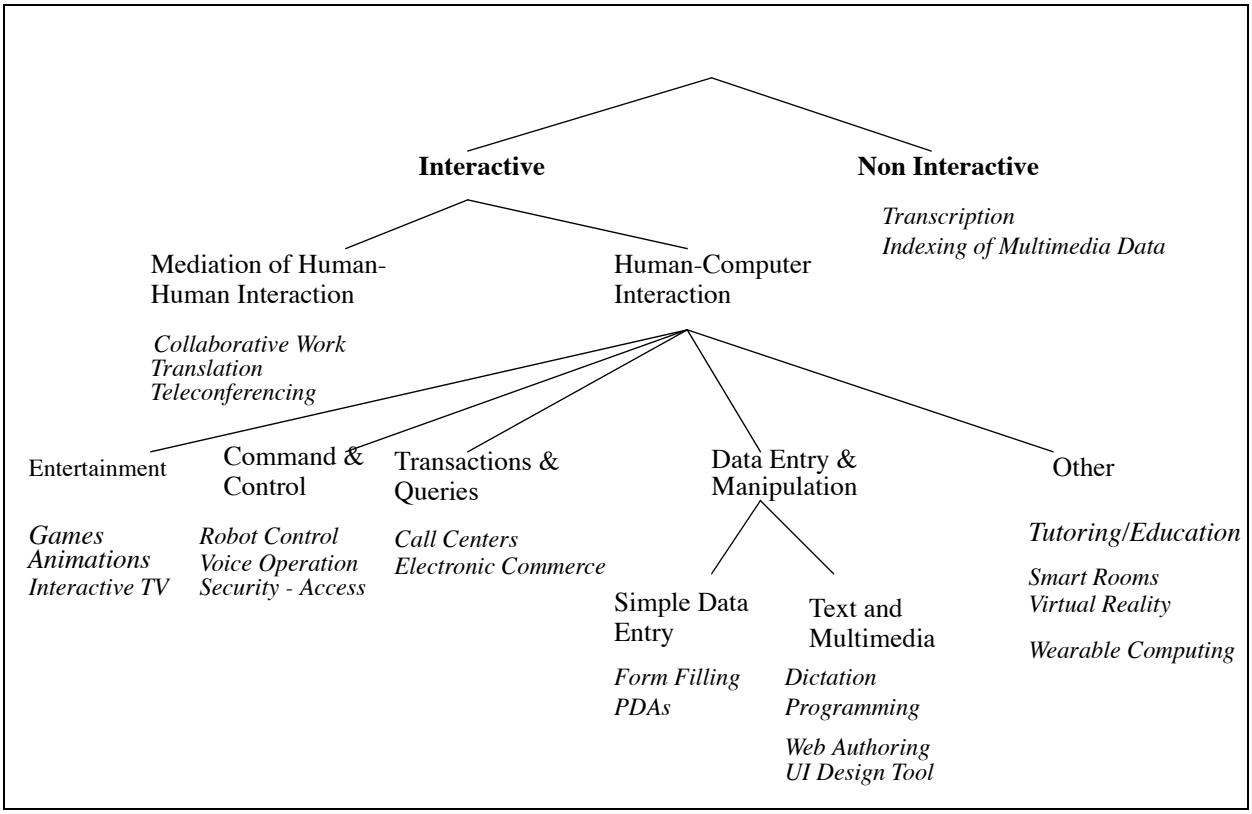


Figure 1-1. *Task-oriented taxonomy of speech recognition applications*

This taxonomy will be further developed as more speech recognition applications emerge. Being a task-oriented taxonomy, applications that involve more than one task do not fit a single category. For example, automatic processing of car rental requests involves filling out a form specifying the type of car, rental period, rates, etc., but may this form be accessed via telephone in a natural language dialogue. Therefore, processing a rental car inquiry matches both the "transactions&queries" and "simple data-entry" task categories. Such ambiguities, however, do not diminish the usefulness of the proposed taxonomy in providing a conceptual framework for the discussion of issues in speech recognition applications.

Dimensions other than task can be used to develop other taxonomies of speech recognition

applications. For example, Nigay and Coutaz [Nigay and Coutaz 1993] present a taxonomy of multimodal systems based on the three dimensions of: level of abstraction at which modality integration is performed, use of modalities (sequential or parallel), and type of modality fusion (independent or combined). Another classification dimension is the types of modalities beyond speech that are supported. The following subsection provides a brief introduction into multimodal interfaces.

1.1.2 Multimodal Interfaces

The ease and robustness of human-human communication is due to highly accurate recognition that exploits the redundant and complementary use of several modalities. Human-computer interaction can benefit from modeling several modalities in analogous ways. This section discusses advantages of multimodal user interfaces, defines how "multimodal" is understood in this dissertation, and provides an overview of published multimodal interfaces.

What are the advantages of multimodal user interfaces? A recent workshop on multimedia and multimodal interface design [Blattner and Dannenberg 1990] identified areas where user interface design can benefit from the use of multiple modalities:

- *Interpretation accuracy and presentation clarity through modality synergy*: On the input side of a computer system, interpreting input conveyed redundantly in several modalities can increase interpretation accuracy. An example is combining speech recognition and lipreading in noisy environments. Using different modalities to provide complementary information can facilitate interaction; for instance, deictic references to graphic objects are easier to express by pointing rather than by speech, and commands are easier to speak than to choose from embedded menus using a pointing device. On the output side of a computer system, multimedia output is inherently more expressive than single modality output.

- *New applications*: Some tasks are cumbersome or even impossible to perform if constrained to a single modality. For instance, interactive TV is much more compelling in a natural language dialogue with the system than when the user is forced to push buttons on a remote control or some other form of keyboard. And architectural designs require at least two modalities (drawings and writing are needed).

- *Freedom of choice*: Although some tasks may be achieved with equal efficiency using different modalities, choice among modalities is valuable because users differ in their modality preferences. Moreover, user needs may differ, for example, for disabled users or for people with Carpal Tunnel Syndrome who cannot use a keyboard.

- *Naturalness*: Offering multiple modalities to interact with a computer can be more natural to the human user, especially when habits and strategies learned in human-human communication can be transferred to human-computer interaction. Also, the mapping of user intention to input can be more direct (p. 206, [Rhyne and Wolf 1993]). "Natural" is however frequently used in vague terms, and generally needs clarification when used.

So far "multimodal" has meant using more than one modality for either input or output in a computer system. The remainder of this dissertation will use the term *multimodal interface* in the following, more restrictive sense: a human-computer interface that integrates speech input with some other input modality. The following overview of multimodal interfaces enumerates modalities that have been associated with speech input in published research systems.

Combining Speech with Pointing and 2d/3d Gestures: "Pointing" is using a pointing device or touch-screen to refer to objects displayed on the screen. "2d gesture" (or simply gestures) indicates movement on a flat surface (e.g., drawn with a pen on a flat panel display). "3d gesture" means movements of fingers or of the whole hand in three dimensions. It is beneficial to use gesture in multimodal interfaces rather than speech alone because deictic references to objects

are much easier to express in gestures than in speech [Bolt 1980]. Furthermore, gestures may be advantageous in indicating the scope of operations. Research systems that combine speech with pointing or 2d/3d gestures include interaction with maps: city maps [Cheyer and Julia 1995], real estate maps [Oviatt, DeAngeli et al. 1997], geographic maps [Koons, Sparrell et al. 1993], and calendars [Vo and Wood 1996]. Other systems combining speech with pointing or 2d/3d gesture were developed for graphic document manipulation [Hauptmann 1989; Fauré and Julia 1993], and analysis of video and image data [Cheyer 1997; Waibel, Suhm et al. 1997]. In summary, for tasks requiring deictic references or the indication of scopes, combining speech with gestures is advantageous.

Pen-based Interfaces: Handwriting input in multimodal interfaces should imitate the use of drawing devices and paper. The user writes with a stylus on a writable display (e.g., a touch-sensitive display). Handwriting input has long been considered an alternative to keyboard input - without combining it with voice input. Pen-computing (or pen-based interfaces) has emerged as a field devoted to developing useful computer devices and tools that are based on handwriting and 2d gesture input. Despite the fact that handwriting is inherently a slow input modality and that the performance of current handwriting technology is considered to be too inaccurate, all studies exploring pen-based interfaces conclude that pen computing is promising for future use (e.g., [Rhyne 1987; Thomas 1987; Briggs, Beck et al. 1992; Frankish, Hull et al. 1995]). Some recent successful commercializations (e.g., 3Com's PalmPilot ®) provide further evidence of the attractiveness of pen-based interfaces. Applications conducive to pen-based interfaces include text editing ("electronic paper"), speadsheets, graphics, and personal digital assistants.

Combining Speech and Pen Input: Combining speech with handwriting and gesture input has so far been explored for visual programming [Leopold and Ambler 1997] and multimodal maps [Cheyer 1997]. A wizard-of-oz simulation suggests that the combination of speech and pen input could be particularly beneficial for error correction in speech recognition applications [Oviatt and VanGent 1996] - an observation which will be discussed in further detail in

this dissertation.

Combining Speech with Eye-Movement and Gaze: Two main approaches have combined speech with eye movement or gaze information. First, gaze information can be used to improve speech recognition performance. Since eye fixations correlate with deictic object references during human-computer interactions, information gleaned from eye fixations can provide hints about what a user is likely to say. For instance, when looking at a map, the user is likely to refer to objects that are within the recent range of fixations [Sarukkai and Hunter 1997]. Second, gaze information has been used for selection and manipulation of objects, equivalent to mouse click and dragging operations [Jacob 1993; Wang 1995; Flanagan 1997].

These examples of multimodal interfaces illustrate the potential benefits of combining speech with other modalities across a variety of fields. So why is speech recognition technology not yet successfully deployed in all of these useful applications? What are the limitations of current speech recognition technology that hinder more successful applications?

1.2 Limitations of Current Speech Recognition Technology

Limitations of current speech recognition technology include lack of performance on general domains and under noisy real-world environments, difficulty of conveying domain restrictions such as limited vocabularies to users, lack of toolkits supporting application development, and recognition errors.

Recognition accuracy appears to be the main factor determining user acceptance of speech applications [Newell, Arnott et al. 1991; Lai and Vergo 1997]. Many speech recognition applications must either operate in noisy environments (in particular, interactive services and applications embedded in the environment) or require high accuracy on very general recognition tasks (any transcription application, text and multimedia data entry) or both (for example, mediation of human-human communication)[1]. Performance of current speech recognition

1. The terminology used here is based on the taxonomy of speech recognition applications presented in Section 1.1.1.

technology is still insufficient on many of these challenging recognition tasks.

Current speech recognition technology works sufficiently well on restricted domains. The performance is adequate so long as the user complies with the domain restrictions. However, it is difficult to convey these domain restrictions to the user. Important domain restrictions include complexity and style of the language (e.g., read versus spontaneous speech), and vocabulary limitations on words and word sequences that can be recognized.

To make a technology available for widespread deployment, non-experts of the technology need to be able to integrate it into applications. Some speech recognition toolkits have become available, such as OGI spoken language systems toolkit, HTK speech recognition toolkit, Microsoft SDK, Dragon's and IBM's Development Toolkits). But integration of speech recognition technology into applications still requires a significant level of expertise. Furthermore, porting standard recognition systems to new applications is still difficult.

Finally, the problem of recovery from the inevitable recognition errors has been insufficiently addressed. Recognition errors are frequently non-intuitive; therefore, strategies for recovering from recognition errors in normal conversation are not applicable to speech recognition applications. This point that will be discussed in more detail in Section 1.4 later in this introduction. Informal surveys (e.g., at the EUROSPEECH '97 conference, and a recent Newsweek technology focus [1998]) suggest that error correction is perceived as a significant weakness in available speech recognition products.

This dissertation addresses the problem of error correction in speech recognition applications. Why are recognition errors a persistent problem, and why are the current solutions insufficient? The following sections attempt to answer these questions by discussing the limitations of current error-correction techniques across different speech recognition applications, based on the taxonomy of speech recognition applications introduced earlier.

1.3 Repair in Different Speech Recognition Applications

One design problem in speech recognition applications is how to handle inevitable recognition errors. Design solutions must balance constraints and trade-offs along several dimensions, including task, application, hardware, and software (both recognition technology and supporting technology). Specific aspects of speech recognition applications influence the design trade-offs and determine the design space for repair in speech recognition applications. These aspects range from the task goals to the limitations of the application environment. Design options have been identified for repair in different speech recognition applications.

1.3.1 Design Space

The design space for repair in different speech recognition applications is determined by the dimensions of interaction style, interaction goal, available modalities, and level of input.

Researchers in the field of human-computer interaction have identified four general interaction styles (see [Shneiderman 1997], p. 73). *Command language interfaces* allow the user to issue commands and to control a system. The interaction is very concise, using a keyboard or voice commands. Form filling interfaces, or more general *data-entry applications*, support question-and-answer type data entry and data modification. *Direct manipulation interfaces* allow the user to manipulate directly objects that are visually or symbolically represented in the interface. *Conversational interfaces* engage the user in a natural language dialogue, typically a spoken dialogue. The design of repair methods must fit these general interaction styles.

Interaction goals in the context of error correction range from correct entry of data item by item to initiation of an action or the communication of information. While semantic correction is sufficient for the latter two goals, the former requires that each item be recognized verbatim.

The application context and hardware determine the available modalities. Current telephone applications limit interaction to speech and touch-pad input. It is commonly anticipated that speech will become the dominant input modality for small mobile devices such as palmtops.

However, small writable displays can be integrated, making handwriting and gesture input possible; future telephones also may be equipped with such displays. On the other end of this spectrum, applications that are embedded in the user's environment (e.g., smart rooms) allow the use of any modality. The design of speech recognition repair is obviously limited by what modalities are available for a speech recognition application.

User input can occur at different levels: single characters or digits, isolated words, phrases, or sentences, or whole conversations. Accordingly, repair input can be at any of these levels.

Given these design dimensions, what options are appropriate for the design of repair in different speech recognition applications?

1.3.2 Design Options

The taxonomy of speech recognition applications presented above defined the following task categories: non-interactive, data entry and manipulation, dialogue and control, and mediation of human-human communication. The ensuing paragraphs discuss appropriate design options for repair of speech recognition errors in each of these task categories. With the design dimensions identified in the previous subsection - namely interaction style, goal of interaction, available modalities, and level of user input - this discussion takes place in a matrix defined by task categories and design dimension. Only a high-level overview is presented, but that is sufficient to provide the context of this dissertation research.

Non-interactive applications depend either on sufficient speech recognition performance (eliminating the need for repair), or the presence of an operator to switch into interactive mode for repair. As non-interactive applications are beyond the scope of this dissertation, they will not be included in further discussions.

For tasks involving data-entry and modification, command/control interaction and data entry are the appropriate interaction styles. The interaction goal is the input of every item with sufficient accuracy, and, in most cases, verbatim recognition. Since tools for data-entry and modi-

fication tasks lend themselves to integration with a graphic user interface, repair options extend to other non-speech modalities such as gesture and handwriting. Different levels of repair input may be appropriate, depending on the type of data being entered or manipulated.

Control and dialogue tasks may entail all types of interaction styles, although conversational interactions are dominant in most of the applications in this category (e.g., all types of interactive services, smart rooms, and virtual reality interfaces). Therefore, such applications suggest the use of conversational repair (also called *clarification dialogues*). Semantic repair is almost always sufficient. Modalities may be limited to speech only (e.g., current automated telephone services), but some applications afford multiple modalities (e.g., interactive travel services and future dialogue applications embedded in the environment); therefore, multimodal error correction is an option even for such applications. User input is at the level of words, phrases, and sentences.

Mediation of human-human communication implies conversational interaction styles. Other interaction styles would interfere with the flow of the primary human-human communication (i.e., the communication that is being mediated by the application). Semantic repair is appropriate in most cases, although some tasks may require verbatim repair (e.g., in the case of collaborating on papers or design plans). The range of available modalities depends on the specific application; while tools supporting collaborative work may integrate a graphic user interface, speech translation tools clearly have speech as the preferred modality. Such applications obviously process the entire conversation being mediated, although repair input typically will range from words to sentences.

After this cursory discussion of repair in different applications, we focus on the category of applications for which this dissertation is most relevant: applications for which non-conversational repair is appropriate and which offer to use multiple input modalities. What non-conversational repair techniques have been developed for such applications, and what are their limitations?

1.4 Drawbacks of Current Non-Conversational Repair

Non-conversational multimodal speech recognition applications include some of the non-interactive applications and all data-entry/modification applications. Non-conversational repair techniques may also be appropriate for applications that use a conversational interaction style, for example, in subdialogues that focus on data entry. Three main non-conversational repair techniques are known and occur in variations in many published research and commercial systems: respeaking (repeating, using continuous speech or isolated words), choosing from a list of alternatives, and using mouse and keyboard.

With current speech recognition algorithms, *respeaking* is an ineffective method because most speech recognizers do not improve performance on repetition and often actually get worse. In human-human dialogue, hyperarticulated repetition is very effective and often preferred over other methods (see the review of repair in human-human dialogue in Section 2.2.3). While people frequently hyperarticulate repeated words or phrases, repetition is not effective in speech recognition applications for two reasons. First, repetition is unlikely to be recognized correctly since, in most cases, recognition errors are not random but caused by deficiencies in the internal models of a speech recognizer. Second, the performance of most recognizers deteriorates on hyperarticulated speech because they are trained exclusively on normally pronounced speech.

Choosing from a list of alternatives has been very effective in some disconnected speech applications such as dictation systems that require a pause between every word. Methods to select words from a list using voice only have been developed (using spelling for selecting and filtering). Some commercial systems offer additional editing capabilities by voice, including navigating within text (e.g., "back two words") and deleting items (e.g., "delete three words"). However, choosing from a list of alternatives is not effective in continuous speech recognition applications because the correct sequence of words is rarely among the top alternatives listed, especially if several consecutive words are misrecognized. Experiments presented later in this dissertation (see Section 8.3) provide further evidence supporting this

claim.

Finally, using mouse and keyboard defeats the purpose of employing speech as an alternative to keyboard as input modality. Correction using keyboard and mouse is possible only in applications that allow the use of a keyboard and graphic user interface. Furthermore, the fact that most computer systems still rely on keyboard input is one major obstacle to making computer technology accessible to a broader public.

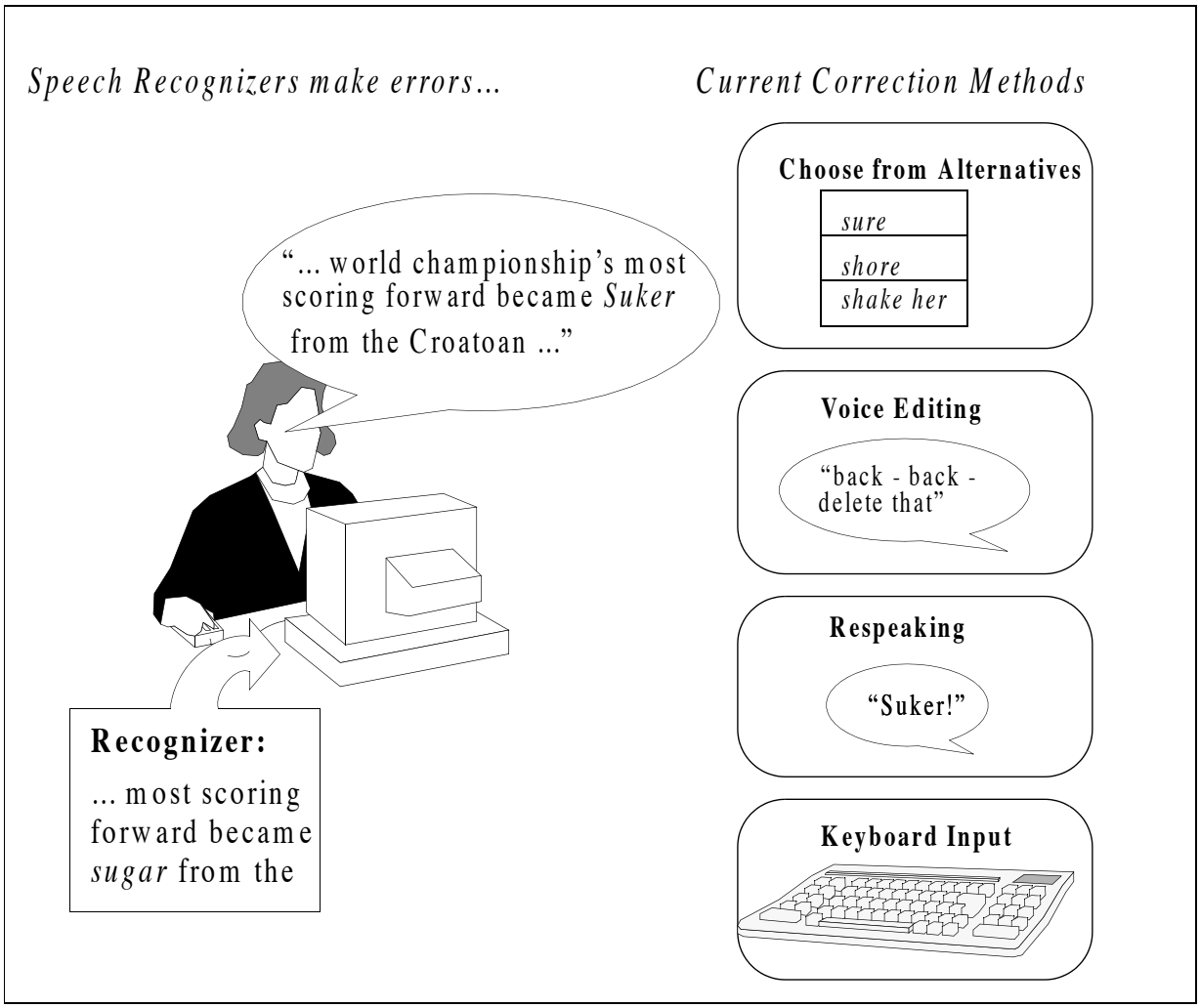


Figure 1-2. *Overview of current correction methods for non-conversational speech recognition applications*

Figure 1-2 illustrates the available correction methods for non-conversational applications: choosing from a list, voice editing, respeaking, and finally keyboard input.

Hence, current correction methods for *non-conversational* speech recognition applications are either ineffective or require keyboard input. For *conversational* speech recognition applications, clarification dialogues provide correction in the context of a spoken dialogue, similar to conversational repair strategies that people use. Our review of research on clarification dialogues (see Section 2.4.2) revealed that such techniques are still not well developed. It is therefore not surprising that informal user surveys of conversational and non-conversational speech recognition applications suggest that users perceive error correction as a significant weakness. The problem of errors in speech recognition applications remains unsolved; more efficient and graceful methods to recover from recognition errors are needed. This dissertation addresses how to reduce user effort spent on error recovery in *non-conversational* speech recognition applications.

1.5 The Research Question

Given relatively low reliability of current speech recognition technology, how can the users' effort necessary to recover from interpretation errors be minimized?

Faced with this research question, one could believe improvements in technology will eventually lead to perfect recognition (or, at least, by delivering sufficient recognition on all applications), thus eliminating the need for error correction. Alternatively, one could believe that speech recognition will remain imperfect in the foreseeable future, and efficient methods to recover from recognition errors are necessary to build useful speech recognition applications.

This dissertation adopts the second viewpoint. This view is supported by the observation that despite continuous gradual improvements, 30 years of research have not resulted in perfect speech recognition. Also, even human performance is limited. Admitting that error correction is necessary does not lessen the benefits of speech recognition technology, because most computer input devices have the potential to produce incorrect input. This is either due to user error or technical malfunction. However, unreliable speech input is acceptable only if appropriate error correction methods are available. In Baber's words: "a prime requirement for ASR

(automatic speech recognition) design is the definition and comparison of error-correction strategies" [Baber, Stammers et al. 1990]. Similarly, from a treatise on recognition-based user interfaces: "Good error-correction methods are critical to the acceptability of a recognition-based system, and should be fast, foolproof, and reduce the probability of future errors" (p. 205 in [Rhyne and Wolf 1993]). This dissertation explores several novel multimodal correction methods, and formally compares them with known methods. Since multimodal methods require some form of graphic user interface, the focus will be on applications with graphic user interfaces. The concepts are demonstrated on a dictation application, which is a well-known example of this kind of speech recognition application.

1.5.1 Example: Error Correction in Dictation Applications

Dictation was chosen as the example application to demonstrate and formally evaluate multimodal error correction. Therefore, an understanding of the problem of error correction in the context of dictation tasks is an important starting point for this dissertation work.

Speaking is generally much faster than typing. Studies showed that on average people can read texts aloud from the Wall Street Journal at 150 words per minute (wpm) [Pallett, Fiscus et al. 1994]. A typing speed of 40 wpm is considered fast unskilled typing [Card, Newell et al. 1983], and expert typists achieve up to 80 wpm. Automatic speech recognition should therefore increase productivity on dictation tasks, although this does not take into account the time spent on correcting recognition errors. If no graceful correction methods are available for speech recognition errors, speech input may be slower overall than keyboard input. Experimental results of this dissertation (see Chapter 8) confirm that speech input is slower than keyboard input when recognition errors have to be corrected without keyboard input. This situation is illustrated in Figure 1-3. Even so, speech input is still an attractive option for applications that do not allow efficient keyboard input, for example, due to hardware limitations (e.g., small hand-held devices), or due to user preferences and needs (e.g., disabled users or those who do not like using the keyboard). The research challenge for this dissertation is to

develop fast correction methods that do not require keyboard input, so that text production using speech is faster than typing. This dissertation proposes multimodal correction to achieve this goal.

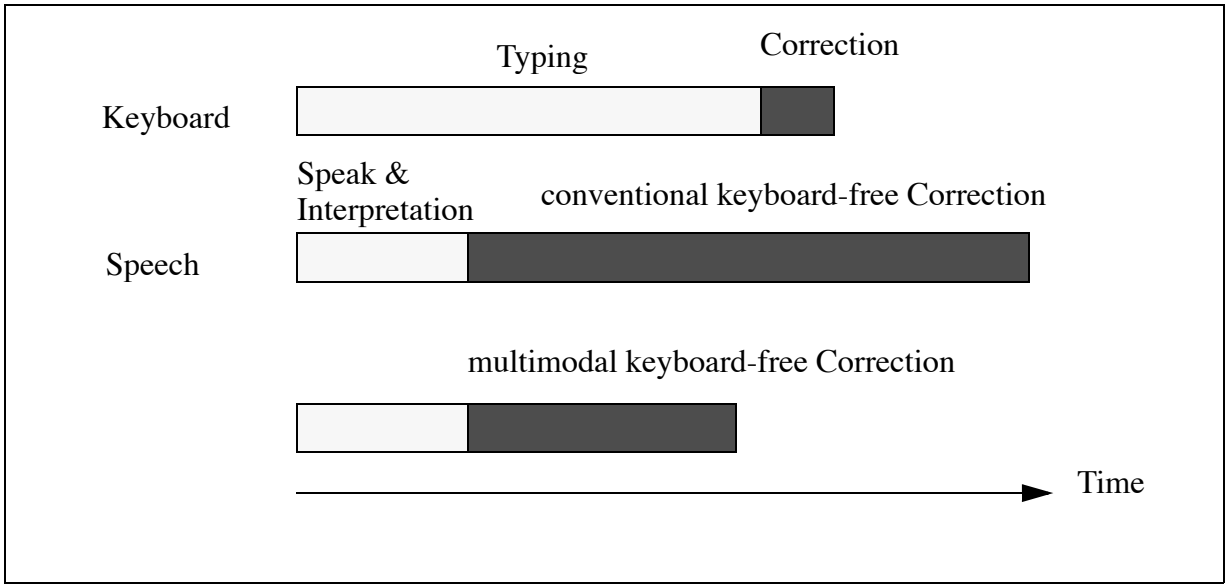


Figure 1-3. *Keyboard versus (continuous) speech input: error correction is crucial to realize productivity gains*

1.6 Thesis and Contributions

This dissertation demonstrates that interactive multimodal error correction provides effective recovery from speech recognition errors in non-conversational applications that utilize a graphic user interface. Speech recognition with multimodal correction achieves text input without any keyboard input at rates that compare favorably to non-expert typing. Research contributions made in this dissertation include:

Concepts and Techniques:

- Novel multimodal interactive correction techniques for speech recognition applications (using graphic user interfaces) for efficient error correction without keyboard input

- Algorithms that significantly increase correction accuracy by correlating correction input with correction context

Artifact: Prototype multimodal dictation system for effective dictation without keyboard input

Theory: Performance model for recognition-based, multimodal interaction (as first step towards a framework for multimodal interaction)

Evaluation:

- Predictive comparison of correction methods, based on the performance model
- Experimental evaluation of novel multimodal correction techniques and formal comparison with conventional correction techniques in user studies

1.7 Dissertation Outline

The remainder of this dissertation consists of four major parts. Chapter 2 contains a literature review. Chapters 3-5, part 1 of this dissertation, describe the technology for multimodal interactive correction. Chapters 6-8 (Part 2) present the evaluation of multimodal interactive correction. The dissertation closes with conclusions in Chapter 9.

Part 1: Technology. Chapter 3 describes the component technologies required for any multimodal system that integrates voice input, handwriting, and gesture input: automatic speech recognition, on-line handwriting recognition, and recognition of 2d gestures. The concept of multimodal interactive correction is presented in Chapter 4. Additionally, algorithms are proposed that make interactive correction effective by correlating correction input with context. Chapter 5 describes the integration of multimodal correction with a large vocabulary continuous speech recognition system to build a prototype multimodal dictation system.

Part 2: Evaluation. The evaluation section begins in Chapter 6 with a review of previous studies of text production techniques. The studies include comparisons of dictation to a machine,

dictation to a secretary, handwriting, and typing, as well as evaluations of simulated automatic listening typewriters. Chapter 7 presents the performance model of recognition-based multimodal interaction. Predictions on the effectiveness of different multimodal correction techniques are derived by applying the model to interactive multimodal correction. Chapter 8 describes the experimental evaluation of multimodal interactive correction and the comparison of novel multimodal methods with conventional methods. The results show the effectiveness of multimodal correction and demonstrate that correlating correction input with context can significantly increase correction accuracy.

In conclusion, Chapter 9 discusses benefits and limitations of multimodal interactive correction. This final chapter ends with a summary of the research contributions and an outlook on future research.

The contents of the appendices is as follows: Appendix A describes the materials used during the user studies of the multimodal dictation system, including all forms, and a quick tutorial of the interface. Appendix B reviews the theory of repair in human-human dialogue. This supplies an important background for investigating the problem of error correction in conversational speech recognition applications, which is beyond the scope of this dissertation. Appendix C briefly describes different standard benchmark tasks used to evaluate continuous speech recognizers, as a reference for readers unfamiliar with the jargon of the speech recognition field. Appendix D contains demographic information of the participants of the final user study. Appendix E contains a glossary of frequently used terms.

2. Literature Review

Previous research on errors in speech recognition technology and on how to address this problem is the foundation for this dissertation work. This chapter provides this foundation by reviewing the literature in relevant fields on these two issues.

The first Section 2.1 summarizes classifications of speech recognition errors. Taxonomies provide a concise language in any scientific discipline. The speech recognition field has adopted mainly two classifications: one describes different types of errors, the other assigns blame for errors to different system components. The second classification naturally leads to research on which factors influence speech recognition performance, which is reviewed in the following subsection. Factors increasing the error rate of speech recognizers include spontaneous speaking style, fast rate of speaking, low signal-to-noise ratio, high frequency of short or out-of-vocabulary words, large vocabulary size, and high perplexity of the language. By looking at these factors, the application developer can investigate why the error rate in the application may be high. However, these factors cannot explain why a particular error has occurred, or predict when an error is likely to have occurred. That knowledge would obviously be very useful for the detection and correction of speech recognition errors. Different methods, have been developed to equip speech recognizers with a notion of how likely the recognition output is correct, and are summarized in the final subsection. Such confidence measures can be applied to predict speech recognition errors.

One of the great hopes in automatic speech recognition has been to make speech - the preferred medium of human-human communication - usable for human-computer interaction. When considering error resolution speech recognition applications it is therefore intuitive to

ask what we know about error resolution in human-human communication. Section 2.2 discusses research on repair in human-human communication in view of error correction in speech recognition applications. Repair strategies people naturally employ in conversations provide useful guidance how to make error correction in speech recognition applications more natural and intuitive. Furthermore, it provides a conceptual framework for the investigation of error correction in speech recognition applications. Research in linguistics has identified three basic strategies which people employ to deal with communication problems in conversation: preventing errors, monitoring the conversation for possible communication problems, and collaborating on repair.

The final section reviews previous work on error prevention and correction in speech recognition applications. This review is ordered by strategies that are analogous to the ones employed in natural language dialogue: preventing errors by improving speech recognition performance (on specific applications), facilitating error detection by well-designed feedback, and collaborating with the system on error correction. Speech recognition performance can be improved on specific applications either by adapting the acoustic models of the speech recognizer to the user and task, or by using interface design to guide the user towards input which is easier to recognize. Design of feedback which is sensitive to the current interaction context can facilitate the *detection of errors*. Early work on *error correction* in speech recognition applications introduced the two interactive methods that are still used (in variations) in today's systems: choosing from a list of alternatives and repeating the input using continuous speech. A Wizard-of-Oz simulation explored whether error correction could benefit from multiple input modalities. Results suggested that switching input modality in error resolution would expedite error correction and alleviate user frustration in dealing with repeated failed correction attempts.

2.1 Classification and Prediction of Speech Recognition Errors

Classifications of speech recognition errors either categorize errors according to what type of

error occurred or to what caused the error. The first classification provides the basis for word error rate as the commonly used measure of speech recognition performance. The second is widely used during development of speech recognition algorithms to identify problems in the speech recognition algorithm.

Distinguishing by the type of error, the first classification uses the common categories of word *substitutions*, *insertions* and *deletions* (for a formal definition of these terms, for instance, consult [Gibbon, Moore et al. 1997]). Although this is a very shallow taxonomy at the phenomenological level, no other, more sophisticated classification scheme has been widely accepted in the field. For natural language applications, specialized measures have been defined; for instance, concept accuracy in parsing (e.g., [Kamm and Walker 1997]), or rate of acceptable translations in spoken language translation (e.g., [Waibel 1996]). However, these measures are commonly used in conjunction with word accuracy, which remains the only measure to characterize the performance of the speech recognition system used.

In the second classification, recognition errors are classified according to what caused them (e.g., [Gibbon, Moore et al. 1997],[Chase 1997]). The following causes of recognition errors can be distinguished:

- *Out-of-vocabulary word* (OOV, new word): Current automatic speech recognizers are constrained to recognize only words within an *a-priori* defined vocabulary. The recognition algorithm is forced to match words outside of this vocabulary on one or more arbitrary words from within the vocabulary. Therefore, a new word typically leads to one or more recognition errors. For example, on large vocabulary dictation tasks (such as the multimodal dictation system developed in this dissertation), one out-of-vocabulary word on the average causes two recognition errors.

- *Search error*: Most speech recognition algorithms are feasible only with approximations. Large vocabularies or real-time requirements make it impossible to search the whole space of arbitrary sequences of words, and to find the globally

optimal best matching sequence of words. Heuristics allow the system to limit the search by excluding (pruning) regions that are very unlikely to contain the correct sequence of words. Sometimes the correct hypothesis may be lost in this pruning process. Recognition errors caused by such flaws in the search for the best matching word sequence are called *search errors*.

- *Language model error*: Language models are designed to capture the typical word usage of language, according to application requirements. Since no current modeling technique is perfect, a wrong prediction from the language model may overwhelm acoustic evidence, which may have favored the correct hypothesis. Alternatively, the language model may not be able to disambiguate acoustically confusable alternative hypotheses.

- *Acoustic model error*: The acoustic models of an automatic speech recognizers capture acoustic characteristics of speech signals. For state-of-the-art large vocabulary recognition systems, acoustic modeling occurs on three level: preprocessing of the signal, phonetic modeling, and word modeling (via pronunciation lexica). Modeling on any of these levels is imperfect and can cause recognition errors.

As mentioned before, this classification can guide the improvement of speech recognition algorithms. For example, search errors can be eliminated by relaxing the pruning in the search, at the cost of an increase in recognition time. In the course of this dissertation work, the classification has helped to identify the main causes for initially very poor performance of the continuous speech recognizer on corrections by respeaking (see Section 4.4.1, page 94).

In designing a new speech recognition application, an important issue is to judge what recognition accuracy can be expected from available speech recognition systems. While the absolute performance obviously depends on the current state-of-the-art of technology, knowing what factors influence speech recognition performance in general can help to extrapolate available benchmark results to the specific application at hand.

2.1.1 Factors Affecting Speech Recognition Performance

Some researchers have tried to identify factors which generally (across recognition systems and speech applications) influence the recognition accuracy. Regular performance evaluations of speech recognition systems on standard benchmark tasks are good opportunities to infer such statistics. For instance, Fisher [Fisher 1996] examined what factors affect recognition error rate based on results of the official 1994 Wall Street Journal dictation evaluation (for information on speech recognition benchmark tasks, see Appendix C). The statistical analyses identified the following factors: speaking rate (fast speech is correlated with high error rates), length of utterance (long utterances are recognized more accurately), and signal-to-noise ratio (low ratio is correlated with high error rates). Another study [Weintraub 1995] suggests that word length (short words are more difficult to recognize) and speaking style ("read dictation" versus "read conversational" and "spontaneous conversational") influence recognition accuracy as well.

A recent study [Alleva, Huang et al. 1997] pointed out the impact of another dimension in speaking style on recognition accuracy: whether speech is spoken fluently, or with pauses between words. People intuitively elongate pauses between words when they encounter communication problems (e.g., [Oviatt, Levow et al. 1996]), because that strategy facilitates understanding in human-human communication. However, the performance of most continuous speech recognizers deteriorates when words are separated by pauses - unless the recognition algorithm is enhanced to handle both continuous and isolated speech well, for example, by pooling continuous and isolated speech data to train the acoustic models [Alleva, Huang et al. 1997].

2.1.2 Predicting Recognition Errors

Predicting whether an automatic speech recognizer committed an error in recognizing some speech input would obviously be useful to facilitate error detection and correction. For instance, a dialogue system could ask the user to repeat a query which is likely to have been

misrecognized. Predicting exactly which words are likely to be incorrect would permit to flag errors automatically and to take appropriate error recovery actions. What action is appropriate depends on the application. For example, in audio annotation applications, likely misrecognized words could be filtered from the speech recognizer's output. In speech-to-text dictation, the system could display likely recognition errors in a different color or style.

Confidence measures are methods that allow to predict speech recognition errors. Most approaches assign an *a posteriori* probability P(correct|word) how likely a word is correct for each word in the recognition hypothesis. The following sections review recent work on confidence measures for speech recognizers, and their application to predicting a certain kind of recognition error, namely out-of-vocabulary words. Confidence measures are used in this dissertation as one method to automatically locate recognition errors (see Section 4.2.2, page 82).

2.1.2.1 Confidence Measures for Continuous Speech Recognition

According to Chase [Chase 1997], confidence measures can differ along the following four dimensions: at what level of abstraction confidence is annotated, how an error is defined, what method is used to generate confidence annotations, and how their goodness is measured. The next paragraph describes typical choices for each of these dimensions in current confidence annotation algorithms.

Confidence can be annotated at either the sentence level (for utterance rejection, e.g. in dialogue systems), the level of semantic concepts, or the word level. An error is defined as mismatch in the alignment of the recognition hypothesis and the true word sequence. All current confidence annotators are based on combining a set of predictor variables. Sets of predictor variables can be combined using regression models, decision trees, or neural networks. To evaluate confidence annotators, the following measures have been used: the cross entropy reduction of predicting the correct confidence labels (which more intuitively equals to the amount of uncertainty left after applying a confidence annotator to predict errors), the rates of false alarms (correctly recognized words mistakenly labelled as error) and missed detections

(misrecognized words not labelled as error), and the overall accuracy of classifying words as either correct or misrecognized.[1] Whereas the first measure is general - but not very intuitive - the two other measures are very intuitive, but specific to applying confidence annotators as error predictors. The set of predictor variables is crucial for the development of a good confidence annotator. What variables are good predictors of recognition errors?

Many different predictor variables have been proposed and evaluated (see [Chase 1997], [Kemp and Schaaf 1997], [Schaaf 1996]). Chase [Chase 1997] defined different acoustic predictors (normalized acoustic score, distance between hypothesized and best score phone, number of occurrences in training data, number of phones in word), language model predictors (language model score, information on how the language model score was computed) and predictors based on information in the N-best list of alternative hypotheses (normalized number of occurrences in the various N-best hypotheses, number of unique words present in the N-best lists averaged over each frame in the guessed segmentation). The best confidence annotator that Chase derived for a large vocabulary read speech recognition in American English achieved a cross entropy reduction of 20.9%, which corresponds to an error/correct classification accuracy of around 80%.

Kemp et al. [Kemp and Schaaf 1997] introduced a new lattice based predictor variable, called "gamma". Interpreting the acoustic scores as emission probabilities and language model scores as transition probabilities, the word lattice for some speech input can be interpreted as a Hidden Markov Model (HMM, for a tutorial on HMMs see e.g. [Rabiner 1991]). Thus, the posteriori probability for each link in the lattice (representing a word) can be computed using the standard forward-backward algorithm for HMMs. This probability can be interpreted as confidence score for each word in the lattice. Schaaf [Schaaf 1996] compared the performance of "gamma" and five other lattice based predictors with eleven non-lattice predictors. "Gamma" alone achieves a classification accuracy of 89% on a database of spontaneous con-

1. For more information on evaluation of confidence annotations, see [Chase 1997; Kemp and Schaaf 1997]

versational (German) speech, which is almost as high as the best combination of any other predictor variable, and "gamma" can be calculated in real-time - crucial for deployment in a user interface. "Gamma" was employed in this dissertation to implement a method to automatically highlight likely errors (see Section 4.2.2)

Is it also possible to predict the specific kinds of error which occurred?

2.1.2.2 Predicting Out-of-vocabulary Words

Some work has been devoted to applying confidence annotators to predict out-of-vocabulary words (OOVs). Since current speech recognition systems try to find the closest match from among the words within a given vocabulary, OOVs inevitably lead to errors. Asadi first investigated the new word problem in speech recognition. He presented a new word model that enables a standard continuous speech recognizer to detect new words [Asadi, Schwartz et al. 1990]. His approach was however tested only on a constrained task, and new words were limited to names that were eliminated from the recognizer's vocabulary. In my Master's thesis [Suhm 1993], I pursued a similar approach and showed that the language model can significantly improve the detection of new words. However, it remains unclear how this approach scales to large vocabulary speech recognition. Chase applied her approach to confidence annotation (described earlier) also to the problem of detecting OOVs (see Chapter 6.10 in [Chase 1997]) - without too much success. For example, on a large database of read speech, a decent OOV detection rate of 70% is possible only at the cost of high false alarm rates of 25%. Since OOVs generally are infrequent, high false alarm rates are unacceptable for most applications.

To illustrate the frequencies of OOVs, Table 1 shows the (static) coverage of matching unseen newspaper text as a function of vocabulary size for English. For more inflectional languages such as French or German, larger vocabularies are necessary to achieve similar coverages. Vocabulary sizes of 20,000 to 30,000 words are typical in current dictation systems. However, this statistics underestimates the rate of out-of-vocabulary words, because in general, text

from many different sources will have to be processed by a general-purpose text processing system, and additionally, language evolves gradually in usage. Hence, out-of-vocabulary rates of around 5% have to be expected when dictating general text [Acero 1998]. This concludes the review of work on classification and prediction of speech recognition errors.

Table 1: *Coverage of (matched) unseen text as a function of vocabulary size (from [Cole, Mariani et al. 1995], page 39)*

Vocabulary Size	Text Coverage
20,000	94.1%
64,000	98,7%
100,000	99.3%
200,000	99.4%

2.2 Lessons Learned from Repair in Human-Human Dialogue

One of the great hopes in automatic speech recognition is to communicate with computers using speech, the medium we are so familiar with from our daily communication with other people. We generally deal with communication problems very effectively and usually do not even notice them consciously. What is known about error resolution and *repair in human-human dialogue*, and what lessons can be learned for speech recognition applications?

The investigation of repair in human-machine interaction via speech can benefit from research on repair in human-human dialogue in a number of ways: First, the repair strategies that users intuitively employ are likely to be similar for human-human and human-machine dialogue, because speech recognition applications are expected to enable more "natural", i.e. more human-like, ways to communicate with a machine. Second, the conceptual framework developed for repair in human-human dialogue can provide a good starting point for the design of (and a theory for) repair in human-machine dialogue. While Appendix B reviews the literature on repair in human-human dialogue, this section focuses on relating this body of research to this dissertation: issues of grounding in speech recognition applications, relevant categories of errors, strategies to deal with errors, and motivation for the approach chosen in this disserta-

tion.

2.2.1 Grounding and Feedback

Grounding is the process of extending the shared knowledge of communication partners. Uncertainty about what knowledge is shared can cause communication problems. Applied to human-machine dialogue, this leads to a well-known problem of user interface design in general: the issue of good feedback. Research on grounding in natural language dialogue can therefore help to address the problem of feedback in speech recognition applications.

Brennan and Hulteen [Brennan and Hulteen 1995] address the issue of *feedback in speech recognition* applications from the viewpoint of grounding. Feedback should be context-sensitive and based on a model of grounding with spoken language systems. They asserted that most current systems provide feedback in a rather ad-hoc way. They applied Clark and Schaefer's collaborative theory of conversation to spoken language systems (see. Appendix B). They pointed out the importance of both *negative* and *positive evidence of understanding*, i.e. evidence if one communication partner notices a potential problem, as well as evidence of what actually has been understood. People use back-channel utterances to provide negative and positive feedback to the speaker. Back-channel utterances include verbal cues (e.g., "Uh huh", "Ok" or "What?") and non-verbal cues (e.g., nodding or shaking the head, facial expressions). Extending Clark and Schaefer's list of communicative stages [Clark 1987], Brennan and Hulteen identified eight stages of human-machine communication (not attending, attending, hearing, parsing, interpreting, intending, acting, reporting), and discuss design options for positive and negative feedback in each of the stages.

A taxonomy of communication problems according to the communication stage, like the eight stages Brennan proposed, are therefore useful for the design of good feedback in human-machine spoken language interaction. What do other taxonomies of communication problems in human-human dialogue contribute?

2.2.2 Error Taxonomies

This section relates different taxonomies of communication problems in human-human dialogue to the problem of errors in speech recognition applications.

Considering repair in human-machine interaction, the distinction between user and system errors is very important. Much research in the field of human-computer interaction is devoted to reducing the frequency of *user errors* through good user interface design. Work on repair in speech recognition application however has to focus on *user corrections* of *system interpretation errors*: speech recognition errors are the additional challenge. Since lack of system competence caused them, the system depends on help from the user to recover from them.

Nevertheless, some research has been conducted that addresses the opposite problem, (system) correction of user errors: Bradford [Bradford 1990] proposed a technique which detects and corrects user errors in command-oriented interfaces, building on the notion of a *do-what-I-mean interface*. Generalizing this approach, Nerzig [Nerzig] examined detection of erroneous plans in human-machine interaction, based on a formal theory of plan recognition. The description why a plan is erroneous can be used to (system-)initiate a clarification dialogue. However, modeling what the user intended is a difficult problem and still beyond the capabilities of current artificial intelligence methods. System correction of user errors therefore remains an open research issue.

A taxonomy of communication problems according to the linguistic level where an error occurs can be applied to errors in speech recognition applications in the following way. Depending on the application, different linguistic levels of errors are relevant. The semantic level is appropriate for command/control interfaces and dialogue systems, because the goal of the interaction is to initiate some system action, for example, retrieval of information or help in making travel reservations. By contrast, the lexical or syntactical level is appropriate for data and text input applications.

Taxonomies of communication problems in human-human dialogue thus help to structure

work on repair in speech recognition applications. The work on repair in human-human dialogue leads to solutions for the problem of error recovery in speech recognition applications as outlined in the following section.

2.2.3 Repair

Repair strategies that people employ in dealing with communication problems provide useful analogies and a conceptual framework for error recovery in speech recognition applications. If we assume that speech recognition applications facilitate human-computer interaction by making the communication more similar to human-human dialogue, an approach to repair in speech recognition applications can be considered "natural" if it takes the human-human counterpart into account. In this sense, the collaborative approach to error correction taken in this dissertation can be considered "natural". But the analogy to repair in human-human dialogue reaches farther: what strategies are used to correct errors, what factor(s) determine strategy preferences, and what approaches to the problem of errors in general exist, can be applied to repair in speech recognition applications.

The fact that repeating and paraphrasing an utterance are intuitively employed in human conversational repairs suggests that repeating input and paraphrasing may be an intuitive correction method in human-computer interaction as well. There is some evidence that in human-computer interaction, repetition and reformulation are "intuitively" employed (see [Robbe, Carbonell et al. 1996]).

Moreover, the *principle of least collaborative effort* (postulated by Clark, see [Clark and Wilkes-Gibbs 1986]) is likely to guide user preferences for different correction strategies also in human-machine interaction, just as in human-human dialogue. It can be considered as a reformulation of Card, Newell and Moran's rational user assumption [Card, Newell et al. 1983] in linguistic terms: just as a rational user will prefer interaction methods that minimize the effort, people strive to minimize the effort spent during conversation. However, while delegation of effort to the machine generally saves work in human-computer interaction, in

human-human dialogue the *cumulative* effort spent by both conversation partners is the relevant measure. Obviously, the effort spent by the machine can be ignored as long as the computational power of the hardware is sufficient.

People employ three main strategies to deal with communication problems in human-human dialogue [Clark and Schaefer 1989]: preventing communication problems, monitoring the conversation for potential problems, and collaborating on repair. These strategies correspond to the following analogous strategies for speech recognition applications:

1) Preventing speech recognition errors by increasing speech recognition performance, in particular, on the specific application at hand

2) Facilitating error detection through good feedback

3) Correcting errors in collaboration with the user.

Ordered by these categories, the following sections review previous work on repair in speech recognition applications. The limited work done on feedback was already reported earlier in this chapter in Section 2.2.1.

2.3 Preventing Errors in Speech Recognition Applications

Figure 2-1 summarizes various approaches to prevent errors in speech recognition applications, which are reviewed in brief in the remainder of this section, including improving baseline recognition algorithms, system adaptation (to application or speaker), and user adaptation (by speaker training and interface design).

The section assumes the reader is familiar with the basics of speech recognition algorithms. For an introduction into automatic speech recognition and descriptions of the different components of a speech recognizer, refer for example to [Lee 1990; Cole, Mariani et al. 1995; Woszczyna 1998].

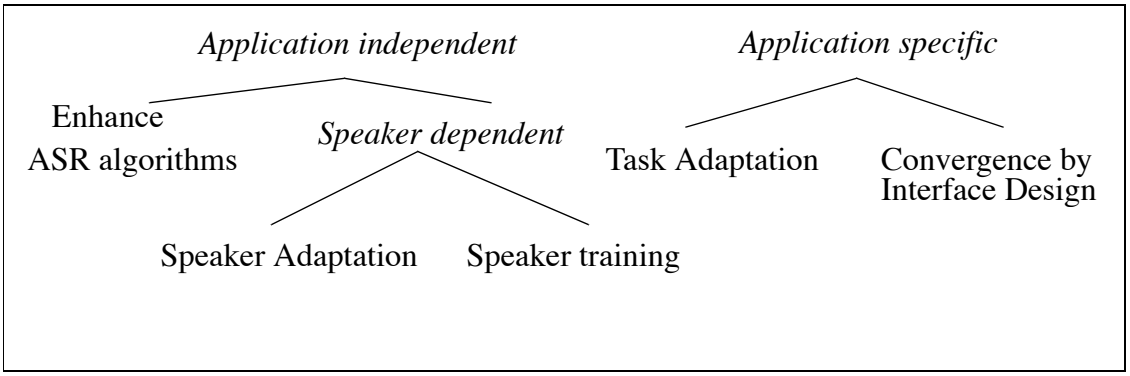


Figure 2-1. *Approaches to improve speech recognition performance on a given application*

2.3.1 Improving Baseline Performance of Speech Recognition Technology

Obviously, an important leverage to prevent errors in any speech recognition application is to improve the recognition algorithm. Without intending to give a full account, this section provides a cursory compilation of some of the most significant advances on speech recognition algorithms in the past decade of research. The algorithmic enhancements can be ordered by components of the speech recognizer: preprocessing, acoustic models, search, and language models. The final subsection presents two approaches to improve speech recognition performance using confidence measures. Along with the main idea of each approach, relative performance improvements on relevant standard benchmark tasks will be mentioned to give the reader an idea how effective each approach is.

2.3.1.1 ***Preprocessing***

Linear Discriminant analysis applies the well-known technique of principal component analysis to the problem of processing the acoustic input signal. Out of a large set of typically 20-50 input features, which are derived from the acoustic signal (including FFT coefficients, delta and delta delta coefficients, silence features), linear discriminant analysis identifies the set of input features (corresponding to the principal components) with maximal discriminative power.

2.3.1.2 Acoustic modeling

Context dependent acoustic models increase the power of the recognizer's acoustic models by taking the phonetic context into account, at the cost of a dramatic increase in the number of modeling parameters. The development began by using acoustic models which model a phoneme with its left and right neighboring phoneme (*triphones*, see [Lee 1990]). However, triphones introduce a large number of model parameters, and the limited amounts of available speech data do not allow derivation of reliable statistical estimates for these parameters. This problem can be overcome by clustering triphones to build subphonetic acoustic units. Several successful techniques have been proposed over the years, for instance Hwang's senones [Hwang 1993], and Dragon's PICs and PELs [Scattone, Baker et al. 1993]. Hwang reported in her dissertation a 20% relative word error rate reduction on the Wall Street Journal large vocabulary dictation task (WSJ).

A new thrust for acoustic modeling improvements was initiated when the field proceeded to include large vocabulary conversational spontaneous speech recognition in the set of standard benchmark tasks. Conversational spontaneous speech is characterized by increased co-articulation. Increasing the phonetic context of acoustic models beyond the left and right neighbor was very successful in improving recognition performance on conversational speech (so-called *polyphones*, see [Kuhn, Lazadrides et al. 1995] and [Finke and Rogina 1997]). Finke and Rogina [Finke and Rogina 1997] reported a 5% word error rate reduction for polyphones on the Switchboard large vocabulary conversational speech task (SWB) and on WSJ.

More recently, *adaptation algorithms* for acoustic models have become very popular. It is well-known that speaker dependent recognizers outperform speaker independent ones. However, speaker dependent recognizers require a large amount of speech from the speaker. Collecting large amounts of speech prior to usage may be impractical or at least inconvenient. Adaptation algorithms improve the recognition performance by adapting speaker independent acoustic models on very small amounts of data. Maximum Likelihood Linear Regression MLLR adaptation [Legetter and Woodland 1996]) has been very successful. For example,

MLLR speaker adaptation achieves a 30% reduction in word error rate for the WSJ recognizer used in this dissertation work (see Section 4.5.1).

2.3.1.3 Search

For a long time, speech recognition algorithms were designed to find the best matching word or the best matching sequence of words only. Schwartz [Schwartz and Chow 1990] introduced an effective and efficient algorithm to find the N best matching recognition hypotheses for time synchronous search methods. Stack decoding based on A* search (e.g., [Paul 1994]) is a popular and straight forward algorithm to find the N best matching recognition hypotheses. N-best lists of matching hypotheses (for short, *N-best list*) and word lattices have become the standard representation of the speech recognizer's output.

Another general technique which yields significant improvements in combination with other techniques is using *multiple search passes*, instead of finding the best matching hypothesis in a single search pass (see p.33 in [Cole, Mariani et al. 1995]). After an initial pass constrained the search space, a later pass can use more sophisticated modeling techniques that would be computationally infeasible in a single search pass. For example, *rescoring of N-best lists* or lattices is quite successful and easy to implement (one of the first references can be found in [Ostendorf, Kannan et al. 1991]). This dissertation will later present some rescoring algorithms that significantly improve the accuracy of repairs (see Section 4.4, page 92).

2.3.1.4 Language Modeling

Despite many attempts at improving the standard statistical word N-gram language model, statistical N-gram models remain the most common language modeling technique. (For a good review, see the chapter on language modeling in [Gibbon, Moore et al. 1997].) Some of the approaches which achieved significant improvements over N-grams include: clustering words [Ney, Essen et al. 1994], collating words to word phrases [Suhm and Waibel 1994; Ries 1996], and using long distance constraints (e.g., trigger pairs [Rosenfeld 1994]). Rosenfeld reported a 15% reduction in word error rate on the WSJ task for a maximum entropy language

model that combines standard N-grams with long distance constraints [Rosenfeld 1994].

2.3.1.5 Improving Speech Recognition Performance with Confidence Measures

Since more reliable confidence measures for speech recognizers have become available, they have been successfully applied to improve the recognition process itself. The following paragraphs present the main idea of two such algorithms.

Setlur [Setlur, Sukkar et al. 1996] proposed an utterance verification algorithm to selectively correct recognition errors. A confidence score is assigned to each of the different alternative hypotheses. If the confidence score of the top candidate is lower than that of the second best, the first and second best hypotheses are swapped.

Zeppenfeld et. al [Zeppenfeld, Finke et al. 1997] and Chase (chapter 8 in [Chase]) showed that confidence scores can significantly increase the effectiveness of unsupervised acoustic model adaptation. Instead of adapting on the whole recognition hypothesis, as in "standard" acoustic model adaptation algorithms, adaptation is performed only on parts of a hypothesis with high confidence scores, thus avoiding adaptation on speech that was not correctly recognized.

Other non-interactive methods to correct errors include postprocessing of recognition hypotheses, for example, using explicit statistical error models that capture the error behavior of a specific recognizer on a certain tasks [Ringger and Allen 1996]. Such approaches yield significant improvements as long as the domain is not yet well modeled.

Although these algorithmic improvements have resulted in impressive increases of speech recognition performance (e.g., from 80% word accuracy on large vocabulary dictation to 94%, see [Rudnicky, Hauptmann et al. 1994]), and the first commercial large vocabulary dictation systems have recently become available, the speech recognition problem is far from being solved: the performance of the best systems on conversational speech and on speech in noisy real-world environments is still too low for many useful applications (e.g., 70% on the Call

Home database, see Appendix C). Improving baseline speech recognition algorithms is, however, beyond the scope of this dissertation.

There are several approaches to improving speech recognition performance on a specific application without having to change the baseline recognition algorithms: adapting the speech recognition system to the application and to the user, and guiding the user towards speech which is easier to recognize. The developer of a new speech recognition application can such methods to increase performance of a baseline system to a sufficient level.

2.3.2 System Adaptation

In addition to developing enhanced speech recognition algorithms and evaluating progress of the technology on standard benchmark tasks - the approach underlying the previous section - performance can be increased on *specific* applications by adapting off-the-shelf speech recognition systems (trained on one of the standard benchmark tasks) either to the application or to the user.

A simple, albeit costly, method for adapting a system to an application is to collect a large number of speech samples. These speech samples have to be "typical" for the intended application. If a working system is not available, simulation techniques (such as Wizard-of-Oz simulations) can be used to collect "realistic" speech samples.[1] After collecting sufficient amounts of speech data, the acoustic models of a speech recognizer trained on benchmark speech databases are adapted using the collected data.

Since speaker *dependent* recognizers outperform speaker independent recognizers, most current commercial speech recognition applications require the user to adapt the delivered speaker independent recognizer. This adaptation is typically organized as an "enrollment" session consisting of two phases: First, the user reads aloud a large number of sentences (50 - 300), which are recorded by the system. In the second phase, the system adapts its acoustic

1. For a review of the commonly used Wizard-of-Oz technique, see [Fraser 1991; Dahlbäck, Joensson et al. 1992].

models using the speech samples collected from the current user. Enrollment yields significant performance improvements, both in accuracy and speed. The improvement is larger for users with low initial performance. The disadvantages of enrollment include inconvenience and loss of time for the user. Obviously, enrollment is not practical for "walk-up and use" applications.

2.3.3 Adapting the User to the System

Besides adapting a speech recognition system (mainly its acoustic models) to either the application or the user, application specific performance can be increased by methods which can be summarized under "adapting the user to the system". Due to normal learning, a user of a speech recognition application will automatically adapt to the system. The user will prefer interactions that the system consistently interprets correctly, and avoid interactions that consistently lead to recognition errors. This natural learning process can be supported in two ways. First, a speech recognition expert can explicitly train the user in speaking styles that cause fewer recognition errors. Secondly, the design of the speech user interface can guide the user to utterances that conform with the system's capabilities, thus also reducing recognition errors.

Improving system performance by training the user can be achieved as follows. First, a person knowledgeable in speech recognition system observes the user while interacting with the speech recognition application, and identifies speaking habits that are known to increase the error rate (some such factors were identified in Section 2.1.1 earlier in this chapter). Then, the user is trained to adopt a speaking style that causes fewer speech recognition errors. A study [Danis 1989] has shown that this approach "considerably" improves the performance of a commercial large vocabulary isolated-word dictation system. However, since the procedure used in this study did not separate user training from adapting the acoustic models (to the user), it is unclear which factor accounts for the improvement. Besides unclear evidence on the effectiveness, user training requires time and an expert who is able to analyze the user's speech habits. Hence, it is generally not an acceptable solution to improve speech recognition

performance.

A rather subtle approach is using the interface design to guide the user towards input which is easier to interpret by automatic recognition systems. Zoltan-Ford [Zoltan-Ford 1991] conducted research on how the vocabulary and phrase structure in system prompts influence user queries. Oviatt [Oviatt, Cohen et al. 1995] extended this research to multimodal applications. In Wizard-of-Oz simulations she investigated how input modality (voice or pen) and presentation format of prompt (structured or unconstrained) can influence user input. The results showed that both modality and presentation format substantially influence linguistic complexity of user input. Hence, the design of system prompts can be used to reduce the complexity of user input, and thus the likelihood of system interpretation errors.

2.4 Interactive Error Correction

Using some of the techniques described in the previous two sections, performance on a specific application can be substantially improved. After exhausting all tricks to increase accuracy, we are again faced with the main research question of this dissertation: given imperfect speech recognition technology, how to minimize the user's effort spent on recovering from recognition errors? Since many research and commercial speech recognition applications already exist, each of them must address the problem somehow. This section presents an overview of interactive error correction methods that have been developed prior to this dissertation.

2.4.1 Basic Interactive Error Correction Concepts and Methods

Baber and Hone [Baber and Hone 1993] were the first researchers to systematically address the problem of error correction in automatic speech recognition applications. They pointed out that error repair consists of two phases: first, an error must be detected, then it can be corrected. The subsequent paragraphs elaborate on the concept of interactive correction and describe the three basic correction techniques that are still used (in variations) in most speech

recognition applications today: repeating input, choosing among a list of alternatives, and clarification dialogues.

Martin and Welch [Martin and Welch 1980] presented the first implementation of interactive correction. A buffer stores (preliminary) recognition results, and the user can perform several interactive operations to edit the buffer: deleting single words or the whole buffer, and the repeating using speech. Some commercial dictation systems still use a buffer.[1]

Correction by repeating using continuous speech (*respeaking*) can be improved in the following simple way. After one attempt at correcting a word, we know that the first-best hypothesis is incorrect. It is therefore straight-forward to eliminate that word from the recognition vocabulary for the interpretation of the repetition. Two groups of researchers [Ainsworth 1992; Murray, Frankish et al. 1992] almost simultaneously proposed this method, calling it "repetition with elimination". They did not formally evaluate the gain by eliminating the word known to be incorrect. For large vocabulary applications, the expected gain is probably small. However, a generalization of this concept, as shown later in this dissertation (Section 4.4), is a very powerful technique to improve the accuracy of correction: if correction input is not interpreted as independent event, but correlated with the context. "Repair by elimination" exploits a bit of context information: the fact that a certain word is known to be incorrect.

The same research papers introduced a second interactive correction method which is still offered in many speech recognition applications: *choosing from a list of alternative hypotheses*. Many variations of this method exist today. The list of alternatives can be presented either visually (when a display is available) or acoustically (e.g., in telephone applications), and the choice can be made either using mouse, keyboard, or voice.

1. The popular disconnected speech dictation system DragonDictate® by Dragon Systems, Inc. (Cambridge MA) is an example of a commercial speech recognition application that offers interactive error correction using a buffer. The buffer keeps the twelve most recent words available for correction. The user can delete and replace words from that buffer by help of voice commands. Correction is effective without using a keyboard, but navigation and choosing from alternatives by means of voice commands is slow.

2.4.2 Clarification Dialogues

Albeit used in research systems beforehand, the idea of correcting errors in a spoken language dialogue (*clarification dialogues*) was first explicitly discussed in [Baber, Stammers et al. 1990]. Clarification dialogues are appropriate for conversational speech recognition applications (also called dialogue systems). Although repair in dialogue systems will not be addressed in this dissertation, the research on repair in this area is important background information.

Clarification dialogues allow the user to recover from recognition errors in the context of a spoken dialogue with the system. *Dialogue systems*, one important category of speech recognition applications, support spoken natural language between user and system. In a dialogue system, several modules interact in complex ways, including automatic speech recognition, robust natural language processing (including a parser and discourse processor), and dialogue management. Although all dialogue systems have to somehow cope with recognition errors, only few published research systems specifically addressed the problem of errors and repair. The following paragraphs highlight some of the published research on clarification dialogues.

Allen [Allen, Miller et al. 1996] presented a robust dialogue system for a train schedule service application which is capable of processing a limited range of user initiated conversational repairs. The user notices system interpretation errors based on system feedback. Similar to repair in human-human dialogue, users tend to correct the error in the next interaction with the system. The system detects and processes the user correction in the following way. The dialogue manager infers the speech act of each user utterance and maintains a goal stack (i.e. the history of what actions the user intended). The discourse model and parser cover a wide range of conversational repairs. By matching speech act and structure of the most recent user query with its internal database of repair templates, a repair can be detected. Hence, clarification dialogues can be implemented by identifying typical patterns of conversational repairs and by using parser and discourse processor to detect repairs among the user queries. Of course this approach is limited by how much the system's parsing and discourse processing schemes are

able to discriminate repairs from other user queries.

Danieli [Danieli 1996] presented a different approach to detecting user initiated conversational repairs. The main idea is that a user initiated repair can be detected from a mismatch between the most recent user query and system predictions on what the user is likely to say next. Such predictions can be derived using a discourse model. Whenever the speech act of the current utterance is not among the derived set of predictions, the system initiates a clarification dialogue. Using pragmatic phenomena that characterize user initiated repairs, the system attempts to process the repair appropriately. For instance in Italian, users appear to initiate repair by repeating the previous utterance. Therefore, if two successive user utterances have the same speech act, the second utterance probably intended as repair. Such pragmatic phenomena obviously depend both on the specific dialogue design and the language. In summary, Danieli exploits mismatches between discourse predictions and the actual discourse to detect conversational repairs, and employs language and system dependent heuristics to successfully process a repair.

In recent work, LuperFoy [LuperFoy and Loehr 1997] presented a generic error recovery algorithm. Like Brennan and Hulteen, she used Clark and Schaefer's analysis of human-human discourse [Clark and Wilkes-Gibbs 1986] as a starting point to develop a four-step process for conversational repair in human-computer dialogue. The four steps to recover from errors include: detection, diagnosis, repair plan selection, and collaborative plan execution. Once an error has been detected, the source of the communication problem has to be known before the most effective repair strategy can be chosen. The problem source is classified according to Brennan and Hulteen's [Brennan and Hulteen 1995] eight categories of human-machine communication failure (presented earlier). Given such a diagnosis, a repair plan can be selected. The most appropriate repair plan may depend not only on the source of communication failure, but also on the type of dialogue system. For instance, in a tutoring system, pointing out system limitations and referring to on-line help may be appropriate, whereas in an operational flight simulator, the system may just enforce the system limitations. – After

selecting a repair plan, user and system collaborate on executing it. Repeated repair may be necessary if the user fails to conform to the selected repair plan, or if the system introduces more errors while automatically interpreting the user repair input.

Recently, initial attempts have been made to abstract a more general framework for dialogue design [Dybjkaer, Bernsen et al. 1996], and to objectively evaluate dialogue systems (e.g., [Kamm and Walker 1997]). Evaluation issues will be discussed in more detail in the evaluation chapter of this dissertation.

2.4.3 Previous Work on Interactive Multimodal Error Correction

The closing section of this literature review summarizes previous work on *multimodal error correction* - the approach this dissertation pursues further.

With speech and handwriting recognition technology maturing, multimodal systems have attracted a lot of interest. Until recently, there was no work that explored the benefits of multiple modalities in the context of error correction. The first subsection summarizes Rhyne and Wolf's general discussion of error correction in *recognition-based interfaces* (i.e. interfaces that automatically interpret user input in speech, handwriting, and other human input modalities). They suggested that repeated errors may be a problem if error correction is limited to one modality, and that switching modalities may avoid repeated errors. The following subsection reports two user studies with a simulated system which was performed while this dissertation work was under way: The first study investigated how multimodal flexibility could improve error resolution in speech recognition applications. The second study explored in how much users can comply with constraints that are imposed on multimodal interaction (to increase system performance, similar to the idea of user convergence presented earlier in Section 2.3.3, page 39). The third subsection reports work on multimodal repair at the Interactive Systems Laboratories prior to this dissertation. The section closes with remarks on the interactive error correction methods offered in current commercial dictation systems.

2.4.3.1 Error Correction in Recognition-based Interfaces

Rhyne and Wolf [Rhyne and Wolf 1993] discussed a range of problems in the design of recognition-based interfaces. Their work also mentioned the problem of error correction. Error correction is initiated after an error has been detected, based on recognition feedback. Three error correction strategies are identified: editing the recognition result, selecting the correct result from a set of alternatives, and adapting the recognizer to learn from an error.

Rhyne and Wolf noted that error correction involves three steps: specifying the scope of correction (e.g., by selecting misrecognized words in the visually presented recognition result), specifying the correction command (e.g., delete, insert, or replace), and providing the correction input. They pointed out that repeated errors may occur, especially if correction is performed using the same modality as for the input. The problem of repeated errors is aggravated for speech input, since the speaker's intuitive response to errors (speaking more slowly and in an overly correct manner) typically further deteriorate recognition performance. They suggested to avoid repeated modality-specific errors by switching to a different modality for correction. The optimum correction dialogue may combine several modalities. The trade-offs among different methods could be calculated based on correction time estimates for each method.

Rhyne and Wolf's general discussion captures the essence of interactive multimodal correction. The main contribution of this dissertation is to get these general ideas to work. The performance model presented later in Chapter 7 builds on the idea of comparing different correction methods based on the time necessary to successfully complete a correction. The following subsection presents two simulation studies that confirm multimodal correction may be useful.

2.4.3.2 Simulation Studies on Multimodal Error Resolution

Work by Oviatt et al. [Oviatt and VanGent 1996] explored potential benefits of multimodal error correction. Offering multiple modalities in corrections is attractive because the set of

words which are difficult to recognize varies across different modalities. Therefore, if a word was misrecognized in one modality, it could be more easily recognized in some other modality. Using a Wizard-of-Oz simulation, error correction on an interactive service system for conference registration and car rental transaction services was simulated. Repeated errors which required 1-6 attempts to correct. The results suggested that user "naturally" switch modality in error correction if given the possibility. Also, switching modality alleviates user frustration in repeated failures. Furthermore, this study provided evidence that multiple modalities are not used simultaneously, in a redundant fashion, but in a contrastive manner. The design of multimodal correction in this dissertation makes use of this observation. Although these results are based on simulations, and although self-reports are known to be unreliable, this study provides scientific motivation for the approach pursued in this dissertation. Furthermore, by building a system that implements and extends the ideas explored in this study, and by evaluating this prototype in user studies, this dissertation will empirically test the study's hypotheses.

In another simulation study, Robbe et al. [Robbe, Carbonell et al. 1996] investigated whether users would comply with constraints that limit how they can interact with a multimodal interface. They observed that repetitions or reformulations - employed as error correction strategies by the research participants - frequently are misrecognized again. The authors therefore suggested that switching to other modalities in repeated recognition errors should be a good strategy.

2.4.3.3 Previous Work on Implementing Multimodal Interactive Correction

At the Interactive Systems Laboratories, work on implementing interactive multimodal error correction has begun prior to this dissertation. McNair and Waibel [McNair and Waibel 1994] describe a method to select an error using speech (*automatic subpiece location*), and a method to interactively correct errors by either speaking again (*spoken hypothesis correction method*), or by spelling the misrecognized words verbally (*spelling hypothesis correction method*). All

these methods assume that the application must get every word correct, such as in dictation applications. One current commercial dictation system uses the automatic subpiece location method (see below).

For automatic subpiece location, the user input is recognized using a language model that allows all substrings of the first-best hypothesis for the original utterance, but nothing else. The two correction methods proposed by McNair assume that the correct hypothesis is somewhere in the list of alternative hypotheses (or word lattice) of the original utterance. This assumption is a severe limitation, as will be discussed in more detail in the next chapter. The user repair input (which may be either continuous speech or verbal spelling) is recognized using a language model that is limited to substrings found in the N-best list for the repaired section. This N-best list is rescored using scores from recognizing the repair input.

2.4.3.4 Interactive Correction in Commercial Dictation Systems

Current commercial continuous dictation systems offer correction methods that are variations of methods discussed in the literature review.

In Dragon System's continuous speech dictation product Naturally Speaking®, the user locates recognition errors using the automatic subpiece location methods. For error correction, respeaking, choosing among alternatives and typing are available. The only difference between Dragon's selection method and the automatic subpiece location method is how the system is set to "selection mode": in McNair's implementation, the system enters selection mode by mouse click, whereas in Naturally Speaking®, selection mode is entered by saying the keyword "Select".

In IBM's continuous dictation product ViaVoice®, errors are selected using the mouse, and errors are corrected by choosing from alternatives or by typing. To get the system to learn from an error correction, the user has to issue a menu-command that triggers a specific correction dialogue box.

In summary, although the literature contains several discussions of the problem of repair in speech recognition applications, only few researchers have actually built and evaluated repair methods. Most are limited to the correction methods that are employed in current speech recognition applications. The multimodal approach (that this dissertation pursues) has been mentioned in two publications; but one of them is only a very general discussion of the problem, and the second only predicts that multiple modalities may expedite error correction based on a simulation study, without engineering and evaluating an actual system. The review of the extended literature on repair in human-human communication, which is presented in this chapter and the Appendix B, provides a useful framework to structure research on the repair problem in speech recognition applications, and motivates why a collaborative approach can be considered natural and intuitive. From this review, we identified a range of techniques that a developer of a new speech recognition application can apply to minimize recognition errors, by working on both the recognition algorithms and the interface design.

Part 1: Technology

Not only the study of interactions, but also new technologies are a major component of research on human-computer interaction in general, and the investigation of recovery from recognition errors in speech recognition applications in particular. This part is devoted to the technology of multimodal interactive error recovery in speech recognition applications. The three steps leading to the development of a multimodal speech recognition application are described: the multimodal component technologies that are used to interpret multimodal input streams automatically, the technology of multimodal interactive error recovery that builds on these components, and the integration of the technology in a (potentially) useful prototypical application.

Chapter 3 "Multimodal Component Technologies" describes the multimodal component technologies necessary for multimodal interactive error recovery, as proposed in this dissertation. It reviews automatic recognition technology for speech (continuous speech and spelled sequences of letters), handwriting, and pen-drawn gestures. For each modality, the description addresses three main issues: the state-of-the-art in recognition technology, details of the specific recognizers used in this dissertation work, and factors that determine recognition performance. These factors play an important role in understanding why recognizing corrections is difficult, and why interactive error correction works.

Chapter 4 "Multimodal Interactive Error Recovery" describes algorithms for interactive multimodal error recovery in general terms, without addressing application-specific issues. A generic interactive error correction algorithm is presented, as well as multimodal methods to interactively detect, locate, and correct errors. Several algorithms that increase correction accuracy by correlating correction input with repair context are proposed and evaluated on a database of multimodal interactive error corrections which was collected as part of this dissertation work.

Chapter 5 "A Multimodal Dictation System Prototype" presents the prototype multimodal dictation system that was developed to demonstrate and evaluate the concepts and algorithms of

interactive multimodal error recovery. A multimodal dictation system consists of a large vocabulary dictation recognizer enhanced with interactive multimodal error recovery. This chapter describes how multimodal error correction was implemented in this specific application.

3. Multimodal Component Technologies

Multimodal interfaces as understood in this dissertation integrate speech recognition with other input modalities. Such a multimodal system typically consists of various recognizers that are capable of processing a certain (input or output) modality, and of one or more integrating modules. The integrating modules control the sampling of a stream of multimodal input data, delegate recognition to the appropriate component or components, receive recognition results, and initiate appropriate action. Recognizers for speech (possibly specialized in different types of speech), handwriting and pen-drawn gestures are the main recognition subsystems for this dissertation work. The following sections provide a brief overview of the state-of-the-art in recognition technology for each of these modalities. Additionally, the specific recognizers are described that were used to build the multimodal dictation system prototype in this work.

3.1 Speech Recognition

The central building block for an automatic dictation system is obviously a large vocabulary continuous speech recognizer. The first subsection explains the principles of continuous speech recognition and some important characteristics of continuous speech recognizers using the example of the JANUS recognition toolkit [Rogina and Waibel 1995], which was used in this work. Spelling words aloud is another speech input modality. It will play an important role as an alternative to continuous speech for error correction. In principle, spelling can be recognized using any continuous speech recognizer by simply training it on a database of spelled words instead of a database of continuous speech. However, spelling recognition performance is better when specialized recognizers are used; recognition is more accurate and

processing more efficient, compared with a general purpose continuous speech recognizer. The second subsection describes the specialized connected letter sequence recognizer employed for this dissertation work.

3.1.1 Large Vocabulary Continuous Speech Recognition

The JANUS Recognition Toolkit [Rogina and Waibel 1995; Finke and Rogina 1997] is a speaker-independent, large-vocabulary continuous speech recognizer. The basic design is very similar to a typical state-of-the-art Large Vocabulary Recognition (LVR) system as described in a recent review [Young 1995]. However, before describing the details of JANUS, we begin with a rough outline of a generic large vocabulary speech recognition algorithm.

3.1.1.1 One Page Overview of Continuous Speech Recognition

In a typical continuous speech recognizer, the speech waveform A is converted into a sequence of acoustic feature vectors in a preprocessing step. Each acoustic feature vector represents a short interval of a few milliseconds of speech with various spectral features. The recognizer determines the most probable word sequence W, given the observed acoustic signal A, using a standard decomposition of the conditional probability (based on Bayes' rule):

$$\hat{W} = \arg max_W P(W|A) = \arg max_W \frac{P(W) \cdot P(A|W)}{P(A)}$$

The first term of the rightmost side P(W) represents the *a priori* probability of the sequence of words $W=w_1w_2...w_N$ and is determined by a statistical *language model*. The second term denotes the probability of observing the sequence of acoustic vectors given some word sequence and is determined by the *acoustic model*. The *search module* of the speech recognizer implements an efficient algorithm to find the most likely word sequence W that satisfies the above equation. A continuous speech recognizer thus consists of four main modules: preprocessing, acoustic models, language models, and search. Typical methods and algorithms underlying each of these modules for large vocabulary recognition include:

- Spectral analysis methods such as fast Fourier transformation and Cepstral transformation for preprocessing
- Hidden Markov Models (HMMs), Neural Networks (NNs) and hybrid HMM/NN architectures for context-dependent acoustic models
- N-gram statistical language models, especially trigrams
- and (dynamic-programming based) time-synchronous or stack-decoding search.

All current continuous speech recognizers limit the search to a sequence of words from a given vocabulary. For each word in the vocabulary, a (phonetic) dictionary determines the sequence of phones that represent the word. If the speech input contains a word outside the vocabulary, it is mapped nonetheless onto some sequence of words within that vocabulary, resulting in one or more recognition errors.

Having reviewed the basics of continuous speech recognition, the following subsection examines in more detail the continuous speech recognizer utilized in this dissertation.

3.1.1.2 Large Vocabulary Continuous Speech Recognition with JANUS

This dissertation work employed the JANUS recognition toolkit that was trained on read speech from the Wall Street Journal (WSJ) database [Rogina and Waibel 1995]. The following paragraphs describe more specifically the preprocessing, acoustic models, language model, and search module of the JANUS WSJ system.

Input speech is digitized at 16 kHz. Sixteen melscale coefficients are computed over a 16 millisecond-wide frame of speech every 10 milliseconds. The preprocessing of these frames of 16 ms of speech consists of a standard spectral analysis (resulting in a melscale fourier spectrum) and a linear discriminant analysis.

For acoustic modeling, continuous density HMMs are employed. Thus, each elementary acoustic unit is modeled as a mixture of a Gaussian codebook with one mixture weight. Most state-of-the-art systems limit the phonetic context to one phone to the left or right (so-called

triphones). The JANUS WSJ system however allows phonetic contexts of arbitrary length, i.e. two or three phones to the left or right of the current phone. To ensure enough training data for each model, these models are clustered to yield 2,000 to 5,000 sub-allophones (*polyphones*) [Finke and Rogina 1997]. The elementary acoustic modeling unit of the JANUS WSJ recognizer is therefore polyphones.

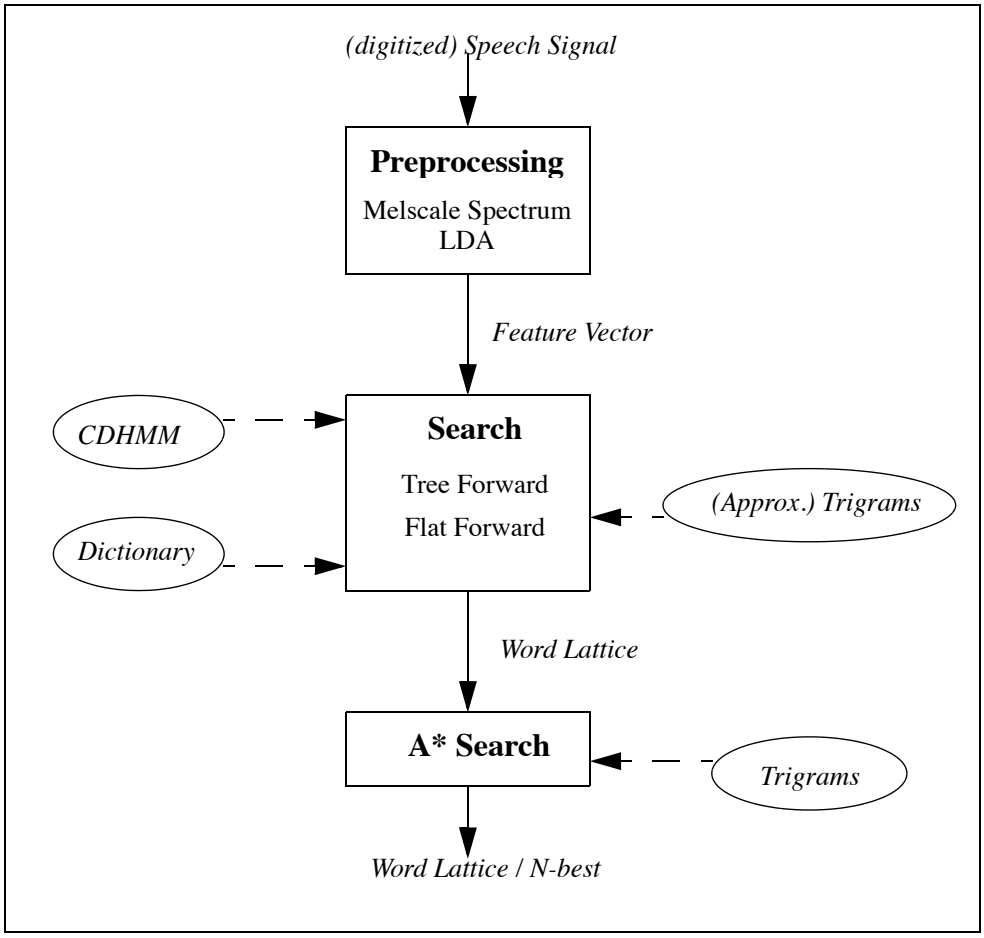


Figure 3-1. *JANUS Recognition Toolkit large vocabulary dictation recognizer*

The search consists of up to three passes: First, a tree-structured forward pass locates preliminary word boundaries and represents all possible segmentations of the speech input into words in a large word lattice. Second, an (optional) forward pass uses a flat dictionary structure to rescore the segmentations located in the first pass, with more sophisticated acoustic models. The first two recognition passes apply approximations of trigrams as the language model. The final pass performs an A* search and applies full trigrams as the language model. The recog-

nition output is a word lattice or an N-best list of alternatives. Figure 3-1 shows an overview of the JANUS WSJ recognizer.

In a recent Ph.D. thesis [Woszczyna 1998], several sophisticated search techniques were developed that reduce recognition time, without retraining the acoustic models, from 50-200 times real-time in the evaluation system to close to real-time performance. The most important techniques employed are more aggressive pruning guided by phonetic and language model lookaheads, cross codebook bucket box intersection to reduce the cost of score computation in systems with very large continuous density HMMs (such as the polyphonic CDHMM Janus WSJ recognizer), and skipping frames in the score computation. Table 2 compares the performance of the evaluation system with the version that was tuned for speed on the standard NAB Wall Street Journal '94 evaluation test set (see Appendix C). As can be seen, near real-time performance can be achieved, albeit with a substantial loss of recognition accuracy.

Table 2: *Benchmark performance of JANUS large vocabulary (60,000 words) WSJ recognizer*

System	Testset	Word Accuracy	Real-time Factor
Evaluation	Nov'94	93%	~50
tuned for speed	Nov'94	80%	1.2

3.1.1.3 Factors affecting Performance of Continuous Speech Recognition

Factors affecting accuracy of continuous speech recognition have been mentioned in the literature review (see Section 2.1.1). For our discussion of interactive error correction, the following three factors are relevant: speaking style (isolated speech is more difficult to recognize with a continuous speech recognizer - contrary to intuition), speaking rate (fast speech is more difficult to recognize), and length of words (short words are more difficult to recognize). In addition, tuning a recognizer to real-time performance is still a serious challenge in any large vocabulary application, such as dictation. The following paragraph provides some quantitative data on the trade-off between recognition accuracy and speed, which any developer of a large vocabulary application must confront.

Speed must be traded off against accuracy in a continuous speech recognizer because searching the space of all possible word sequences is the main computational bottleneck. Pruning and beam search are the main heuristics employed to make that search computationally feasible. At any given point in the search, the beams determine whether a hypothesis is maintained or abandoned (pruned). Since the size of the beam determines how much of the search space is considered in the recognition process, changing the width of the beams is a powerful and simple technique to determine the speed-accuracy trade-off. Wider beams mean that larger parts of the whole search space are actually searched, increasing the computational cost but achieving increased accuracy. Figure 3-2 shows the speed-accuracy trade-off for the JANUS WSJ system by means of the beam parameter. Decreasing beam size speeds up recognition at the cost of accuracy, and increasing beam size slows recognition speed down, gaining accuracy. The beam parameter is thus implicitly connected to the x-axis in this graph.

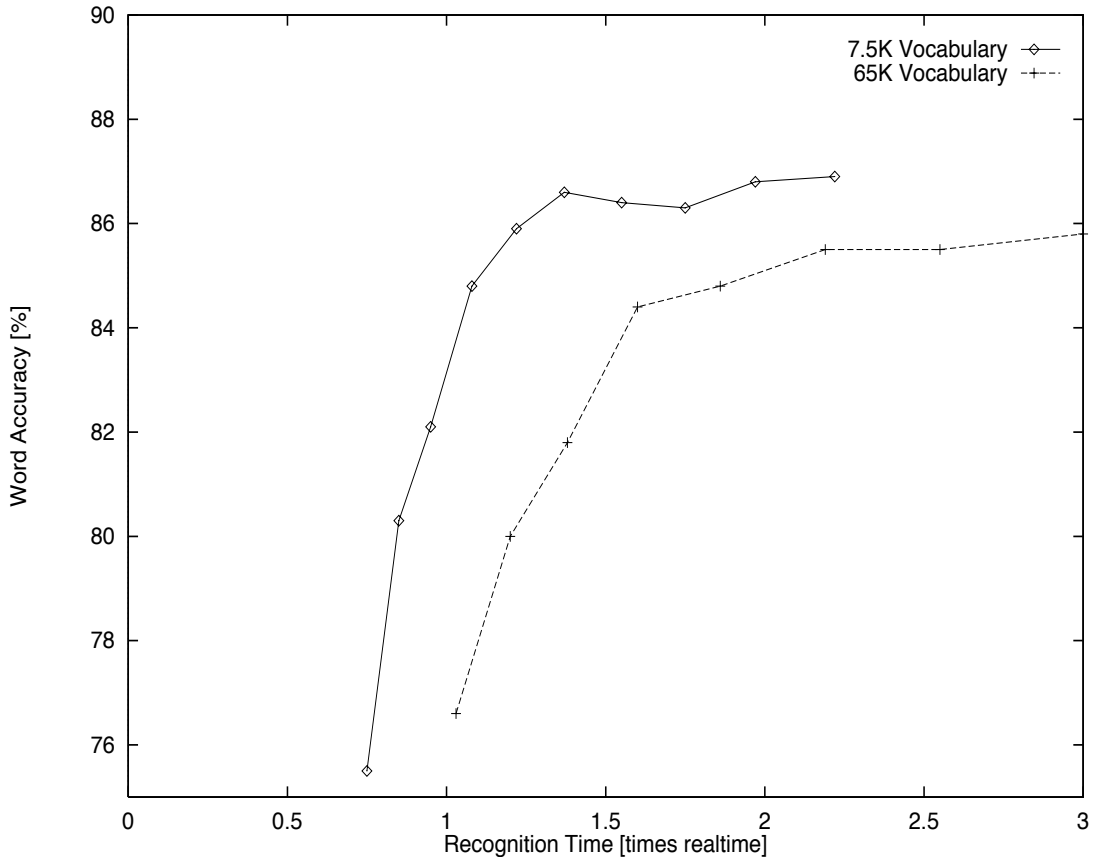


Figure 3-2. *Speed-accuracy trade-off for the JANUS WSJ recognizer*

3.1.2 Connected Letter Recognition

Recognizing sequences of spelled letters is difficult, despite a small vocabulary, because letters are easily confused - at least in most Western languages. Therefore, it is beneficial to develop recognizers that are specialized in connected letter recognition, rather than using a standard continuous speech recognizer trained on a database of spelled speech. NSpell is a speaker-independent recognizer specialized in recognition of (connected) sequences of letters [Hild 1997]. The next subsection presents an overview of the NSpell recognizer: its preprocessing, the special kind of neural network used for acoustic modeling, and additional constraints employed in the search to achieve high performance. The final subsection of this section presents performance results with a focus on factors that influence the performance of connected letter recognition.

3.1.2.1 Connected Letter Recognition with NSpell

Analogous to a continuous speech recognizer, a connected letter recognizer consists of three main modules: preprocessing, acoustic models, and search. The characteristics of each of these modules for the NSpell recognizer are described in the following paragraphs.

Preprocessing the audio input signal in NSpell employs standard techniques which are very similar to the JANUS WSJ system described in the previous section. Speech is sampled at 16 kHz and processed in frames at a rate of 10 ms/frame. For each frame, a fourier spectrum is computed and transformed into 16 melscale coefficients as the feature vector.

Using (artificial) neural networks for acoustic modeling distinguishes NSpell from a standard continuous speech recognizer. A Multi-State Time Delay Neural Network (MS-TDNN, [Haffner and Waibel 1992]) determines acoustic scores for each phoneme in any frame within the search. Similar to other neural network architectures, the TDNN consists of an input and a hidden layer. It differs from other networks because it uses a sliding window of several input features to calculate scores, both in the input and the hidden layer. Therefore, it can model

dynamic features of the input signal more effectively, which is crucial for signals that are acoustically highly confusable, such as connected letters. Phoneme-level scores are transformed into word-level hypotheses by modeling each spelled letter as a sequence of phonemes. Similar to most continuous speech recognizers, a dynamic time-warping (DTW) search identifies the best matching hypothesis as the path with the highest cumulative score.

NSpell can operate in two basic modes: recognition of arbitrary sequences of letters and recognition of letter sequences defined by a vocabulary. In the mode without vocabulary constraints, the search is helped by phonetic constraints inherent in any (Western) language: a statistical N-gram model [Jelinek 1990] assigns probabilities to each sequence of letters, thereby guiding the search for the best sequence of letters, similar to the use of N-gram models on the word level for continuous speech recognition. In the mode with vocabulary constraints, the search for the best matching sequences of letters can be modeled as a finite state automaton. A finite state automaton provides stronger guidance for the search than a statistical language model; therefore recognition with vocabulary constraints is more accurate. Additionally, such a search can be performed very efficiently utilizing tree data structures, hence the

name *tree search*. An overview of the NSpell system can be seen in Figure 3-3 below.

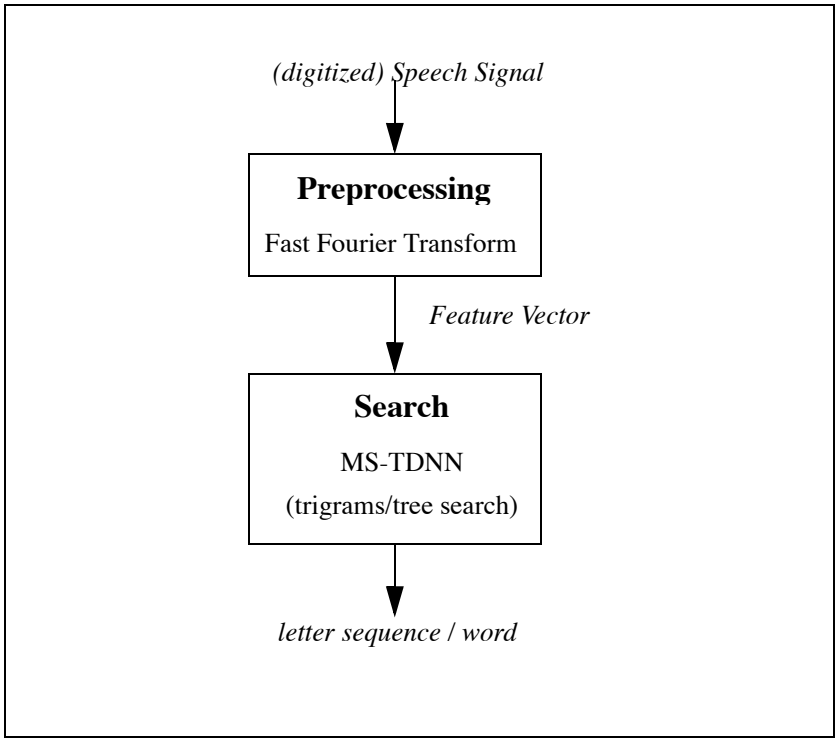


Figure 3-3. *NSpell connected letter recognition system*

3.1.2.2 Factors affecting Performance of Connected Letter Recognition

Based on Hild's extensive evaluation of NSpell in his dissertation [Hild 1997], this subsection identifies vocabulary size and word length as factors with a high impact on recognition accuracy of connected letter recognition, when constrained to a vocabulary.

The results in Table 3 show the impact of search mode on accuracy. The results were obtained on a test set of 685 spelled last names. In unconstrained search mode, i.e. without vocabulary constraints, an arbitrary sequence of words can be recognizerd with a word level accuracy of approximately 70% (see last row in the table). Although this word (name) accuracy may appear low, it corresponds to a letter-level accuracy of 92.5%. Such a performance could still

help to determine the spelling of new words in a continuous speech recognition application.

Table 3: *Benchmark performance of NSpell connected letter recognizer on spelled names (from [Hild 1997])*

Vocabulary Size	Name Accuracy
1,000	97.7%
100,000	94.4%
1,000,000	91.5%
14,000,000	89.3%
14,000,000 (no probabilities)	75.2%
Unlimited	70.2%

In the search mode constrained to a vocabulary, the size of the vocabulary has a significant impact on accuracy. Additional probabilistic constraints are crucial to achieve high performance for large vocabularies. In the first four rows, the search is biased with the empirically determined probability of each name. The second to last row reveals that without this additional constraint, 14% word (name) accuracy is lost. Therefore, either small vocabulary size or additional probabilistic constraints are necessary to achieve high performance.

Word length is the second important performance variable. Figure 3-4 reveals that short words are more difficult to recognize. Furthermore, the figure confirms that vocabulary size is an important performance variable.

In summary, vocabulary size and word length are important performance variables for connected letter recognition with vocabulary constraints. Probabilistic constraints (e.g., unigram biases) can ensure high performance with large vocabularies. In addition to vocabulary size and word length, accuracy varies significantly across different speakers. For instance, with a 20,000 word vocabulary, NSpell shows a standard deviation in the word accuracy of 5% across speakers.

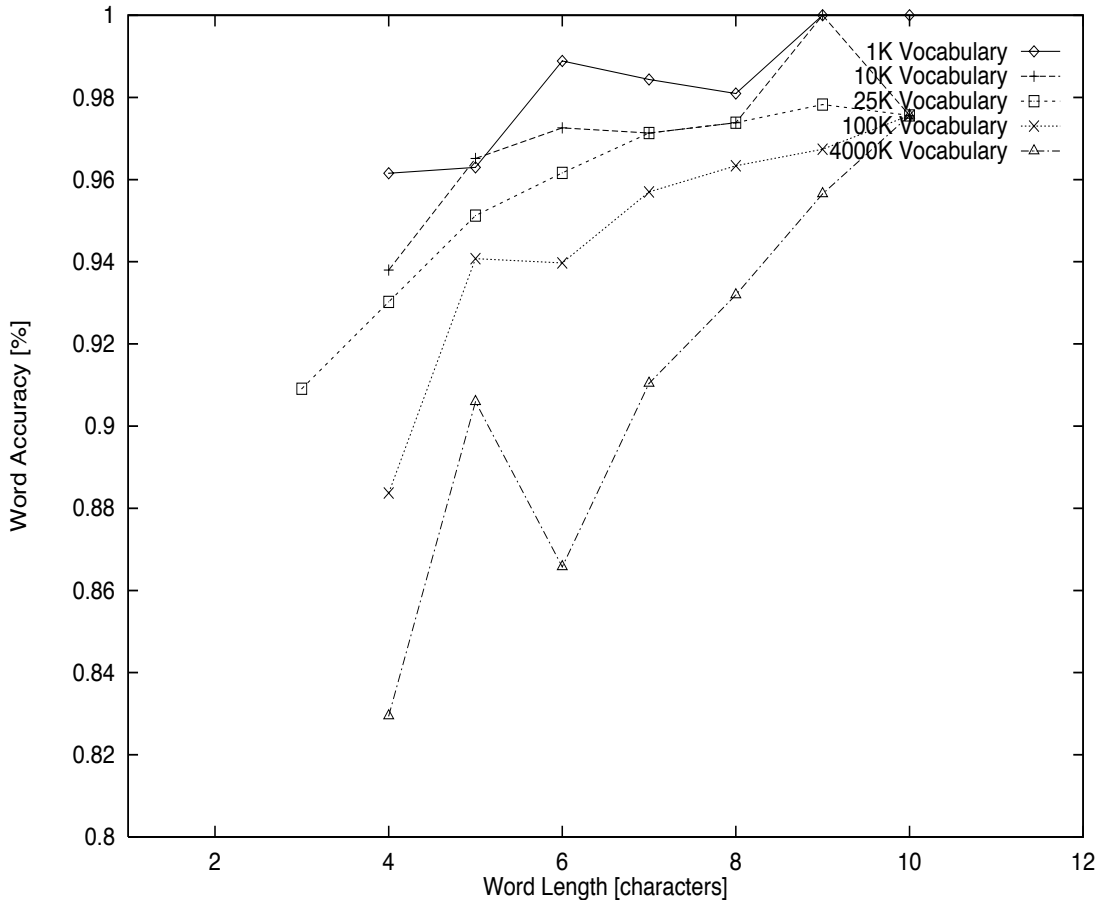


Figure 3-4. *Word length and vocabulary size as primary performance variables of connected letter recognition*

3.2 On-line Cursive Handwriting Recognition

After the overview of continuous speech and connected letter recognizers, this section considers on-line cursive handwriting as an additional input modality available in multimodal interfaces.

Written language recognition transforms language represented in its spatial form of graphical marks into its equivalent symbolic representation as ASCII text. Handwriting recognition shares many of the challenges of speech recognition. These challenges include: recognition independent of the author (writer-independent); segmentation at the level of characters or digits, words, or sentences; writing styles (printed vs. cursive, North American vs. European);

vocabulary size; and finally, For on-line handwriting recognition, the hardware must approximate the look and feel of paper and pencil as much as possible. Hardware considerations will be discussed later in the description of the prototype multimodal dictation system which was developed as part of this dissertation work (see Section 5.4).

The remainder of this section on handwriting recognition is organized as follows: the first subsection presents a brief overview of the state-of-the-art in handwriting recognition, main approaches, and published performance results. The second subsection describes in more detail NPen++, the on-line handwriting recognizer used in this dissertation work. Finally, factors affecting the recognition rate of on-line handwriting are discussed in the last subsection, using NPen++ as the example.

3.2.1 Overview of the State-of-the-Art in Handwriting Recognition

This overview of the state-of-the-art in handwriting recognition is based on several recent surveys [Nouboud and Plamondon 1990; Hildebrandt and Liu 1993; Govindaraju, Gyeonghwan et al. 1997; Manke 1998]. Recognition of isolated characters is feasible at 95% accuracy both for Western and Eastern languages, recognition of isolated words at 90-98% (depending on the vocabulary size), and recognition of handwritten sentences at close to 90%.

According to the mode of data acquisition used, automatic handwriting recognition systems can be classified into on-line and off-line systems. In *Off-line* systems (or Optical Character Recognition OCR), the handwriting is given as an image or scanned text, without time sequence information. In *On-line* systems, the handwriting is given as a temporal sequence of coordinates that represents the pen trajectory. Our discussion will focus on on-line systems because most applications in multimodal interfaces require on-line systems. First, the main paradigms of handwriting recognition (holistic and analytical) will be reviewed. Then, more details of each approach are described, including the performance of the most accurate systems.

3.2.1.1 Paradigms in Handwriting Recognition

Handwriting recognition can be at the level of isolated characters (or digits), at the level of words, or at the level of sentences. Some recent commercial products (e.g., 3Com's PalmPilot) require the user to learn a simplified alphabet (which makes recognition easier), but the recognition algorithms are similar whether the "native" or a modified alphabet is used[1].

Character recognition is a typical pattern recognition problem; shape and time features are extracted from the trajectory (given as time sequence or spatial representation) and are used to assign the trajectory to the appropriate class. Artificial Neural Networks (NNs), Hidden Markov Models (HMMs), and hybrid approaches (that combine neural network modeling techniques with Hidden Markov Models) have been successfully employed as classifiers for character recognition.

There are two basic approaches to word recognition: the *analytical* approach first identifies the individual characters (using character recognition methods) and then builds word-level hypotheses from character-level hypotheses. By contrast, the *holistic* approach identifies the word directly from its global shape. In both cases, constraining recognition to a vocabulary increases accuracy substantially.

Recognition of sentences builds on word-level recognition methods. In addition, language models are useful to incorporate statistical information about word sequences, similar to the use of language models in automatic speech recognition systems. For instance, a trigram language model increased the performance of an on-line handwriting recognition system with a 21,000 word vocabulary from 80% to 95% [Srihari and Baltus 1993].

The problem of handwriting recognition shares many of challenges of speech recognition. Therefore, it is not surprising that similar algorithms and techniques are successful. With adaptations of the preprocessing and topology of the basic modeling units, a continuous speech recognition system can be trained on handwriting data and achieve very reasonable

1. The PalmPilot employs a modified alphabet to achieve higher recognition accuracy.

performance. For instance, using the BYBLOS continuous speech recognizer without changing its algorithms, a word accuracy of greater than 95% was achieved on a 3,000 word vocabulary [Starner, Makhoul et al. 1994]. However, algorithms specialized in handwriting recognition yield higher performance.

Therefore, we will review the most successful specialized handwriting recognition algorithms. The next subsection briefly describes the different features that are extracted from the input image and that serve as input to the ensuing classification step. The following two subsections review algorithms to classify handwriting using these features: holistic and analytical approaches to word and sentence recognition.

3.2.1.2 Feature Extraction for Handwriting Recognition

Features for handwriting recognition can be classified into local and global features. *Local features* represent the main topological characteristics of a small subsection of the trajectory. *Global features* represent the relationship of different line segments within a trajectory. While local features are applicable to any character set, global features attempt to capture specific characteristics of certain character sets (e.g., strokes in Chinese characters). For detailed discussion of different local and global features, refer to [Hildebrandt and Liu 1993; Manke 1998].

3.2.1.3 Holistic Approaches to Handwriting Recognition

Holistic approaches to handwriting word recognition identify words directly from their global shapes; features are extracted from the global shape, and standard pattern classification methods are applied to assign the shape to one of the words within the vocabulary. Holistic methods must constrain the search to a given vocabulary, unlike some analytical methods that can recognize any word.

The following features have demonstrated their usefulness for holistic handwriting recognition: word contour (e.g., ascenders, descenders, holes, i-dots), length of word (e.g., estimated

by the number of crossing of the center line), and "significant" visual structures, called *graphemes*.

Additional methods are necessary to make holistic recognition feasible for large vocabularies. Lexicon reduction algorithms determine, from a large lexicon (vocabulary), a set of words that is likely to match some handwritten input [Madhvanath 1996].

Performance of holistic methods is sufficiently high for small vocabularies (e.g., 98% on a 10 word vocabulary [Farag 1979]). Lexicon reduction could used to apply holistic methods to large vocabulary tasks (a 3,000 word lexicon can be reduced to 50-100 words with a 95% accuracy [Madhvanath 1996]), but no such system has been published to date.

3.2.1.4 Analytical Approaches to Handwriting Recognition

Analytical approaches to handwriting word recognition first identify the constituent characters. Then, based on character-level information obtained in the first step, a second step identifies word-level hypotheses.

Analytical approaches can be classified further into approaches with explicit and implicit segmentation. Explicit segmentation, also called *OCR postprocessing*, has two distinct stages: the first stage identifies sequences of characters, and the second stage matches character sequences with ASCII representations of words. By contrast, approaches with implicit segmentation use a lexicon to drive the segmentation and the recognition process. The algorithm matches input with words within a given vocabulary in a single step, using both character and word-level information. While approaches with implicit segmentation have superior accuracy, they inevitably fail when the word input is not present in the given vocabulary (new word). OCR postprocessing can be extended to recover from the presence of new words.

The best published recognition accuracies for analytical handwriting recognition systems are more than 95% for character recognition [Guyon, Henderson et al. 1992], 93.4% for word recognition (with a 20,000 vocabulary) [Manke 1998], and 86.6% for sentence recognition (with

a 20,000 word vocabulary) [Manke 1998]. Each of these systems is writer-independent.

3.2.2 On-line Handwriting Recognition with NPen++

NPen++ [Manke, Finke et al. 1995] is a writer-independent, large-vocabulary on-line recognizer for cursive handwriting. As an example of an analytical approach to handwriting recognition, it uses implicit segmentation in a lexicon-driven search for the best matching word-level hypothesis. The following paragraphs describe some details of the preprocessing and recognition phase in NPen++.

The preprocessing applies several normalization techniques to remove undesired variability in the coordinate sequence. First, to compensate for differences in sampling rates and writing speeds, the coordinates that originally were sampled equidistant in time are resampled equidistant in space. After smoothing the trajectory, several lines characterizing word orientation are estimated: the baseline, and the lines demarcating ascenders and descenders. Using the directional information obtained from the baseline, the word is rotated to a near horizontal orientation. Finally, the word is rescaled to a normalized size using the distance of baseline and centerlines.

Feature extraction follows the normalization step. Two sets of features are calculated; local features describing (among others) the curvature of the trajectory and the writing direction, and global features capturing the context in low resolution, bitmap-like descriptions of coordinates that are close to the current coordinate.

The recognition module of NPen++ segments the trajectory into words in a single step, using a MS-TDNN as classifier. (The MS-TDNN is similar to the one employed in the connected letter recognizer NSpell.) A TDNN neural network determines a score for each character (out of 26 in the Roman alphabet) given a sliding window of five (in the hidden layer) times seven (in the input layer) input feature vectors. The search for the best word hypothesis combines character scores to word scores for each sequence of characters in the given vocabulary by

performing a standard non-linear time alignment (dynamic time warping, DTW). Figure 3-5 shows an overview of the NPen++ system.

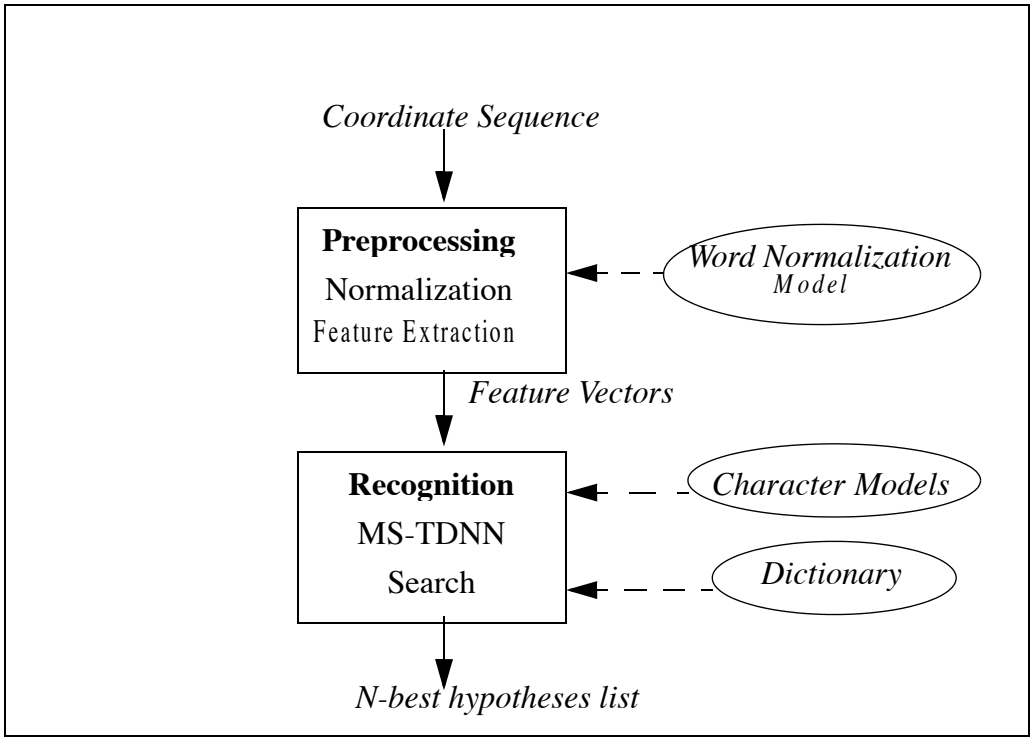


Figure 3-5. *NPen++ on-line cursive handwriting recognition system (from [Manke, Finke et al. 1995])*

3.2.3 Factors affecting the Performance of Handwriting Recognition

Using the NPen++ system [Manke 1998] as an example, this section identifies factors that influence performance of handwriting recognition. Similar to recognition of letter sequences, vocabulary size, word length, and the individual writer have a significant impact on recognition accuracy. In addition, writing style has a minor influence on accuracy. The following paragraphs quantify each of these aspects.

Table 4 shows the performance of NPen++ with different vocabulary sizes, on a test set of 2500 words chosen randomly from the standard dictionary for the NAB Wall Street Journal task (see Appendix C). The time performance is close to real-time on a fast workstation or PC. High performance of more than 90% requires vocabularies of no more than 20,000-40,000 words.

Table 4: *Benchmark performance of NPen++ on-line cursive handwriting recognizer on WSJ vocabularies*

Vocabulary Size	Recognition Accuracy
5,000	95.3%
10,000	93.4%
20,000	91.4%
100,000	82.9%

Figure 3-6 illustrates the dependence of recognition accuracy on the length of the input. As with recognition of spelled words, accuracy decreases on short words. There are two reasons for the difficulty of recognizing short words. First, short words provide less context information. Badly written or poorly modeled characters are more likely to cause recognition errors. Second, the normalization algorithms in the preprocessing phase work less accurately on short words. For example, it is more difficult to determine the baselines and the lines demarcating ascenders and descenders.

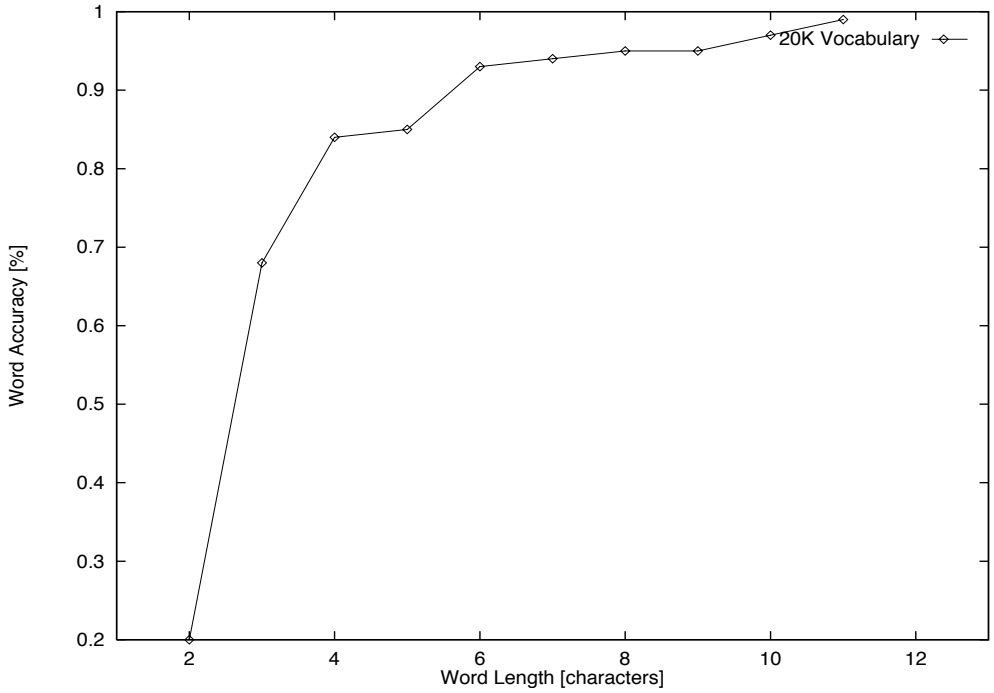


Figure 3-6. *Influence of word length on handwriting recognition accuracy (from [Manke 1998])*

The following example illustrates the high variation of recognition accuracy across different writers. Using a 40,000 word vocabulary, on a test set from 130 writers, the standard deviation of the word accuracy is 11.9%, with a mean accuracy of 80%.

Finally, the writing style (printed vs. cursive handwriting) has a minor influence on recognition accuracy [Manke 1998]. On a testset of German written words, the accuracy on the cursive subset was 90.8%, while the accuracy on the printed subset was 97.1%. On a testset of English words, the difference was much smaller (92.2% vs.vs. 92.8%). In accordance with intuition, printed words are easier to recognize than cursive written words.

3.3 Recognition of Pen-drawn Gestures

With the development of algorithms to recognize gestures, so-called gesture-based interfaces have become feasible. Gesture-based interaction with a computer offers an alternative to traditional interfaces driven by keyboard, menus, and direct manipulation input. Gesture-based interaction may appeal to both novice and expert users for a number of reasons [Wolf and Morrel-Samuels 1987]: objects, operations and optional parameters can be specified efficiently in the same movement, the learning and recall is facilitated since gestures tap into well-practiced paper and pencil behaviors, and using gestures is an obvious extension of direct manipulation interfaces which have improved significantly the usability of human-computer interfaces. In this dissertation, we are concerned with gestures that are drawn directly on a flat-panel display (subsequently called *pen-drawn gesture* or *pen gesture*). Our usage of the term gesture does not include gestures of fingers, hands, or the whole body in three-dimensional space (*3d gestures*). Such 3d gestures constitute a whole research field on its own; automatic recognition can be achieved either with dedicated devices (e.g., data gloves) or based on video images (by applying sophisticated computer vision algorithms). The following paragraphs review different approaches that have been developed for the recognition of *pen-drawn gestures*. There are two main approaches: hand-coded algorithms, and a feature-based approach using either decision trees or template-matching as classifiers.

While creating hand-coded gesture recognizers is feasible (e.g., [Coleman 1969]), it makes the resulting system difficult to create, maintain and modify. Since hand-coded gesture recognizers are useful only within the application for which they were created, they will not be discussed in any more detail. However, some hand-coded heuristics were developed in this dissertation to improve the performance of a generic gesture recognizer for the multimodal dictation system prototype.

Kankaanpaa [Kankaanpaa 1988] introduced a more general approach to gesture recognition that is illustrated in Figure . Similar to handwriting recognition, features are extracted from the gesture input, potentially using additional preprocessing steps. For example, Kankaanpaa's gesture recognizer applies some smoothing and filtering to the sequence of time-stamped sample points, and then computes features that characterize the shape, size, direction and orientation of the gesture. After preprocessing and feature extraction, standard pattern classification algorithms can be used to classify the gesture, including template matching, decision trees, and artificial neural networks. For example, Kankaanpaa used decision trees, while Vo's feature-based gesture recognizer employed template matching as classification algorithm [Vo 1998].

Rubine's algorithm [Rubine 1991] is another gesture recognizer that adopted the feature-based classification approach. It was used in this dissertation work for the gesture recognition module of the multimodal dictation system. His system allowed the application developer to specify gestures with small sets of examples (typically, 15-20 examples per gesture class are sufficient). Although it was designed for single-stroke gestures only, it can be applied without modification to multi-stroke gestures, provided the set of gestures does not contain (multi-stroke) gestures that are ambiguous when interpreted as a single-stroke gesture. Rubine's recognizer achieved a writer-dependent accuracy of 97% on gesture recognition problems with no more than 15 gesture classes (trained on around 40 examples for each gesture class).Architecture for feature-based gesture recognition system.

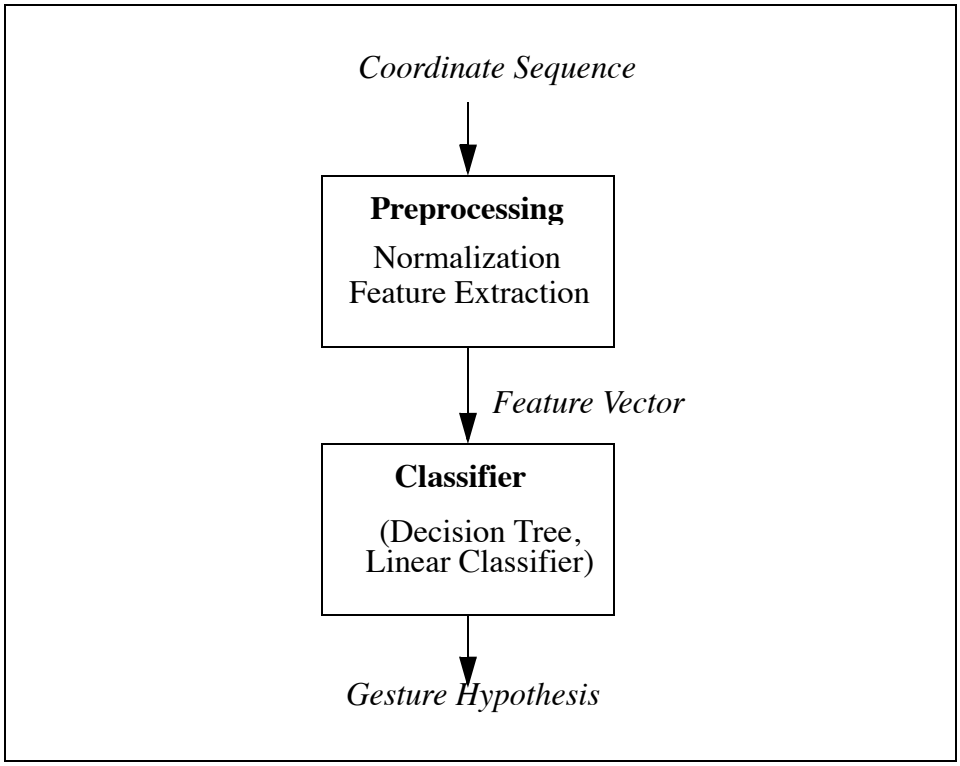


Figure 3-7. *Architecture for feature-based gesture recognition system*

Summarizing the whole chapter, we reviewed the state-of-the-art in the recognition technologies necessary for multimodal error correction: large vocabulary speech recognition, connected letter recognition, on-line cursive handwriting, and pen-drawn gesture recognition. The performance of these recognizers on standard benchmark tests indicates what recognition accuracies are feasible with current recognition technology; however, the performance on new applications may be quite different. Factors that determine recognition accuracy suggest why performance on some new application is different. Across different types of recognizers, these include vocabulary size and length of words. The difficulty of recognizing short words contributes to the difficulty of recognizing corrections by repeating: short words are misrecognized more frequently, and therefore, repeated input (in any modality) represents a more difficult recognition task than standard benchmarks. It is therefore not surprising that the accuracy of recognizing multimodal corrections (reported later) is well below the more than 90% reported in this section on standard benchmark tasks of continuous speech, connected letter,

and cursive handwriting recognition. However, although short words are more difficult to recognize across all modalities, there is still a significant accuracy gain when correcting multimodally, compared to unimodal correction by respeaking.

4. Multimodal Interactive Error Recovery

This chapter proposes *multimodal interactive error recovery* for non-conversational speech recognition applications with graphic user interfaces. *Interactive* error recovery means that the user collaborates with the system to recover from recognition errors. *Multimodal* error recovery means that the input modality can be switched for error correction. This main concept of multimodal error recovery was proposed in general terms in earlier work (see Chapter 2).

Interactive error recovery proceeds in two phases: *locating* and *correcting* errors. Different methods are available for each of these phases. Moreover, choosing the appropriate method depends on the application, because applications vary in the methods they offer. Methods by which to locate errors are presented in Section 4.2, and methods by which to interactively correct errors are presented in Section 4.3.

This thesis assumes that differential effectiveness of correction methods (across different error types, across different correction or editing tasks, and across different users) determines which tools users prefer[1]. Thus, the designer of a new speech recognition application determines the bag of multimodal correction tools that is feasible, given current recognition technology and application constraints, but the user (at least in principle) has choice among different correction methods.

Since recognizing correction input is difficult (as argued in the summary of the previous chapter), this dissertation presents algorithms that improve correction accuracy. The idea common

1. Effectiveness of correction methods as main determinant of user preferences will be quantified in the performance model of multimodal error correction presented in Chapter 7. The user study in Chapter 8 will provide some empirical evidence that correction accuracy is one of the main factors determining user preference.

to these algorithms is to correlate correction input with the (repair) context, instead of interpreting correction input as an independent event. This can be achieved in several ways. In this thesis, four approaches were developed and evaluated: word context modeling, bias towards frequently misrecognized words, correlation of N-best lists, and vocabulary reduction in partial-word corrections. These algorithms are presented in more detail in Section 4.4.

The best method of error recovery is obviously to avoid recognition errors in the first place. Once the user goes through the trouble of correcting errors, the rational user will expect the system to learn from the correction. Section 4.5 outlines some ideas how speech recognition applications can interactively learn from recognition errors. However, learning speech recognition algorithms involve research challenges that are beyond the scope of this thesis.

The following section describes multimodal interactive error recovery in very general terms. The details of the various steps and methods of multimodal interactive error recovery are explained later in Sections 4.2 through 4.4.

4.1 Multimodal Interactive Error Recovery Algorithm

This section describes multimodal interactive error recovery at the level of user interactions, application feedback, system components, and the flow of control between the components. The main components of an application that supports multimodal interactive error recovery include the multimodal components (as described in the previous chapter; in this thesis, recognizers for continuous speech, spelled letters, handwriting, and gestures), input/output modules, and integrating modules (e.g., the dialogue manager, the module implementing correction algorithms, and the application module). Figure 4-1 shows the flowchart of multimodal interactive error recovery. The following paragraphs describe the different steps of the generic multimodal interactive error recovery algorithm.

User interaction with a multimodal interface begins with user input in some modality. In speech recognition applications, such *primary user input* is frequently continuous speech. The

primary user input is automatically recognized using an appropriate multimodal recognition component ("continuous recognition" in the flowchart). Depending on the application, the "continuous recognition" module may range from solely a continuous speech recognizer (for instance, in dictation applications), to an array of recognizers specialized in different modalities (similar to the ones used for interpreting multimodal correction input shown in the flowchart).

After primary user input has been recognized and processed, the application provides some form of feedback on the recognition. This feedback may range from visual presentation of the recognition output (e.g., in dictation applications) to execution of the action intended by the user input (e.g., in an automatic flight booking system, retrieval of information on flights, and presentation of the results). After the feedback phase is completed, either the application or the user decides whether the recognition is accepted or whether a recognition error has occurred that requires correction. If the recognition is accepted, no repair is necessary, and user interaction with the application can proceed (cf. "repair done" in the flowchart). If an error is detected, correction interactions follow to recover from the error. Before correction, the exact location of an error within a larger sequence of input may have to be determined. The steps of *detecting* and *locating recognition errors* are described in more detail in Section 4.2.

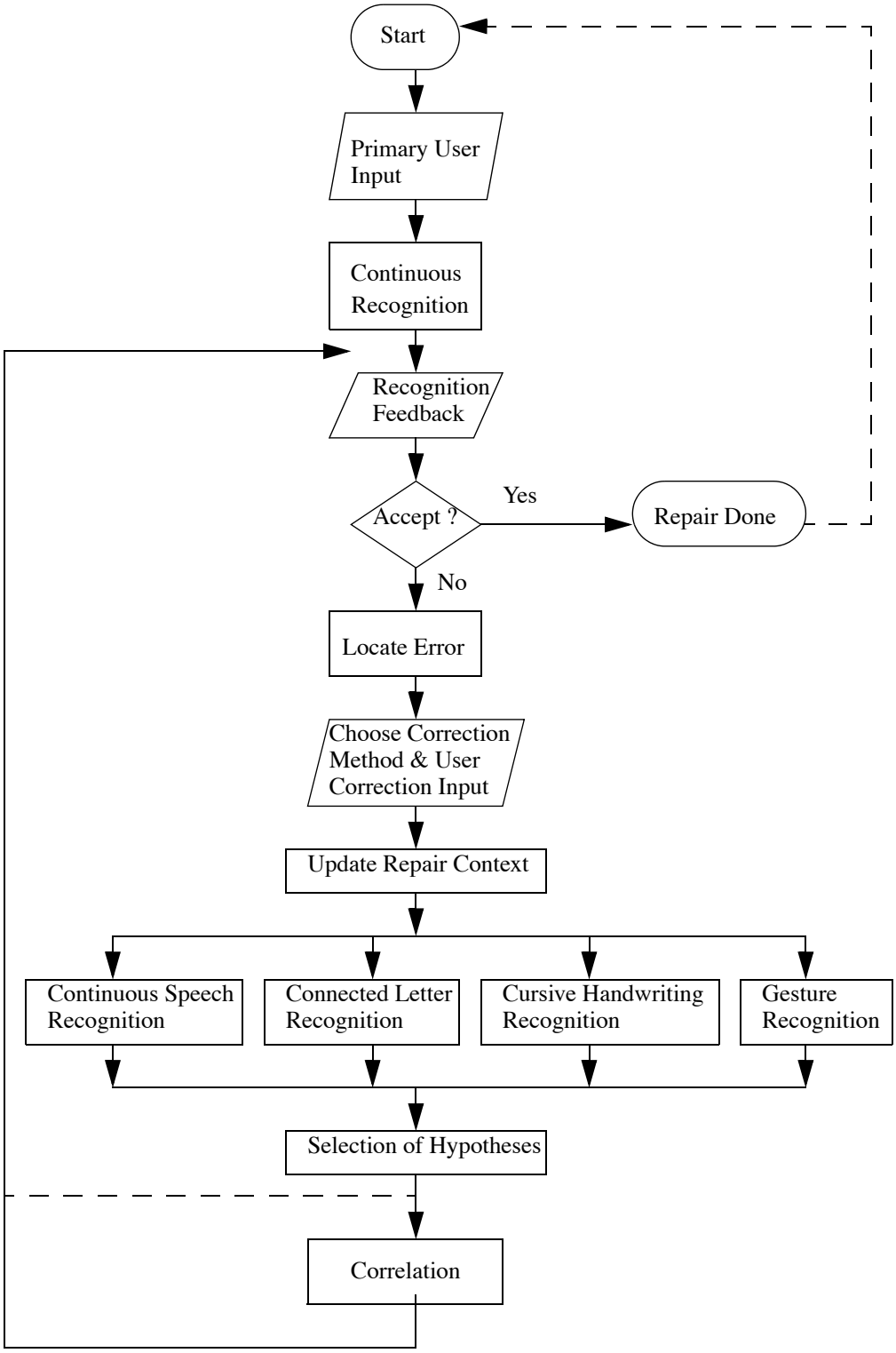


Figure 4-1. *Flowchart of multimodal interactive error recovery*

After an error has been detected and located, the user chooses a multimodal correction method from the bag of correction tools offered by the application, and provides the correction input required by the chosen method (e.g., speaking some words again). Section 4.3 presents the bag of correction tools that was developed in this thesis. While simultaneous use of several modalities for correction could be explored, a simulation study [Oviatt, DeAngeli et al. 1997] suggests that simultaneous use of modalities in multimodal human-computer interaction is rather infrequent, probably due to the inherently sequential nature of most human planning and acting. Therefore, this dissertation does not explore simultaneous use of several modalities for error correction.

The application delegates the recognition of this correction input to the appropriate multimodal recognition component. Before recognition is begun, the repair context is updated with the most recent primary user input, the recognition result, and information on the located error. This information may be used in the correlation step. The correlation step selects the recognition output from appropriate recognizers, and it increases the likelihood of successful correction by correlating correction input with the repair context. Algorithms for achieving this are presented in Section 4.4.

When the final hypothesis has been selected (with or without the correlation step), the application provides feedback on the completed correction attempt. The decision must be made whether the correction has been successful, or whether repeated correction is necessary.

The flowchart in Figure 4-1 provides an overview of the various steps in multimodal interactive error recovery. But how can errors be located and interactively corrected? What kind of algorithms can increase correction accuracy by correlating correction input with repair context? The remainder of this chapter addresses these questions.

4.2 Locating Recognition Errors

Before recognition errors can be corrected, they must be detected. When the recognized input consists of more than one elementary input unit, detection of an error may not be sufficient, the error may have to be located within the current input. Therefore, the first step of interactive error recovery - locating recognition errors - may involve two stages: error detection and identification of error location. This dissertation does not propose a novel methods to locate errors. However, this section briefly reviews available methods with which to detect and locate (recognition) errors. Before describing these methods in the next subsections, the following paragraphs discuss various general issues related to locating errors: what level of granularity is appropriate for error detection and location, and who initiates error detection.

The *granularity* of error detection and error location methods depends on three factors: elementary input unit, interaction goal, and type of recognition feedback. First, for different applications, different elementary input units may be appropriate; input may be on the level of characters or digits, words, phrases, or sentences (cf. Section 1.3.1). According to the level of input, errors must be detected either at the level of isolated characters, at the level of words, or at the level of whole sentences. Second, the interaction goal may range from data entry to issuing actions or conveying information. In data-entry tasks, any recognition error is relevant, and must be detected and corrected. By contrast, in tasks for which semantic accuracy is sufficient, recognition errors that result in semantically equivalent interpretations can be ignored. Third, recognition feedback ranges from very immediate and salient feedback by presenting the recognition result to the user, to implicit feedback by immediately executing the intended action. The type of feedback obviously has a large impact on what methods are appropriate for detecting and locating errors.

By analogy to Schegloff's model of repair in human-human dialogue (as reviewed in the literature review chapter in Section 2.2), errors in speech recognition applications can be located either by the user, by the system, or in collaboration of user and system. The following two subsections discuss different user- and system-initiated methods to locate errors, and indicate

for what kind of speech recognition application they are appropriate. Application variables that determine the appropriateness of methods include: type of recognition feedback (explicit visual or audible feedback versus implicit feedback by performing some action), type of required accuracy (verbatim versus semantic), whether the application suggests a conversational interaction metaphor, and whether a graphic user interface is possible.

4.2.1 User-Initiated Error Location

The user can detect and locate errors by pointing, by selecting with voice commands, or by using conversational error detection and location techniques. The following paragraphs briefly describe each of these methods.

User-initiated detection and location of errors by pointing (e.g., with the mouse, or by tapping on a touchscreen) is natural and effective if the application permits visual feedback. Using voice to detect and locate errors is possible and doesn't require visual feedback (and, thus, no graphic user interface is required). These methods lend themselves more naturally to non-conversational applications that require verbatim accuracy, although they could also be useful for some conversational applications, and applications that require semantic accuracy. Some commercial dictation systems offer user-initiated detection and location of errors by voice selection - the user can navigate through an editing buffer of words using voice commands such as "back N words" (e.g., in the disconnected speech dictation product DragonDictate®); the user can select one or more words by speaking them (e.g., McNair's automatic subpiece location method [McNair and Waibel 1994]); or using a voice keyword "select" (e.g., in Dragon's continuous speech dictation product NaturallySpeaking ®).

Errors can also be detected and located based on conversational error cues from the user. Such conversational cues typically include *paraphrases* and certain trigger phrases that people use when they have noticed a communication problem (e.g., "No, I meant Y..."). Paraphrases can be detected using mismatches between expectations derived by the application from the discourse context and the actual user input (cf. [Danieli 1996]).

4.2.2 System-initiated Error Location

System-initiated location of errors is possible based on user responses to requests for confirmation (mainly applicable to conversational speech recognition applications), or by automatically detecting and locating errors, for example, using confidence measures.

Requests for confirmation allow the system to detect recognition errors as follows. The system prompts the user to confirm whenever a significant input item has been processed, or immediately before the system takes some action. The user either confirms or denies the request, for example, via direct manipulation (e.g., by pressing a key in telephone applications), voice (e.g., by responding with "yes" or "no" to a request such as "Do you want me to connect you to Mr. X" in a call-routing application), or some other, more sophisticated method (e.g., in a multimodal interface with vision capabilities, by shaking or nodding of the head). A negative response to the request for confirmation indicates recognition errors. However, frequent requests for confirmation are annoying and therefore must be avoided, for example, by carefully calibrating such requests, according to the kind the application.

Confidence measures can be used to flag likely recognition errors in applications with explicit recognition feedback such as visual or audible presentation of the recognition result. This application of confidence measures is particularly attractive for data-entry and dictation tasks. For example, likely misrecognized words can be displayed in a different color, and the user can decide whether or not to select (and correct) the words. Such automatic highlighting of errors was integrated into the prototype multimodal dictation system which was developed for this thesis, to test the hypothesis whether imperfect automatic error highlighting can speed up error correction. More details on the implementation of automatic error highlighting can be found in Section 5.2.1, and the results of its evaluation can be found in Section 8.3.7.

4.3 Multimodal Interactive Error Correction

Corresponding to the three basic types of speech recognition errors - substitutions, insertions, and deletions - interactive correction must provide methods for substituting, deleting, and

inserting items. This section describes multimodal interactive methods for each of these correction tasks. The first subsection presents correction by repeating as a simple and effective method to substitute and insert items. For insertion corrections, location of the insertion must also be indicated. The second subsection proposes the use of pen-drawn gestures to indicate where to perform an insertion correction, and for other simple editing tasks. Finally, it may be more intuitive or more effective to perform corrections at the level of characters, rather than at the level of whole words. The third subsection introduces partial-word correction methods to perform corrections on the level of characters.

4.3.1 Correction by Repeating

Repeating input that has been misrecognized is a very simple and intuitive correction method. Repetition is the preferred correction method in human-human dialogue (see Section 2.2.3 in the literature review, and the review of repair in human-human dialogue in Appendix 3.3). Unlike in human-human dialogue where repetition in the same modality is generally a very effective correction method, repeating input in the same modality decreases the chances of success of automatic interpretation. The following paragraphs explain why repeating in the same modality is an ineffective correction strategy for automatic interpretation, and they introduce two approaches for effective *correction by repeating*.

Repeating in the same modality is ineffective for the following reasons: first, recognition errors are not random. Therefore it is likely that repetitions will continue to be misrecognized unless the recognition error was caused by deficiencies in the user input. Second, concerning repeating in continuous speech, people tend to hyperarticulate when repeating input (see Appendix B and [Oviatt, Levow et al. 1996]). Although people usually understand *hyperarticulated speech* better than normally pronounced speech, the performance of most speech recognizers deteriorates on hyperarticulated speech, in part because recognizers are trained only with normally pronounced speech.

This dissertation proposes two approaches to make correction by repetition effective: switch-

ing modality for repetitions *(cross-modal repair)*, and correlating correction input with repair context. Switching modality is effective because the words that are frequently misrecognized vary with each modality, with the notable exception of short words. If input is repeated in a different modality, chances for success are higher, albeit significantly lower than benchmark accuracy of the respective recognizers[1]. The following subsections illustrate correction by repetition in various modalities. The second approach for effective correction by repetition, correlating correction input with repair context, is presented later, in Section 4.4.

4.3.1.1 Correction by Respeaking (Repeating using Continuous Speech)

In correction by respeaking, the user simply respeaks verbatim the words (or items) that were misrecognized, and the located error (region) is replaced by a hypothesis for the spoken correction input. To improve correction accuracy, spoken correction input can be correlated with the repair context, as described in the flowchart of multimodal interactive correction presented earlier in this chapter. Figure 4-2 illustrates correction by respeaking. The user replaces the misrecognized word "one day" by speaking "Monday".

As mentioned before, if the primary input is continuous speech, the likelihood of success by respeaking is not very high. Therefore, *multimodal* interactive error correction offers to correct recognition errors by repeating input in other modalities. The following subsections describe two such *cross-modal* correction methods.

1. The evaluation and results in Chapter 8 will quantify this aspect. Since short words are more difficult to recognize in all modalities, correction input that replaces a misrecognized (short) word is more difficult to recognize than some arbitrary other input. However, the accuracy of repeating input in a different modality is still much higher than of repeating in the same modality.

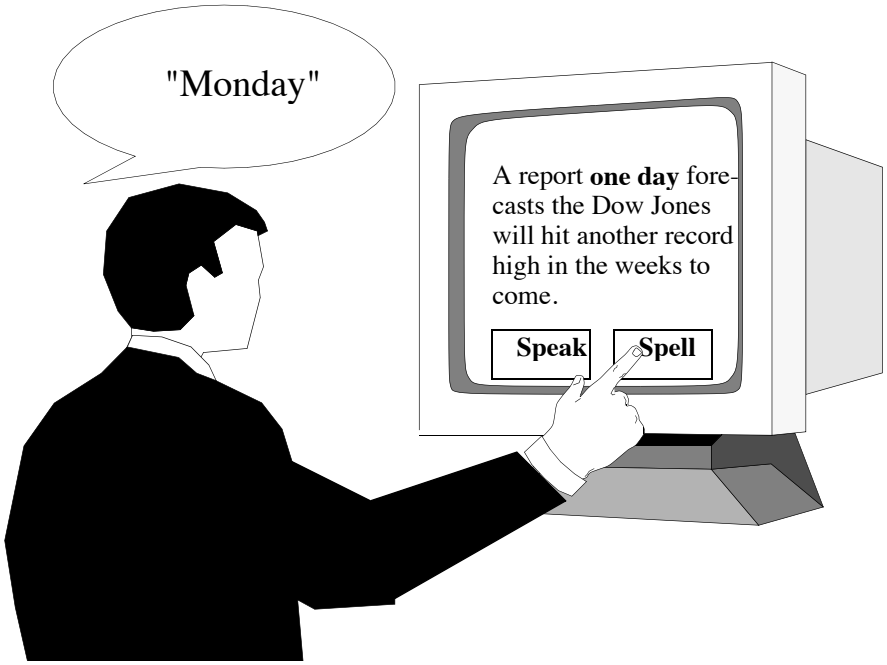


Figure 4-2. *Correction by respeaking (repeating in continuous speech)*

4.3.1.2 (Cross-modal) Correction by Spelling

In *correction by spelling*, the user repeats misrecognized words by spelling them as a sequence of letters. An example is shown in Figure 4-3; the user replaces "one day" by spelling "M-O-N-D-A-Y".

Accuracy of correction by spelling is significantly higher when a recognizer specialized in recognition of connected letters is used, and when corrections are constrained to all words within a given vocabulary (see Section 3.1.2). By using the same vocabulary as employed by the continuous speech recognizer, this limitation is kept consistent within the overall system. However, with this limitation, recognition errors that were caused by out-of-vocabulary words cannot be corrected using repetition by spelling. The new-word problem must be addressed separately. Section 4.5.2, later in this chapter, will describe some solutions to this problem.

Since people typically spell one word at a time, this correction modality is intuitively limited to isolated-word corrections. The user can correct multiple words as a sequence of several isolated-word corrections.

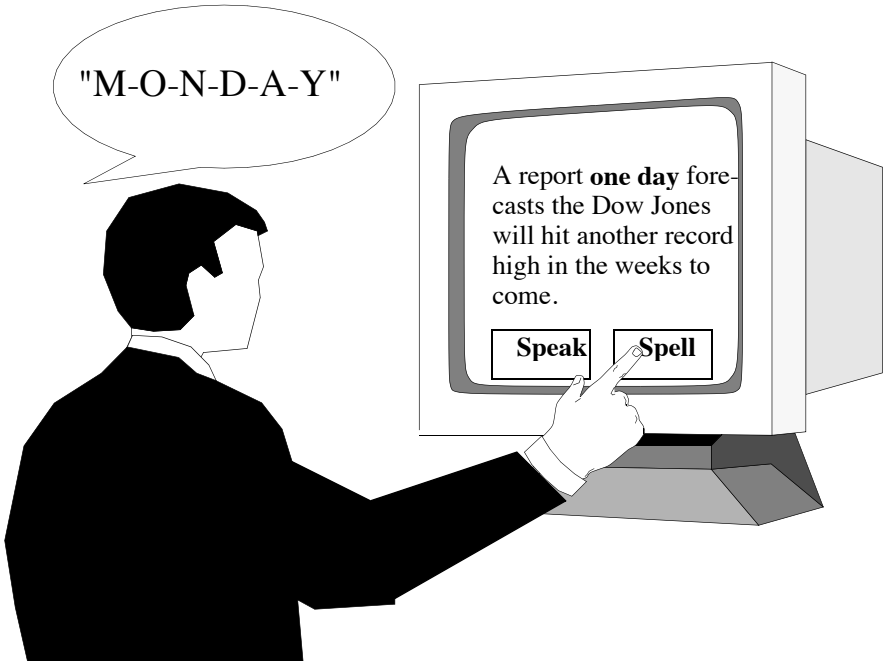


Figure 4-3. *Correction by spelling (repeating as spoken sequence of letters)*

4.3.1.3 (Cross-modal) Correction by Handwriting

In *correction by handwriting*, the user repeats misrecognized words by (hand-)writing them on a writing-sensitive display (e.g., touchscreen). Although users can learn to write on a vertically oriented standard desktop display, flat-oriented displays are easier to handle. Immediate visual feedback appears to be important to preserve the intuitiveness of handwriting as a correction modality, as has been learned in years of writing with pen on paper [Rhyne and Wolf 1993].

Just as with correction by spelling, the accuracy of correction by handwriting is significantly increased by constraining correction input to a given vocabulary. Unless application-specific requirements suggest otherwise, the same vocabulary should be used in all recognizers for equivalent types of correction input (e.g., recognizers for continuous speech, spelling, and handwriting).

Handwriting intuitively allows correction of both isolated and multiple words. However, the recognition technology is still significantly less accurate for handwritten phrases and sen-

tences than for isolated words (see Section 3.2.1). Therefore, correction by handwriting in this thesis is limited to isolated-word corrections.

4.3.2 Editing using Pen-Drawn Gestures

Correction by repeating (either in continuous speech, connected letters, or handwriting) addresses only two of the three basic correction tasks: substituting (replacing) recognition errors, and inserting deleted words. What remains to be addressed are methods to indicate the location of corrections by inserting, and methods to delete words. This section proposes *pen-drawn gestures* for editing tasks. Some commercial dictation products provide various forms of voice editing that appear to be quite efficient. Nevertheless, a formal comparison of voice editing with gesture-based editing was not performed as part of this dissertation.

Editing tasks that are part of error correction include deleting items, indicating where items should be inserted, moving items, positioning items, and formatting. Such editing tasks consist of two parts: selecting a command and indicating the scope of the command. Pen-drawn gestures appear to be intuitive and effective to issuing commands and indicating their scope (cf. [Wolf and Morrel-Samuels 1987; Rhyne and Wolf 1993]), because they combine the ease of referring to objects directly on the screen with employing marks to specify the type and scope of the commands. Such editing by pen-drawn gestures is applicable only to speech recognition applications with graphic user interfaces. The following paragraphs review results from a study on the use of pen-drawn gestures for editing tasks. The last paragraph in this section outlines how editing by pen-drawn gestures can be implemented within a multimodal correction system.

A paper and pencil study [Wolf and Morrel-Samuels 1987] investigated the use of *hand-drawn* gestures for simple editing tasks. The study focussed on identifying gestures that are *intuitively* utilized. The key finding was that people consistently used similar gestures - without explicit training. We can therefore expect that a multimodal application must support only a limited set of gestures for each editing task. Automatic recognition of limited sets of ges-

tures is sufficiently accurate. Figure 4-4 shows the most common gestures that Wolf's study identified for deletion, insertion, movement, and positioning tasks on the level of characters, words, and phrases.

Landay [Landay 1996] makes extensive use of pen-drawn gestures in his tool for rapid interface development. Interfaces are sketched on the screen, and interface widgets are recognized directly as they are being drawn. Several interface elements and screens can be combined to model execution of complete tasks in storyboards. The sketched design can be executed directly, thus saving the step of implementing the design using an interface prototyping toolkit (such as MS Director, or Hypercard).

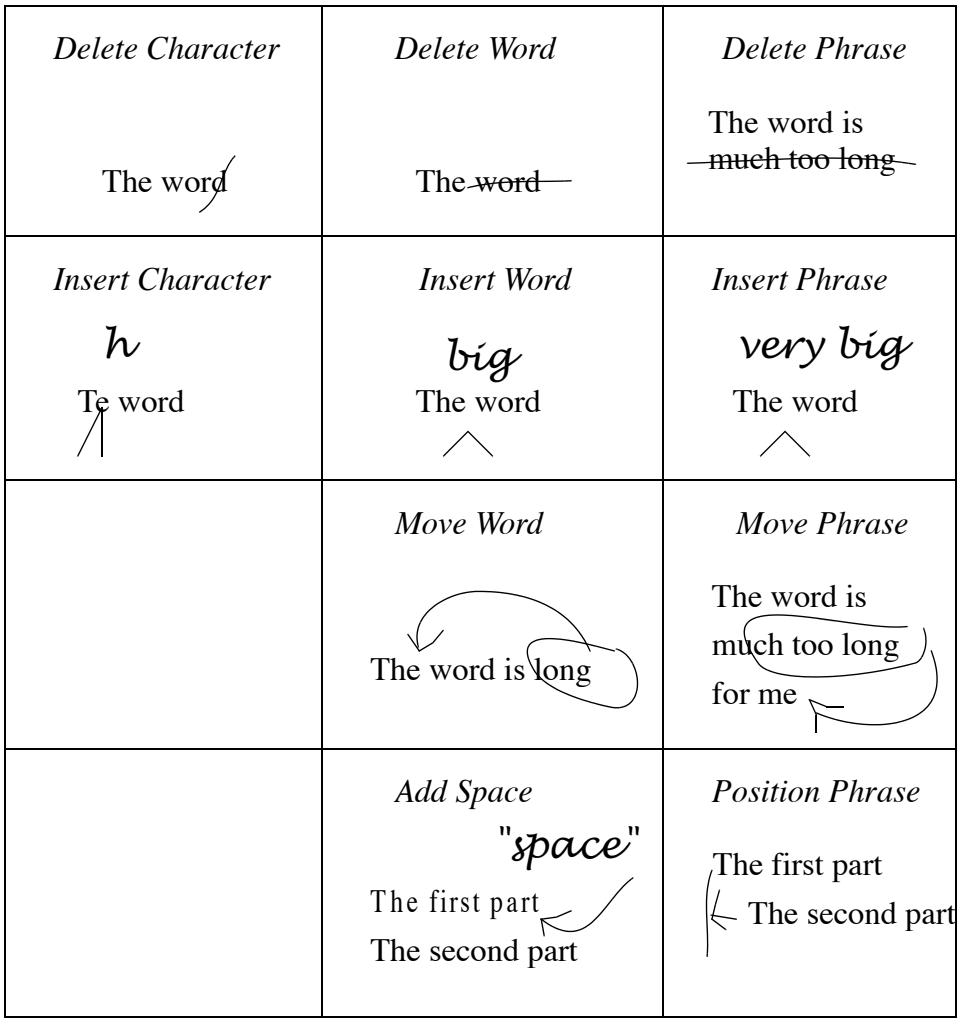


Figure 4-4. *Common gestures for simple editing tasks (from [Wolf and Morrel-Samuels 1987])*

How can such editing gestures be supported in a multimodal application? Gestures can be classified by means of a gesture recognizer as described in Section 3.3, page 71. Integration of the gesture recognizer into the overall system requires the following functionality: communication with a gesture recognizer, automatic distinction of gestures from handwriting input, and disambiguation of object references and gesture scopes. Communication with a gesture recognizer is analogous to communication with a handwriting recognizer. Algorithms to distinguish handwriting from gesture input are presented in Section 5.3.2, page 125. Spatial relationships between displayed items and the gesture trajectory can be used to disambiguate object references and gesture scopes. For example, for a deletion gesture, the system must determine which of the displayed items the user intends to delete. The next chapter describes how these problems were solved for the multimodal dictation system prototype, and identifies what set of editing tasks and gestures this prototype supports.

4.3.3 Partial-word Correction

This section introduces methods with which to correct characters within a word (so called *partial-word correction*), as an alternative to correction methods at the word level. Error correction can be performed at different levels: on the level of whole sentences, phrases, isolated words, or characters within a word. Which level is appropriate depends on the application (or rather, the task within a speech recognition application), considerations of efficiency, constraints from the recognition technology, and constraints of the input (correction) modality. As an example for a modality constraint, it is very intuitive to speak multiple words, whereas it is not intuitive to spell multiple words. As an example for an efficiency issue, it may be faster to correct only the one or two letters that are incorrect in a misrecognized word, rather than having to repeat the whole word.

Partial-word correction includes methods to delete, insert, and substitute characters within a word. Methods to delete, insert, and replace whole words can be generalized to partial-word

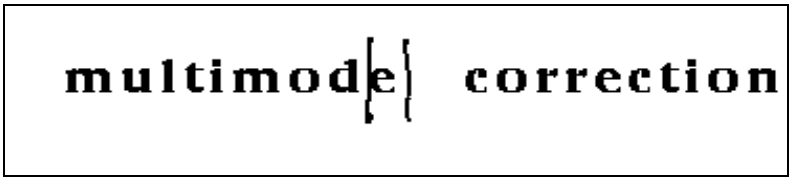
corrections by applying to the level of characters within a word similar operations to those used at the word level. For example, a partial-word deletion is issued by covering parts of a word, rather than the whole word, with a deletion gesture. For a partial-word insertion, the same gesture that is employed to indicate the location of a word-level insertion can be used to indicate the point at which characters should be inserted within a word. (However, the scope of insertion marks within a word is difficult to determine.) For partial-word substitution, the user must be able to select letters within a word. Then, just as with partial-word insertions, the correction input (intended to replace the selected characters) is provided in the same way as for word-level corrections.

Figure 4-5 illustrates partial-word correction. The upper part shows an example of a gesture to select characters within a word (in this case, the "e" at the end of the word "multimode"), and the lower part shows a partial-word substitution by handwriting (in this case, replacing the word "multimode" with the word "multimodal").

The interpretation of partial-word correction requires extensions of both the recognition subsystems and the integrating modules of the multimodal correction system. The recognition subsystems must be extended to handle character sequences, in addition to (sequences of) whole words. Recognition of arbitrary sequences of characters is possible both for spelling and handwriting. However, recognition accuracy is significantly lower than it is for recognition constrained to a (word-level) vocabulary (see Chapter 3 on the multimodal component technologies). Section 4.4.5 later in this chapter will describe an effective way to reduce the vocabulary size for partial-word corrections.

The main additional challenge for the integrating modules is to distinguish between partial-word and whole-word corrections. The scope for the different types of correction (substitution, insertion, deletion) implies the user intends a word-level or partial-word correction. If the scope of the correction method is at the level of characters within a word, the system switches to partial-word correction mode; otherwise, word-level correction is assumed.

Double-bar gesture to select characters within a word:



Partial-word correction by handwriting ("e" is selected):

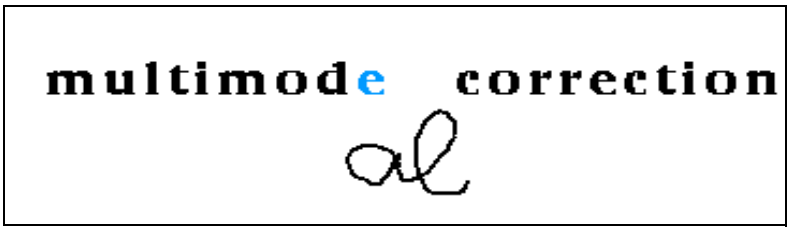


Figure 4-5. *Partial-word correction by handwriting*

Partial-word corrections rely on visual feedback, and that the application lends itself to presenting the recognition result to the user. Partial-word correction are applicable only to speech user interfaces with a graphic user interface.

4.3.4 Conversational Error Correction

Cross-modal methods for correction by repeating, pen-drawn gestures for simple editing tasks, and partial-word corrections, present an effective set of correction methods for predominantly non-conversational speech recognition applications with a graphic user interface. But what about conversational speech recognition applications, or non-conversational applications without a graphic user interface? The taxonomy of speech recognition applications presented in Chapter 1 included important examples in these categories: interactive services via telephone, interactive TV, and smart rooms. Although this dissertation does not explicitly address such applications, the following section gives a rough outline of conversational repair methods. *Conversational error correction* offers methods to recover from recognition errors in a dialogue, similar to repair in human-human dialogue. The following paragraphs outline conversational repair methods.

In human-human dialogue, paraphrases are frequently used to recover from communication problems, particularly if repetition failed (cf. Section 2.2 and Appendix B). Therefore, using paraphrases to substitute misrecognized content or to insert missing content, should therefore be an intuitive correction method for conversational speech recognition applications. However, detecting a paraphrase repair, resolving the reference (i.e., which content the user intends to replace or insert), and determining the new contents, are difficult to interpret automatically - more difficult than recognizing the original input. Consequently, correction by paraphrasing is an area for further research.

Many published research systems and some commercial dialogue systems implement conversational correction methods through sophisticated dialogue control and robust processing of natural language input. Some of this work has been reviewed earlier in Chapter 2.

Multimodal interactive correction can be applied, with modifications, to some applications without graphic user interface. For example, modality can be switched between different variations of speech, such as continuous, discrete, and spelled speech. Furthermore, editing can be performed by using voice commands instead of pen-drawn gestures and pointing (e.g., "Delete", "Insert after X", "Replace X with Y"). The scope of the command is either implicit, or it can be indicated by using speech (e.g., "Insert N words backwards", "Delete Nth word backwards"). Some commercial dictation systems provide such voice-editing capabilities.

4.4 Increasing Repair Accuracy by Exploiting Repair Context

Interactive error correction is effective if the modality is switched for correction, although recognizing correction input is less accurate, compared with standard recognition benchmarks. Without modifications to the recognition algorithms, accuracies in the 90% range (which current recognizers achieve on standard benchmark tasks) cannot be achieved on correction input. To increase effectiveness of such cross-modal correction, or to make interactive correction effective when unimodal correction is preferred, this section presents algorithms that improve correction accuracy by correlating correction input with the (repair) context, instead of inter-

preting correction input as an independent event. Future work may explore efficient correction methods that combine several modalities. Correlation of correction input with the repair context can be achieved in several ways.

A very simple way is to avoid making the same mistake twice. Once an error has been identified for correction, this item can be excluded from the correction vocabulary. Ainsworth and Pratt proposed this idea in their work on interactive correction as "repeating with elimination" (see [Ainsworth 1992]).

This section describes other, more powerful algorithms. It begins by describing some simple methods to increase the accuracy of respeaking (correction by repeating in continuous speech). The speech recognition algorithm is modified by adapting certain parameters for correction input. In subsections 4.4.2 through 4.4.5, four algorithms to correlate correction input with repair context are developed and evaluated: N-gram context modeling, bias towards frequently misrecognized words, correlation of N-best lists, and vocabulary reduction in partial-word corrections.

The algorithms are evaluated on a database of multimodal interactive corrections. The database was collected during the user studies that evaluated multimodal interactive correction in the context of the multimodal dictation system prototype. The database contains data from both the pilot and the final study. These user studies are described in detail in Chapter 8. Table 5 summarizes the statistics of this database as they are related to the comparisons of correction accuracies presented in this section. Note that speech repairs can comprise of multiple words, therefore we distinguish the counts for repairs (which may be an isolated-word repair or a multiple word repair) from the word counts.

The evaluation of methods to correlate repair context and correction input uses the standard methodology of measuring word accuracies (as commonly used in the speech recognition field) by comparing the true input with the recognized hypotheses, but to distinguish primary input from correction input, we will use the term *correction accuracy* to denote word accuracy

on correction input.

Table 5: *Database of dictation input and multimodal corrections*

Type of Data	Items in Database
Initial Dictation	503 Sentences (9750 Words)
Respeaking (multiple words)	515 Repairs (1778 Words)
Spelling (word-level)	816 Words
Handwriting (word-level)	1301 Words
Spelling (partial words)	40 Corrections
Handwriting (partial words)	65 Corrections

4.4.1 Improving the Accuracy of Corrections by Respeaking

In previous sections, we suggested that recognizing corrections by respeaking is more difficult than primary continuous speech input (e.g., dictation input). Empirical data from our user studies confirms the hypothesis that respeaking is significantly more difficult to recognize than initial dictation input ($p<0.01$). This result appears to generalize across different continuous speech recognizers[1], but the magnitude of the effect will vary between different recognizers. An error analysis was performed to determine why the accuracy on respeaking was so low with the JANUS WSJ recognizer. The following paragraphs describe some simple measures that arose from the error analysis to significantly improve the accuracy of corrections by respeaking. The final paragraph argues that new recognition algorithms, specialized in hyperarticulated speech, are needed to achieve high accuracy on corrections by respeaking.

A few simple measures could significantly increase the accuracy of corrections by respeaking. According to a widely acknowledged rule of thumb for speech recognizers, the rate of insertion and deletion errors should be balanced for maximal recognition accuracy. The error anal-

1. To reject the hypothesis that only the JANUS continuous speech recognizer has low recognition accuracy on corrections by respeaking, we performed the following simple experiment. We compared the accuracy of recognizing initial dictation with recognizing respeaking corrections using a different state-of-the-art continuous speech recognizer (SPHINX, cf. [Lee 1990]) on the same data, without modifications to its recognition algorithm. The experiment confirmed that spoken corrections are significantly more difficult to recognize.

ysis revealed that although the rate of insertion and deletion errors was balanced on the initial dictation, corrections by respeaking had many more insertion than deletion errors. This problem was solved by using separate language model weights when recognizing corrections by respeaking. The separate weights are optimized to balance the rate of insertion and deletion errors on the subset of corrections by respeaking.

The error analysis also revealed that most recognition errors in the real-time version of the JANUS WSJ recognizer were due to search errors, rather than to acoustic model errors or language model errors (for an explanation of the different types of errors, and how to identify them, refer to [Chase 1997]). Search errors can easily be reduced by relaxing the pruning parameters of the search module, albeit at the cost of slowing down recognition speed.

Table 6 shows the improvement of accuracy on corrections by respeaking after balancing insertion and deletion errors (using separate language model weight parameters), and after reducing search errors through relaxing the pruning parameters.

Table 6: *Improving accuracy of corrections by respeaking (multiple word corrections only)*

Correction by Respeaking Recognition Method	Word Accuracy
baseline	40%
after reducing search errors	46%
after balancing insertion/deletion errors	52%

Some observations suggest that high accuracy is only possible by using specialized recognition algorithms for corrections by respeaking. First, corrections by respeaking contain many isolated-word corrections, and the performance of many continuous speech recognizers severely deteriorates on isolated words [Alleva, Huang et al. 1997; Soltau and Waibel 1998]. Including isolated word and disconnected speech data in the training data for the continuous speech recognizer increases the accuracy on isolated words, but it hurts the performance on continuous speech [Alleva, Huang et al. 1997]. Second, corrections by respeaking tend to be hyperarticulated, in speech recognition applications [Oviatt, Levow et al. 1996] just as in

human-human dialogue. This observation was confirmed in our study of error correction in the context of dictation. There are currently no speech recognition algorithms that work well on both normally pronounced and hyperarticulated speech. One could train two sets of acoustic models, one on normally pronounced and one on hyperarticulated speech, and switch between the two models whenever appropriate. In summary, it can be assumed that difficulty of recognizing isolated words and hyperarticulated speech accounts for the difference in accuracy between recognizing initial continuous speech input and recognizing corrections by respeaking.

4.4.2 Word Context Modeling

As first method to correlate correction input with repair context, (word) *context modeling* exploits the observation that once an error has been located, the input surrounding the error ca be assumed to be correct. The basic idea is to recognize correction input with the appropriate dependencies - the ones that are used in recognizing continuous input. This dissertation implemented this idea for word sequence input and dependencies modeled as statistical language models. However, the idea can be generalized to other types of input (e.g., digit sequences) and other dependencies between input items (e.g., digit sequences modeled as a finite state automaton). This algorithm is obviously not applicable to isolated-word input. The subsequent paragraphs formalize the *N-gram* context modeling algorithm that exploits dependencies modeled as statistical N-gram language model, and applies the algorithm to correction by respeaking, spelling, and handwriting. The effectiveness of N-gram context modeling is demonstrated on the database of multimodal corrections.

Statistical language models determine probabilities of word sequences. An N-gram language model factorizes the joint probability of a word sequence into a product of conditional probabilities, as expressed in the following equation (cf. [Jelinek 1990]):

$$P(w_1 \ldots w_N) = \prod_{i}^{N} P(w_i \mid w_{i-N+1} \ldots w_{i-1})$$

Correcting by replacing an error region (or also inserting words) can be formalized as follows. For simplicity, the notation assumes that a trigram language model is used (N=3). Let the error region (or *reparandum*) of (M+1) subsequent misrecognized words be denoted as $w_i ... w_{i+M}$, the word context to the left of the error region as $w_{i-2}\ w_{i-1}$, and the word context to the right of the error region as $w_{i+M+1}\ w_{i+M+2}$. Then, instead of recognizing the correction input as an independent event, N-gram context modeling recognizes the correction input as if it occurred in the context of the *pre-context* $w_{i-2}\ w_{i-1}$ and the *post-context* $w_{i+M+1}\ w_{i+M+2}$, enforcing the appropriate language model constraints. Figure 4-6 below illustrates this situation, as well as the basic idea of context modeling: exploiting the knowledge $F[w_{i-2}, w_{i-1}, w_{i+M+1}, w_{i+M+2}]$ in interpreting some repair input.

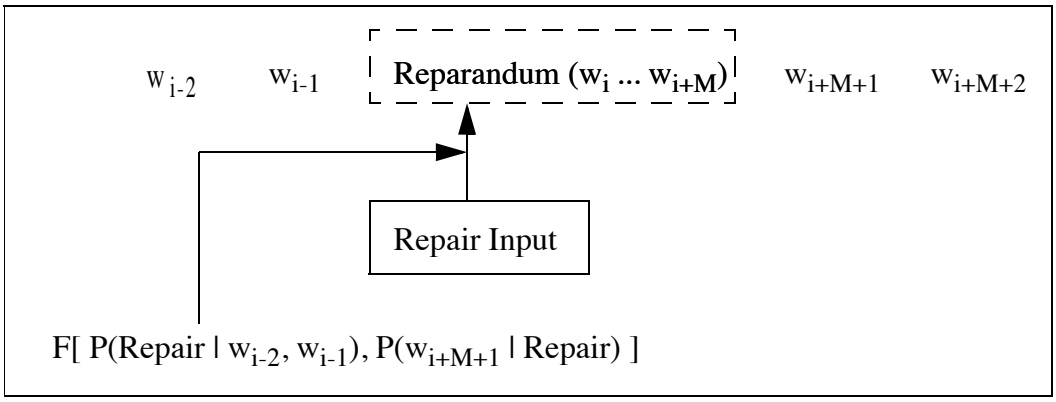


Figure 4-6. *Context modeling for correcting by replacing (and inserting) words*

Given this situation, word context modeling can be implemented for continuous speech repair input as follows. Continuous input recognizers that use a search driven by a statistical language model typically recognize any new input with a "neutral" language model context. Customarily, pseudo symbols for the beginning and end of an utterance are introduced to vocabulary and language model (e.g., "<s>"and "</s>" in the widely used standard N-gram language model file format defined by NIST). A new utterance $\mathbf{v_1...v_K}$ is typically recognized as <s> $\mathbf{v_1...v_K}$</s>. [1] With N-gram context modeling, the neutral language model context is

replaced with the appropriate pre- and post-context from the current error region. For example, when a trigram language model is employed, the new utterance is recognized as w_{i-2} w_{i-1} $\mathbf{v_1 \cdots v_K}$ w_{i+m+1} w_{i+m+2}. Thus, at the beginning of recognizing correction input, language model scores $P(v_1|w_{i-2},w_{i-1})$ are used instead of the neutral $P(v_1|<s>,<s>)$. In Figure 4-6, the context modeling function F represents this change in the computation of language model scores.

Word context modeling can be applied to isolated-word recognizers by using a rescoring algorithm - assuming that the recognizer is able to provide the K best recognition hypotheses. First, the recognizer interprets isolated-word correction as an independent event in the usual way, and provides a K-best list of alternative hypotheses as recognition output. The K-best list of hypotheses $\{c^1,...,c^K\}$[1] is then rescored using context modeling scores. Assuming a trigram language model, and the pre- and post-context as defined earlier, the context score CS(k) for the k-th alternative hypothesis is defined as:

$$CS(k) = P(c^k \mid w_{i-2}w_{i-1})P(w_{i+M+1} \mid w_{i-1}c^k)P(w_{i+M+2} \mid c^k w_{i+M+1})$$

Using such context scores, the context modeling function F in Figure 4-6 can be implemented by interpolating context scores CS(k) with the recognition score for the k-th alternative hypothesis. In terms of the general flowchart of multimodal interactive correction in Figure 4-1, context modeling is part of the correlation step.

In the multimodal dictation system prototype, N-gram context modeling was implemented as follows. For the continuous speech modality, context modeling was integrated in the standard speech recognition algorithm, a time-synchronous search driving by a language model. For the spelling and handwriting modalities, context modeling was integrated as a rescoring pass

1. The pseudo words <s> and </s> do not correspond to any acoustic event.
1. To clarify the notation: alternatives for the same input are indexed with a superscript, for distinction from indexes for sequences of input (as subscript).

after isolated-word recognition.

N-gram context modeling was evaluated using data from the multimodal correction database. Table 7 shows the performance of N-gram context modeling for correction by repeating in continuous speech, spelling, and handwriting. In addition, using only the pre-context is compared with using both pre- and post-context. The first row shows the baseline accuracy without context modeling, the second row shows the accuracy if only pre-context is used, and the third row shows the accuracy if both pre- and post-context are used. Context modeling significantly decreases the correction failure rate by a relative 18-26% ($F(2,4)=85.9$, $p<0.01$). It may be surprising that using more context (post-context in addition to pre-context) does not consistently improve accuracy. Since users apparently do not consistently select maximal contiguous regions of errors, the post context is frequently not correct, and using an incorrect post-context in context modeling deteriorates accuracy. In summary, context modeling is a very effective algorithm to increase the accuracy of interactive correction, both for cross-modal corrections and for corrections in the same modality (e.g., correction by respeaking).

Table 7: *Increase of correction accuracy by N-gram word context modeling*

Experiment Condition	Continuous Speech	Spelling	Handwriting
baseline (no context modeling)	43%	73%	67%
pre context	53%	80%	75%
pre and post context	52%	81%	74%

4.4.3 Bias Towards Frequently Misrecognized Words

As a second method to correlate correction input with repair context, this dissertation proposes to bias recognition of corrections towards frequently misrecognized words. This algorithm exploits the fact that errors are not randomly distributed; within one input modality, certain words are more frequently misrecognized than others. In first-order approximation, the recognizer's error behavior can be modeled as unigram distribution $P(incorrect|w, m)$ that

indicates how likely a word *w* is recognized incorrectly in modality *m*. How can this unigram distribution be used in recognizing correction input?

Similar to the context modeling algorithm described in the previous subsection, we must make a distinction between recognizers whose search is driven by a language model, and isolated-word recognizers which do not employ a language model. If the recognizer employs a language model, we can modify the language model to compute the joint probability that a word is correct, as opposed to merely computing the probability for a certain word. For example, if a unigram language model is employed, we modify the language model to compute $P(w, correct)=P(w)P(correct|w)^{\mu}$ as the product of the regular unigram distribution $P(w)$ and the weighted unigram bias that *w* is correct $P(correct|w)^{\mu}$. A weight parameter μ determines how the regular language model and bias should be balanced. The value for μ an be determined empirically by maximizing correction accuracy on a cross-validation set of correction data. Figure 4-7 illustrates how the usual score for a word *w* given an input signal *A* is computed out of the signal model score and language model score, and how the bias is introduced as an additional factor.

$$Score(w, A) = \underbrace{\log P(w|A)}_{\text{Signal Model}} \underbrace{P(w)}_{\text{Language Model}} \underbrace{P(correct|w)^{\mu}}_{\text{Bias}}$$

Figure 4-7. *Extending word scores by a bias towards frequent errors*

If the recognizer does not utilize a language model, recognizing correction input can be recognized with a bias by applying a rescoring technique; for each alternative hypothesis c^k (obtained from recognizing the correction input as an independent event), a bias score $B(k)$ is computed as $B(k)=\log P(correct|c^k)$. Interpolating the bias score for each alternative in the K-best list with the recognition score results in a new K-best list of hypotheses.

The technique of biasing recognition of correction input towards frequently misrecognized

words was evaluated using data from the multimodal correction database. Table 8 compares the correction accuracy when the bias is used with correction accuracy when the bias is not used, across different modalities. In all cases, the bias was applied in addition to pre-context modeling. For the continuous speech and spelling modality, the bias was integrated with a language model; for the handwriting modality, the bias was implemented as additional rescoring pass. As can be seen, correction failure rates decrease 8% and 20% for the handwriting and spelling modalities, respectively. Unfortunately, no improvement was achieved for continuous speech corrections. Differences in the effectiveness across modalities could be due to differences in how the bias was integrated in the recognition, as described above.

Table 8: *Increase of correction accuracy by biasing towards frequently misrecognized words*

Experiment Condition	Continuous Speech	Spelling	Handwriting
without bias	52%	80%	75%
with bias	52%	84%	78%

In summary, biasing corrections towards frequently misrecognized words can help to further increase correction accuracy.

4.4.4 Correlating N-best Lists

So far, very little information obtained from recognizing previous user input for the error region was used in recognizing current correction input. This section presents an algorithm that uses the information contained in the N-best lists of alternatives for primary and correction input. Ideally (i.e., if the N-best lists can be sufficiently large), each N-best list should contain the true hypothesis, albeit possibly far down in the list. Since the sets of words that are confusable differ across modalities, correlating the N-best lists in cross-modal corrections should identify the true hypothesis almost immediately; only the true hypothesis is likely to occur in both N-best lists.

However, the following practical constraint impedes this method of correlating correction input with repair context. The recognizer is tuned to achieve real-time performance in interac-

tive applications, commonly by pruning more aggressively in the search for the best matching hypothesis. Under such conditions, the true hypothesis is frequently *not* among the N-best list. For example, with the real-time JANUS large vocabulary continuous speech recognizer used in this thesis, the maximal word accuracy is 93% for primary input, 79% for multiple word corrections by respeaking, and only 36% for isolated-word corrections by respeaking. Under these circumstances, not much improvement through correlating N-best lists can be expected.

Correlating N-best lists was implemented in the following way. For any correction, the N-best list obtained for the correction input is merged with the current N-best lists for the error region (obtained either from the primary input or from previous corrections). Alternatives that occur in both lists are placed first, followed by the alternatives that occur only in the N-best lists for the correction input.

4.4.5 Vocabulary Reduction for Partial-word Corrections

Earlier in this chapter (see Section 4.3.3), partial-word correction was proposed as an additional correction method, attractive mainly for data-entry and dictation applications. Correlating correction input with repair context can be applied to partial-word correction, resulting in a significant improvement in partial-word correction accuracy as follows.

Vocabulary reduction for partial-word corrections exploits constraints on the level of words. The recognition vocabulary for partial-word corrections (insertion or substitution repairs) is limited to all words that complete the word fragment to a word within the word-level vocabulary. The following example for large vocabulary dictation illustrates the idea. Assuming that the user dictated "*The market seems to continue very favorably for Blue Chip stocks*", and the system recognized "*The market seemed to continue very favorably for Blue Chip stocks*". The word "*seems*" was misrecognized as "*seemed*". The user decides to try a partial-word correction and selects the final "*s*" in "*seems*". By applying the vocabulary reduction algorithm described above, only the words "seem", "seemed", "seeming", "seemingly", and "seems" match the word fragment "seem" within a standard 20,000 word vocabulary. Knowing that

"seems" cannot be the correct alternative, and applying the vocabulary reduction algorithm, the recognition of the following partial-word correction input can therefore be limited to the character sequences "ed", "ing", "ingly", instead of arbitrary character sequences. Figure 4-8 below illustrates the situation. Although we described corrections by replacing the end of the word, the technique applies to corrections at the beginning or in the middle of a word.

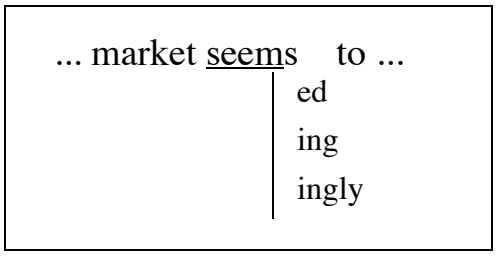


Figure 4-8. *Vocabulary reduction in partial-word correction of "seem"*

Vocabulary reduction for partial-word correction was evaluated on the subsets of corrections by handwriting and spelling from our database of multimodal corrections. Table 9 compares the correction accuracy for correction on the level of whole words with partial-word corrections. Partial-word corrections are significantly more accurate ($p<0.05$).

Table 9: *Increase of partial-word correction accuracy by vocabulary reduction*

Experiment Condition	Spelling	Handwriting
whole word corrections	84%	76%
partial word corrections	97%	81%

In summary, vocabulary reduction makes partial-word corrections superior in accuracy to word-level corrections. The result is a significant increase in correction accuracy. Other algorithms that correlate correction input with repair context, such as N-gram context modeling and biasing towards frequent recognition errors, can be applied to partial-word corrections in addition to vocabulary reduction. Yet this observation alone does not ensure that partial-word corrections actually expedite the correction process, as we will see in the evaluation of partial-word correction in a user study, as presented in Chapter 8.

Figure 4-9 below illustrates the improvement of correction accuracy obtained by various algorithms that correlate correction input with the context of repair, which were presented in this section.

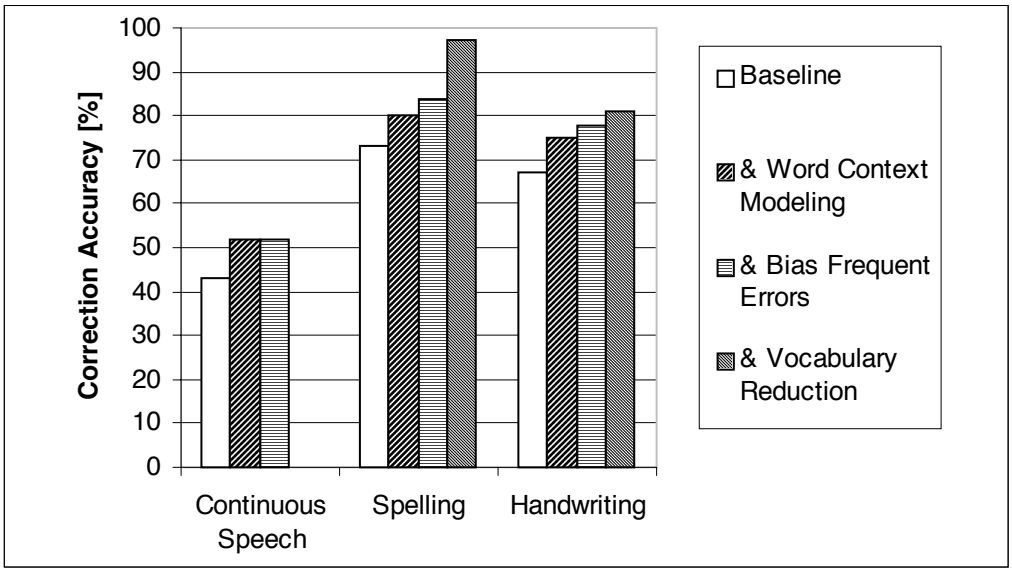


Figure 4-9. *Improvement of correction accuracy by different algorithms that correlate correction input with the repair context*

4.5 Towards a Self-Improving System

For any application with limited performance, it is useful to have methods that can improve application performance "on the job". This applies in particular to (multimodal) speech recognition applications, because the main factor limiting system performance is the recognition accuracy of the multimodal components. Accuracy of automatic recognition is bounded by two factors: the recognition algorithms, and the quality of the (statistical) models that are used during automatic recognition.

The quality of a recognizer's models can be improved "on the job". This section outlines two methods that were examined as part of this thesis work: adapting the acoustic models of the continuous speech recognizer, and dynamically adding new words to the vocabulary. First, the acoustic models are one of the major knowledge sources used during automatic speech recog-

nition. Adapting the acoustic models - to the current acoustic environment, to the current user, or to the task- improves recognition accuracy considerably. The first subsection briefly discusses integration of acoustic model adaptation with interactive error correction. Second, the fact that automatic recognizers typically recognize only words within a given vocabulary is one of the major limitations of current recognition technology. Interactive error correction - as presented in this chapter - did not address this problem. However, provided the user realizes that a repeated recognition error is probably due to the presence of a new word, they can be eliminated "on the job" by dynamically adding new words to the system's vocabulary. The second subsection outlines an algorithm to dynamically add new words, ensuring vocabulary consistency when several recognizers are employed.

4.5.1 Integrating Acoustic Model Adaptation

This section briefly outlines different methods to adapt the acoustic models of a speech recognizer. In addition, benefits from integrating *acoustic model adaptation* with interactive error correction are described.

Within the field of automatic speech recognition, it is widely known that speaker *dependent* continuous recognition is more accurate than speaker *independent* recognition. For applications where one user interacts with the system over longer periods of time, continuous speech recognition performance can be improved by adapting the acoustic models to the current user. This idea is exploited in all current commercial dictation systems. Since the performance improvement is significant, most commercial systems make acoustic model adaptation mandatory for every new user, in the form of an "enrollment session". Obviously, adapting the acoustic models to the current user is not an option for walk-up-and-use applications.

Other methods to adapt the acoustic models include: adaptation to the acoustic environment, adaptation to the task, and improving acoustic models on consistently misrecognized words. First, adaptation to the acoustic environment is frequently realized the same way as adaptation to the user. In fact, the way that adaptation to the user is commonly realized (by collecting a

number of speech samples from the current user in an "enrollment" session) automatically adapts the acoustic models to the environment, in addition to adapting to the user. This can lead to problems when an application is used in various kinds of acoustic environments. For example, performance of an automatic dictation system can deteriorate dramatically when it is moved to a noisier room. Second, adaptation to the task is typically employed during application development. The speech recognition community shares an infrastructure of commonly used standard speech databases that are indispensable for the development of any recognizer. To achieve high performance on any specific application, it is necessary to adapt generic acoustic models, which are trained on such a standard speech database, to the application's task. Generic acoustic models can be adapted to a specific task by collecting a sufficient amount of speech samples on the specific task, and retraining the acoustic models with that additional data. Third, the acoustic models of a continuous speech recognizer can be improved on words that are frequently misrecognized by adapting the models specifically on such words. Such methods are integrated in commercially available dictation systems (e.g., IBM's Via Voice).

Integrating acoustic model adaptation with interactive error correction is beneficial for the following reason: the performance improvement of acoustic model adaptation is higher when the learning is supervised, i.e. if the reference text is available. During adaptation of a recognizer to the current user in an enrollment session, as described in the first paragraph of this section, the reference is known since the user is prompted to read certain sentences out loud. During regular use of a speech recognition application, the reference is usually *not* known. However, after interactive error correction, the reference text for the speech input is available. This knowledge can be used to improve recognition accuracy incrementally through *supervised* acoustic model adaptation.

4.5.2 Adding New Words to the Vocabulary

One known limitation of current recognition technology is the fact that only words within a given vocabulary can be recognized. This limitation is not intuitive to most users, as informal observations confirm: "Users of a speech recognition system are surprised when the system knows one form of a word but not another" (p. 207 in [Rhyne and Wolf 1993]).

One solution to the problem is to offer interactive methods to dynamically add new words to an application's vocabulary. Current commercial dictation systems offer to add new words to the continuous speech recognizer's vocabulary by typing them in. For a multimodal speech recognition application, the situation is more complicated; to ensure consistency across multiple modalities, new words must be added to all multimodal components (that support word input). Consistency across multiple components can be ensured using the "observer" design pattern, known from object-oriented software engineering [Gamma 1995]. The following paragraph briefly describes this algorithm.

An object-oriented model of a multimodal speech recognition application that integrates interactive error correction includes (among others) the following objects: subsystems for automatic recognition of different input modalities, and objects modeling the repair context (i.e., all information required for the algorithms that correlate correction input with the repair context, as described in Section 4.4). For multimodal interactive correction as presented in this dissertation, the relevant subsystems include: audio input subsystem (continuous speech, spelled sequences of letters), pen input (handwriting, pen-drawn gestures), and repair context[1].

Within this object-oriented model of multimodal applications, adding words dynamically to the vocabularies of all recognizers can be realized as follows. The "repair context" model offers a method to dynamically add a specific word, given as its orthographic spelling. This

1. The system architecture for the multimodal dictation system developed in this thesis is described in Section 5.3 in more detail.

method may be triggered by a user request, e.g. the user pressing a certain button, or issuing an "add word" command from some menu. The application's input subsystems are subscribed to the repair context as "observers", and the input subsystems therefore receive an "update" call for each "add word" call. The update call includes the orthographic transcription of the new word as argument. This update call is forwarded to any recognizer subscribed to an input subsystem. Each recognizer implements a specific "add word" method that performs the necessary steps to update the recognizer's vocabulary. Thus, the new word is added to the vocabularies of all recognizers within a multimodal application, ensuring vocabulary consistency. Figure 4-10 illustrates the general flow of method calls while dynamically adding the word "Suhm" to the application's vocabulary. Although the gesture recognizer receives a call to update its vocabulary, it will obviously ignore such a call, since a gesture recognizer does not support word level input.

Before being able to interactively add new words, the words must be detected within user input and distinguished from recognition errors that are due to other causes. Therefore, interactively adding new words to an application's vocabulary solves only half of the "new word problem". Algorithms to automatically detect new words in user input are not yet sufficiently reliable (cf. the literature review in Chapter 2). A general solution to this problem is left as a challenge for future research.

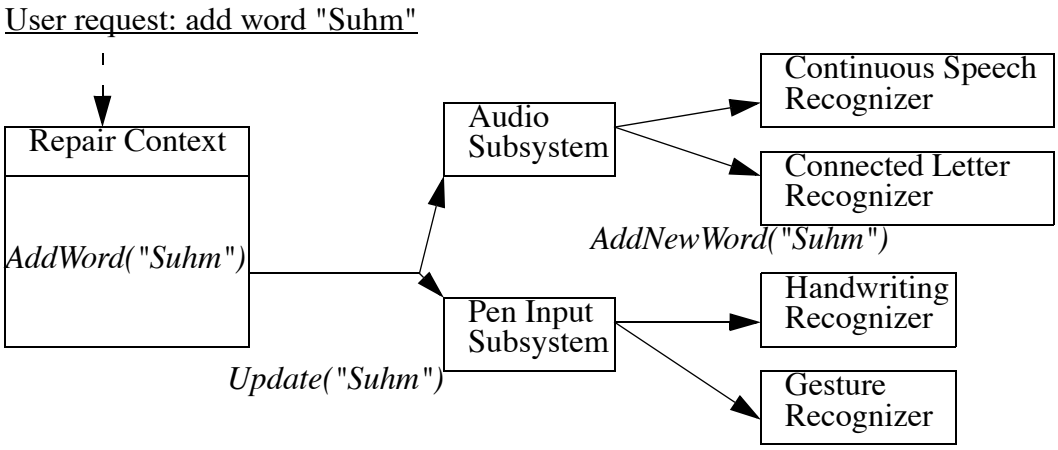


Figure 4-10. *Algorithm to dynamically add new words within a multimodal application*

In summary, this chapter described the general approach and implementation of multimodal interactive error correction. The set of multimodal correction tools developed in this dissertation includes: repeating input by spelling and handwriting, editing using pen-drawn gestures, and (cross-modal) partial-word corrections. Since recognizing correction input is challenging, modifications to standard recognition algorithms are necessary to recognize multimodal corrections with high accuracy. We presented three algorithms that increase correction accuracy by correlating correction input with repair context: word context modeling (that exploits language model constraints in interpreting correction input), bias towards frequently misrecognized words, and vocabulary reduction for partial-word corrections. Using these algorithms, cross-modal corrections using spelling or handwriting are 80-90% accurate, compared to 50% for unimodal correction using continuous speech.

5. A Multimodal Dictation System Prototype

This chapter describes the prototype *multimodal dictation system* that was developed in this thesis work. The prototype serves two purposes: first, to demonstrate the effectiveness of multimodal interactive error recovery in a potentially useful speech recognition application; and second, to compare different interactive error-correction methods in user studies.

Section 5.1 presents background information on dictation systems: for what kinds of tasks dictation systems are useful, and other text production methods used historically. Section 5.2 describes how multimodal interactive error recovery was implemented in the context of a dictation application to build a *multimodal* dictation system: what correction methods were chosen, how application-specific issues such as triggering of different methods were solved, and what editing gestures are supported. Section 5.3 describes how the processing of multimodal input was implemented. Both the general system architecture and the algorithms for automatic classification of input modalities presented here may be useful for other multimodal applications. Finally, Section 5.4 discusses hardware issues. While audio hardware belongs to the standard features in all PCs and laptops, a good (and affordable) solution for displays on which users can write has yet to be found.

5.1 Dictation Systems

Document generation, or more generally data entry, plays an important role in today's workplaces. The methods used for document generation have evolved over time - from handwriting to book print, (mechanical) typewriters, and word processors. The first commercial automatic dictation systems have recently become available. As a prominent example of a non-conversational speech recognition application with a graphic user interface, dictation was chosen for

this dissertation to demonstrate the effectiveness of multimodal interactive error correction. This section provides background information on dictation systems. The first subsection distinguishes between the two tasks that dictation systems can support: text reproduction and text composition. Then, a brief review of text-production methods follows, from handwriting via print to automatic listening typewriters (also called dictation systems). Previous studies on dictation systems present important background information for evaluation issues. A review of such studies is therefore deferred to the evaluation part of this dissertation, in Chapter 6.

5.1.1 Text Reproduction versus Text Composition

Dictation systems can support different tasks. The task of *text reproduction* or transcription is to recreate some given text in machine-readable format. Typing a handwritten or dictated manuscript are typical examples of text reproduction tasks. *Text composition* focuses on the creative act of generating the text. Examples include dictation of a business letter to a secretary by an executive, composition of electronic mail, writing a paper or a book. Converting the text into a medium suitable for printing is only a secondary concern in such tasks. Extending the type of content being created, future "dictation" systems may support the generation for multimedia documents (e.g., Web authoring tools).

5.1.2 From Handwriting to Listening Typewriters

Until this century, text production methods were limited to handwriting, dictating text to a scribe or secretary, and print (for the dissemination of written material in larger quantities). How did text production methods evolve during the past century? The invention of the typewriter revolutionized text production. Typewriters made it possible to generate high-quality documents efficiently. Since the 1950's, tape recorders (used as dictation machines) made it possible to separate the process of dictating text from the process of producing a written form of the text. Not long after the invention of computers, engineers dreamed about replacing the standard keyboard with automatically recognizing speech, in a so-called listening typewriter (or dictation system). Gestures and handwriting as input to computers have been considered

since the 1960's; more recently, with progresses in miniaturization of computer technology, the development of portable devices has led to the creation of a new subfield of computer applications called "pen-computing". Both continuous speech dictation systems (listening typewriters) and small hand-held devices that support handwriting recognition have become available commercially since the 1990's.

The following quotations from well-known researchers are intended to give the reader an impression of the fascination that a listening typewriter has exercised on the field. A listening typewriter has been termed the "holy grail of ASR" (Automatic Speech Recognition technology) [Baber and Hone 1993]. One of the most enthusiastic pledges for the usefulness of such an application can be found in [Gould, Conti et al. 1983]. We ask the reader's pardon for reproducing this quotation in full. "A listening typewriter is a potentially valuable aid in composing letters, memos, and documents. Indeed, it might be a revolutionary office tool, just as the typewriter, telephone, and computer have been. With a listening typewriter, an author could dictate a letter, memo, or report. What he or she says would be automatically recognized and displayed in front of him or her. A listening typewriter would combine the best features of dictating (e.g., rapid human output) and the best features of writing (e.g., visual record, easy editing). No human typist would be required, and no delay would occur between the time an author creates a letter and when he or she gets it back in typed form. This might lead to faster and better initial composition by the author, psychological closure because of no wait for (and uncertainty about) a typed copy, quicker and better communication, and displaceable typing and organizational costs." The near future will reveal whether automatic listening typewriters are ready to live up to these claims.

5.2 Multimodal Interactive Error Recovery for Dictation Applications

A *multimodal dictation system* (or *multimodal text editor*) is an automatic listening typewriter that offers effective keyboard-free editing and error correction using multimodal interactive correction methods. This section describes how multimodal interactive error recovery (as

introduced in general terms in the previous chapter) was integrated with a state-of-the-art large vocabulary dictation system to build a multimodal dictation system prototype.

5.2.1 Locating Recognition Errors

For locating recognition errors, two approaches were implemented and evaluated in the multimodal dictation system prototype: one user-initiated and one system-initiated method. For user-initiated detection and location of recognition errors, the recognition hypothesis is presented visually on the screen. The user reviews the recognition result and selects recognition errors by tapping on words. For system-initiated detection and location of recognition errors, automatic highlighting of errors based on confidence measures was implemented and evaluated, as described in more detail in the following paragraphs.

Confidence measures can be used to detect and locate recognition errors by applying a threshold criterion on the confidence scores; if the confidence score for a word exceeds the threshold the word is tagged as correct, otherwise it is tagged as a recognition error. This application of confidence measures to the problem of error identification was proposed previously in [Chase 1997]. Confidence measures themselves are not reliable, therefore the tags assigned to the words may be incorrect. More specifically, misrecognized words may mistakenly be tagged as "correct" (i.e., missed detections of recognition errors), and actually correctly recognized words may be tagged as recognition error (i.e., false alarms). Hence, an automatic method to highlight errors based on imperfect confidence measure must balance missed detections and false alarms. The following paragraph describes how this was achieved for the automatic highlighting of errors in the multimodal dictation system.

Assuming both missed detections and false alarms are equally harmful[1], the threshold can be tuned to minimize the number of classification errors. Figure 5-1 shows the number of classification errors (the sum of missed detections and false alarms) across different thresholds. The confidence tagger is based on the gamma feature.[2] A detailed description of gamma can be

1. Future research is needed to determine whether this assumption is valid.

found in Kemp's paper [Kemp and Schaaf 1997]. To apply gamma in the multimodal dictation system, the best error classification performance of 89% is achieved with a threshold 0.6.

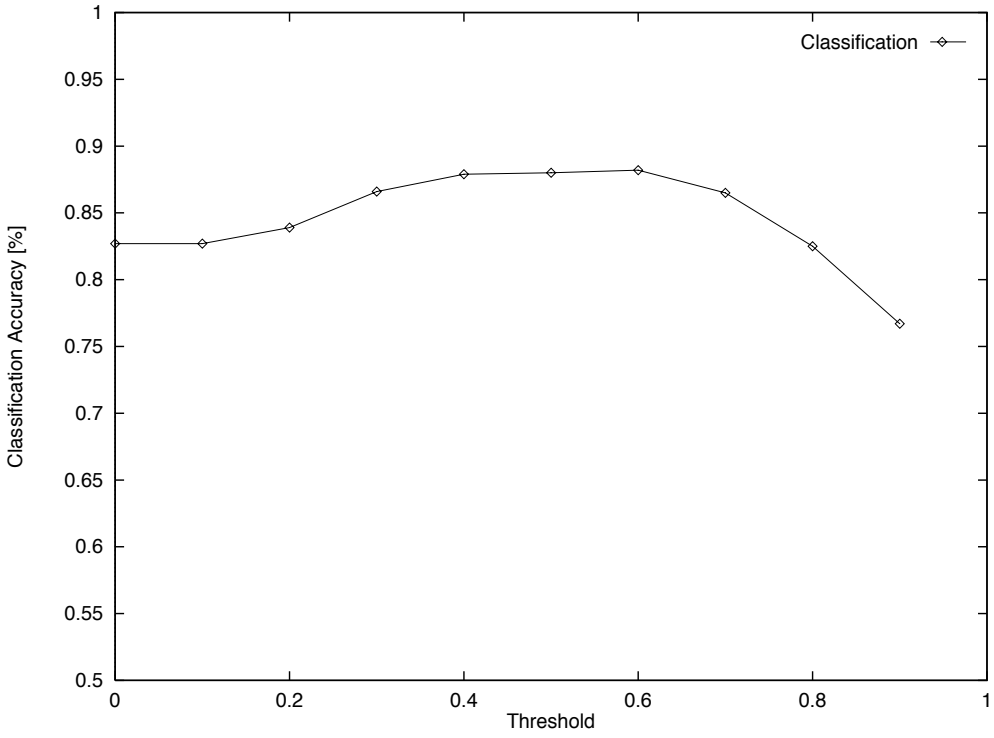


Figure 5-1. *Optimizing threshold for system-initiated location of recognition errors based on confidence measures*

The method of automatically locating recognition errors described above was evaluated in user studies. To determine the impact of classification errors, the imperfect methods are compared to a perfect automatic error locating algorithm using a cheating experiment (i.e., automatically locating errors based on knowledge of the true input). For more information, refer to the evaluation in Chapter 8.

5.2.2 Interactive Correction

The multimodal dictation system prototype provides the following methods for interactive correction of recognition errors: unimodal correction by respeaking, cross-modal correction

2. Gamma (see Section 2.1.2) was evaluated on the set of initial dictated sentences from the database of multimodal corrections (cf. the introduction of Section 4.4), and using the JANUS WSJ large vocabulary recognizer (see Section 3.1.1). The recognition word accuracy on this test set was about 83%.

by repeating input using spelling or handwriting, cross-modal partial-word corrections using spelling or handwriting, choosing from a list of alternatives, and editing using pen-drawn gestures. The correction algorithms were described in Section 5.2 in the previous chapter and needs no further description. This section focuses on specific design problems that were encountered during implementation of these correction algorithms for the multimodal dictation system prototype.

The most difficult design problems can be summarized under the issue of how to trigger and distinguish different correction modalities. Table 10 enumerates these design problems, the different designs that were tried, and the usability problems of each design. Faced with these design problems, the following decisions were made for the multimodal dictation system prototype (in the order presented in the table). First, to trigger the list of alternatives, the pull-down gesture proved to be too unreliable to recognize automatically, and too confusing to the user. Instead, touching a word for an extended moment (approximately one second) appeared to work best in our informal user tests. Second, to distinguish between the two speech modalities (continuous speech and spelling), separate buttons were introduced (one for continuous speech, and one for spelling). Automatic classification of speech input in continuous speech and spelling is the desired solution, but too unreliable with current technology on large vocabulary tasks[1]. Finally, the end of the two types of pen input (handwriting and editing gestures) was determined using a time-out of approximately one second of no pen input.

1. Algorithms to automatically distinguish continuous speech from spelled letter sequences were developed and evaluated in the context of locating spelling sequences, which are embedded in spontaneous dialogues [Hild and Waibel 1995]. However, the vocabulary size was much smaller (only 3,000 words); therefore, this task is significantly easier than 20,000 words necessary for large vocabulary dictation. For more information, see Section 5.3.2 later in this chapter.

Table 10: *Selected design and usability problems of multimodal interactive correction*

Design Problem	Design	Usability Problems
Trigger list of alternatives	Button	Clutters interface
	Double tap on word	Confusable with single tap on word (used to select word)
	Touch word "long"	Either slows interaction down or is confusable with single tap on word
	"Pull-down" gesture	Gesture recognizer confuses pull-down with other editing gestures
Distinguish continuous speech from spelling	Separate button for each modality	Clutters interface
	Classify automatically	Classification imperfect and leads to additional errors
	One button for continuous speech, tap selection to trigger spelling	Mode errors
How to trigger inserting by spelling (applies only when tap on selection triggers correction by spelling)	"Long" tap between two words	Confusable with selecting either word (or gap between words has to be overly large)
	Tap cursor	May work well
Determine end of handwriting/gesture input	Time-out criterion	Time-out has to be adapted to user, and slows interaction down
	Button to launch recognition on input	Users forget to press recognition button

Thus far, methods to replace and insert words have been described. The following section presents the use of pen-drawn gestures to support simple editing tasks (deleting and positioning the cursor).

5.2.3 Editing using Gestures

Simple editing tasks (such as deleting, selecting, positioning the cursor, moving items, and formatting) can be performed efficiently using pen-drawn gestures (drawn on a writing-sensitive display), as described in Section 4.3.2 in the previous chapter. The following paragraphs describe the editing tasks that are supported in the multimodal dictation system prototype, and

the set of gestures (editing marks) chosen for each of these editing tasks.

The multimodal dictation system supports the following three editing tasks, which are the only indispensable ones for correction of recognition errors in dictation applications: selecting items, deleting items, and positioning the cursor. All of these operations can occur either on the level of words (selecting or deleting one or more words, positioning the cursor between words), or on the level of characters within a word (selecting or deleting one or more characters within a word, positioning the cursor within a word).

There are two approaches to distinguish between word-level and character-level as the scope for editing gestures: defining separate gestures for each scope, and determining the scope from the interaction context. In the multimodal dictation system, the scope of editing gestures is determined from the context. For insertions, if the mark is made between characters within a word, the cursor is positioned between those characters (and the system switches into partial-word correction mode), whereas if the mark is made between two words, the cursor is placed between those words (and the system switches into whole-word correction mode). For deletions, if the mark covers only some characters within a word, those characters are deleted; if the mark covers the whole word (or several words), it is interpreted as a word-level deletion. For selecting, a separate gesture had to be introduced because characters are small and difficult to select. Words are selected by tapping them; characters are selected by demarcating them with two vertical bars. Figure 5-2 shows a screen shot of the multimodal dictation system prototype, and Figure 5-3 illustrates the various types of gestures supported by the multimodal dictation system, across different scopes (character, word, word phrase).

Even though several designs were tried, not all usability problems with distinguishing the scope of editing commands could be eliminated. Positioning the cursor on the character-level is difficult at the beginning or at the end of a word, deleting characters within a word is ambiguous with deleting the whole word, and demarcating characters within a word remains difficult. These usability problems were alleviated in our prototype by using large fonts to display

recognition results. Further improvements of the hardware may eventually eliminate these usability problems (cf. Section 5.4.3).

Figure 5-2. *Snapshot of multimodal dictation system prototype*

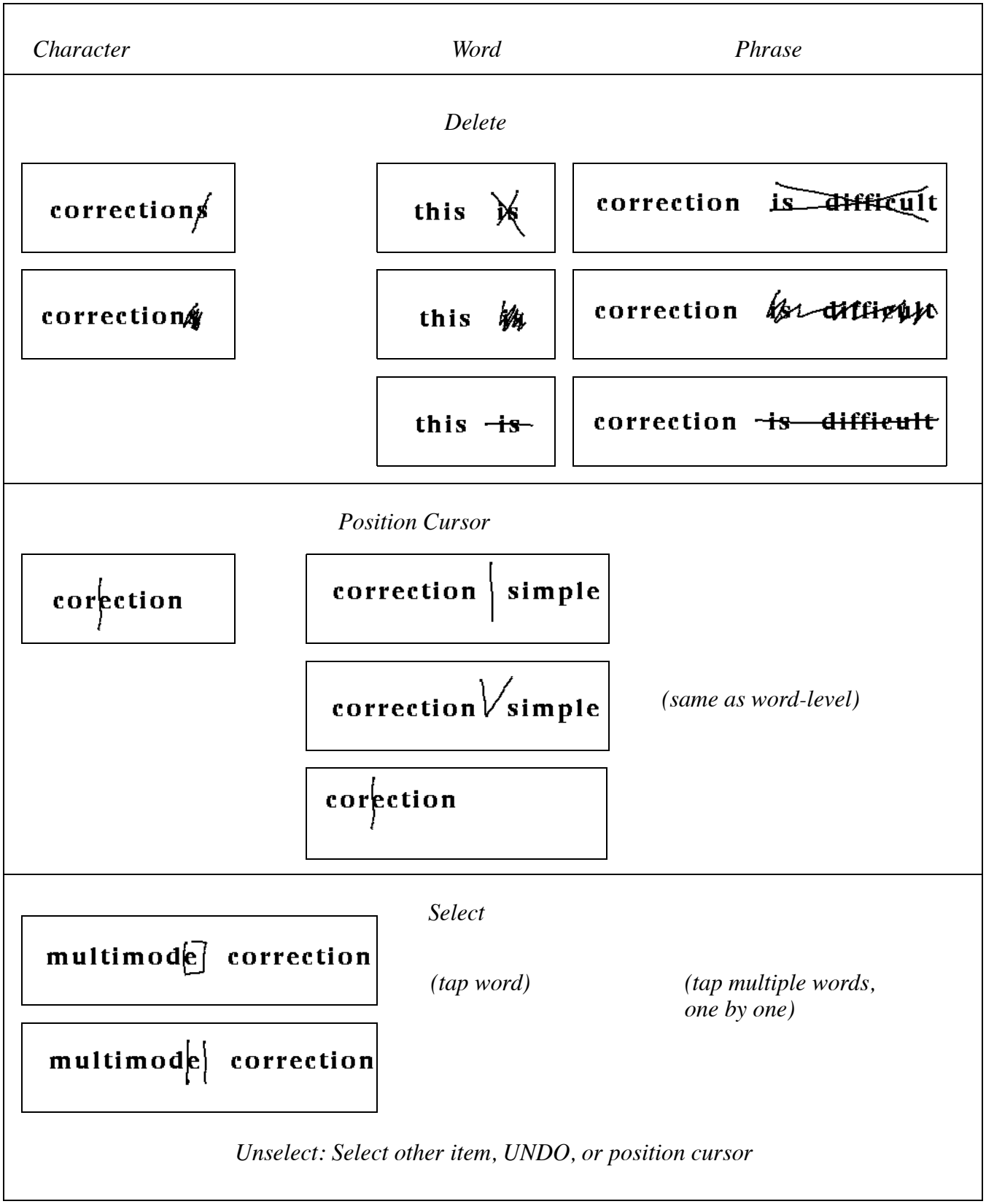


Figure 5-3. *Editing gestures supported in the multimodal dictation system*

5.3 Processing Multimodal Input

Implementation of multimodal applications is challenging from a software engineering point of view. First, the system architecture must support a highly distributed system. Since automatic recognition requires high computational resources, each recognizer should be hosted in a separate process, to distribute the computational burden among different computers if available. Second, input streams in different modalities must be diverted to the appropriate recognizers, and the recognition result of the matching recognizer must be retrieved. To divert input streams to the appropriate recognizers, effective and reliable methods to classify different input modalities are required.

This section describes how these issues were addressed in the implementation of the multimodal dictation system prototype. The first subsection presents the system architecture. Using a client-server architecture and object-oriented software design, a front-end that accepts input and presents feedback, is separated from a back-end that performs all necessary processing. Thus, the multimodal dictation system front-end can run in different environments (e.g., under standard X, or in a web browser), while the computationally intensive back-end processing can be hosted on powerful servers. The second subsection describes automatic classification methods for audio input (continuous speech and spelling) and pen input (handwriting and editing gestures).

5.3.1 System Architecture of the Multimodal Dictation System

The multimodal dictation system was implemented using a client-server architecture that could be useful for other multimodal applications. This architecture addresses two important issues: first, it separates input capture and system feedback from all processing; second, it implements the processing of multimodal input as a server that delegates input to the appropriate recognition subsystems. Both ideas are well-known within the software-engineering and user-interface communities. The first idea makes it possible to run the application's front-end (the part visible to the user) in heterogeneous computing environments, for example, in

both an X windows environment and a web browser. The second idea distributes the heavy computational burden on the application's back-end (needed for the automatic recognition of multiple modalities) among several powerful server hosts. What follows is a review of the "observer" pattern, an important design pattern of object-oriented programming that cleanly separates visualization and feedback from the main processing. This design separates the interface (visualization and feedback) from the back-end (recognition and interpretation) in the multimodal dictation system prototype. The section concludes with a description of the system architecture of the multimodal dictation system.

Strict separation of input/output from representation and processing was realized using the "observer" design pattern of object-oriented programming [Gamma 1995]. The observer design pattern is a generalization of the MVC (Model-View-Control) object-oriented design. The main idea of the observer design pattern is to encapsulate the data-structures underlying the application and all major information processing in so-called "model" objects, and to encapsulate the control for input/output and visual representation in so-called "observer" objects. The design pattern ensures all necessary communication between model and its observers; any change in the model is forwarded to all observers using a generic update call. The observers know how to represent data visually, how to accept user input, and how to pass it on to the back-end. Several observers of the same model can realize completely different visualizations and feedback schema. Thus, input capture and feedback can be easily adapted to different application requirements.

What follows is an explanation of how this design pattern can be applied to separating interface from back-end processing in multimodal applications. The back-end implements the processing of input streams and interpretation of the recognition results in different "model" objects. Hereby, it is useful to further encapsulate recognition subsystems, data structures associated with methods that implement the application's functionality, and an integration module that implements all major policies. Such policies determine how to relay input coming from the front-end to the recognition subsystems, how to retrieve recognition results from

them, and how to initiate appropriate actions. The front-end implements the actual user interface of the application as an "observer" of the back-end. The front-end's functionality is limited to capturing user input, communicating input to the back-end, and generating appropriate feedback for each of the model's "update" calls. Ideally, no input processing or interpretation is performed in the front-end[1]. Figure 5-4 shows how these ideas were realized in the system architecture of the multimodal dictation system prototype. On the front-end ("client") side, user input is captured from two input streams: audio data (via head-set and appropriate audio-hardware) and pen input (via a writing-sensitive display). Furthermore, the front-end interprets "update" calls from the back-end for two main cases: display of the current text input, and feedback on the status of the application (whether it is accepting new input, or currently interpreting input). On the back-end (server) side, there are three main modules: the integration module "repair logic", the "model", and the recognition subsystems for audio and pen input. The "repair logic" implements all major policies pertaining to multimodal interactive error correction: relaying user input, delegating recognition to the appropriate subsystem, and initiating the appropriate error-correction method. The "model" contains representations for the current text and the repair context (i.e., all information necessary to implement the advanced correction methods that correlate correction input with repair context). The recognition subsystems provide a common interface to the diverse recognizers used in this thesis work: for audio input the continuous speech recognition JANUS recognizer and the connected letter recognizer NSpell; and for pen input the on-line handwriting recognizer NPen and a gesture recognizer.

1. Efficiency under real-time constraints can be optimized by shifting of some processing from the back-end to the front-end.

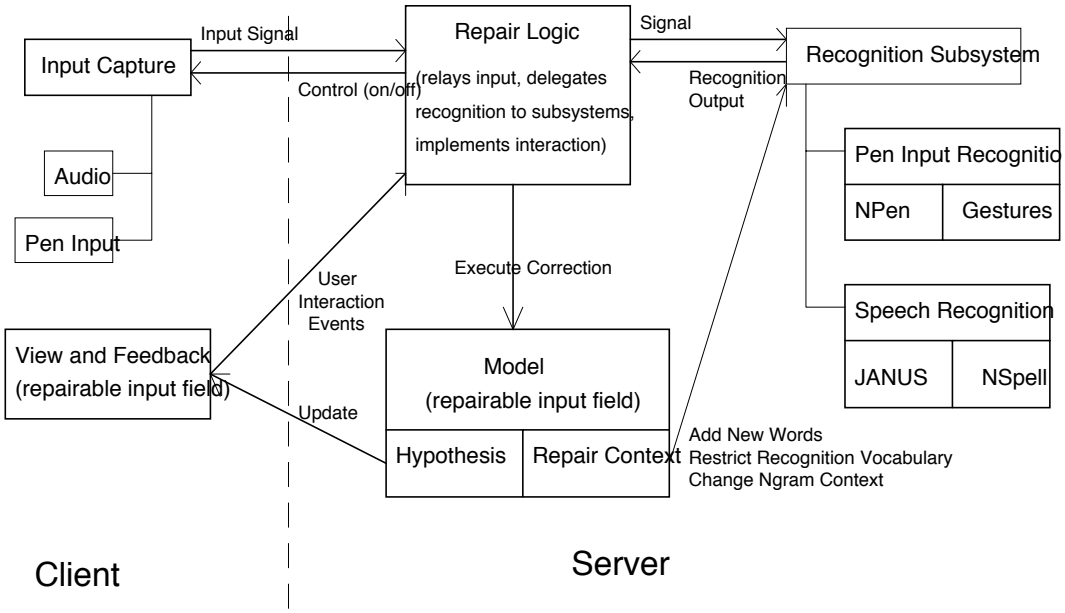


Figure 5-4. *System architecture of the multimodal dictation system*

Figure 5-5 illustrates how to implement interactive error correction in a general manner, applicable to applications other than a multimodal dictation system. The "Correction Algorithms" module encapsulates the error correction functionality as described in the previous chapter, and kept separate from all application specific functionality (summarized as "Application Kernel" module). Both these modules subscribe to automatic recognition services, which are encapsulated in the recognition subsystems. A recent Ph.D. thesis [Vo 1998] describes an object-oriented approach to building a toolkit for rapid prototyping of multimodal applications. Future work could extend this toolkit with multimodal interactive error correction modules.

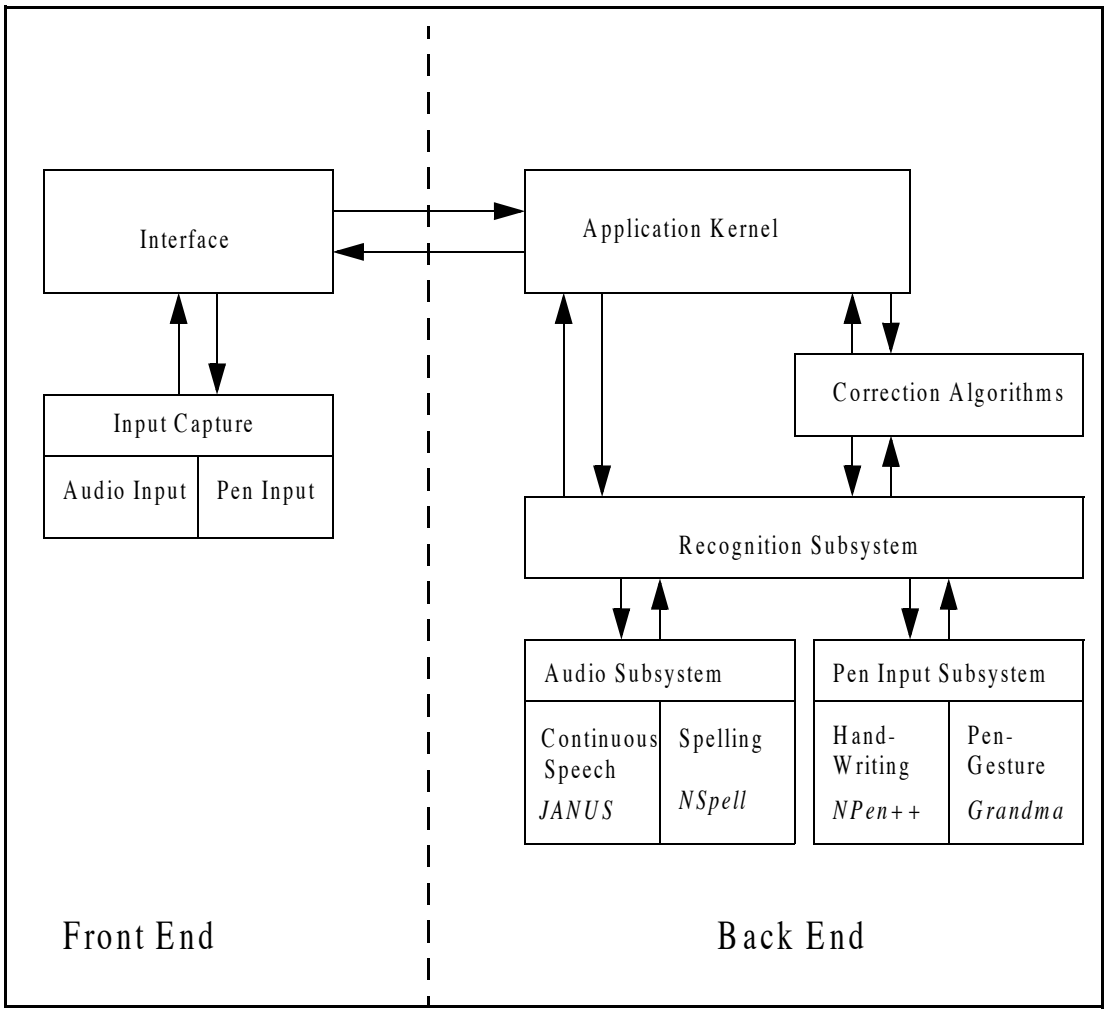


Figure 5-5. *System architecture to integrate multimodal correction in arbitrary applications.*

5.3.2 Classification of Input Modalities

For the processing of different input streams in a multimodal application, the inputs streams must be relayed to specialized recognizers unless recognizers capable of processing all input types are available. If specialized recognizers are used for the same kind of input (e.g., continuous speech and spelling recognizers for audio-input), the application's back-end must split the input stream and delegate recognition to the appropriate recognizer. This section presents methods to split the input stream: using automatic classification algorithms and deriving the appropriate modality from the context of a user interaction. We first present how to split the

audio-input stream into continuous speech and spelling, and then how pen-input can be classified as handwriting and pen-drawn gesture, respectively.

5.3.2.1 Classifying Audio Input in Continuous Speech vs. Spelled Letter Sequences

In this thesis work, two algorithms for automatic classification of audio input into continuous speech and spelled letter sequences were explored. The first approach is a simple application of Hild's method to locate sequences of spelled letters within spontaneous continuous speech utterances [Hild and Waibel 1995]. The second employs standard pattern classification methods. These algorithms are outlined in the following two paragraphs.

Hild's method adapts a standard continuous speech recognizer to recognize both continuous speech and spelled sequences of letters. A speech recognition system can process both continuous speech and spelled letters by modifying its dictionary to contain items for each spelled letter, in addition to regular word-level items (e.g., an item "A - /ey/" for the letter "a", an item "B - /b/ /iy/" for the letter "b", etc.), and by modifying the language model to include sequences of letter items, in addition to regular word sequences. The key point is in which contexts such letter sequences may occur. Hild was only interested in locating spelled names. Since names tend to occur only in very specific word contexts (cf. [Suhm 1993]), the language model can contribute significantly to locating sequences of spelled letters. For the more general problem of distinguishing continuous speech input from sequences of spelled letters, the distinction must be made based on acoustic evidence only. For small vocabularies (5,000 words), classification accuracy of greater than 90% was achieved, but the classification accuracy deteriorated to unacceptable levels on for large vocabulary dictation tasks (20,000 words and more).

The second algorithm applies standard pattern classification methods (a linear classifier) to separating normalized scores. Normalized recognition scores are obtained by interpreting audio input in two ways: with a continuous speech recognizer and a connected letter recognizer. The performance of this algorithm appears to be sufficiently high on small-vocabulary

tasks (e.g., 95% classification accuracy with a 1,000-word Wall Street Journal vocabulary), but the algorithm does not scale to large-vocabulary dictation.

Since the accuracy of automatic classification methods was insufficient, the multimodal dictation system prototype classifies into continuous speech and spelled sequences of letters based on the interaction context. The user interface contains two separate buttons, one to activate continuous speech input, and another to activate spelling input. Although this simple method clutters the user interface with two buttons, it avoids mode errors (users confusing continuous speech and spelling input), and no additional errors are introduced due to incorrect classification of modalities.

5.3.2.2 Cursive Handwriting vs. Pen-drawn Gestures

An automatic classification algorithm was developed to distinguish between two types of pen input: cursive handwriting and pen-drawn gestures. The algorithm is based on the fact that the recognition scores for gesture and handwriting input are sufficiently different. For the template-matching gesture recognizer employed in this thesis work, the Mahalanobis distance proved to be the most useful feature in distinguishing gesture from handwriting; the distance is significantly higher on cursive handwriting input than on gesture input. When aided by additional application-specific heuristics (e.g., the ratio of the pen trajectory over a word helps to identify deletion gestures), automatic classification of gesture and handwriting input is more than 90% accurate. Problematic is handwriting input consisting of only one letter, which consistently led to classification errors in the prototype.

5.4 Hardware

A multimodal interactive dictation system reacts to speech as a "natural" and efficient communication medium, and imitates some favorable properties of paper, including salient visual feedback and convenient storage and information editing. Consequently, there are two challenges for the input hardware of a multimodal dictation system: capture of audio input and capture of pen input. For audio input, headsets in different forms are still the best option for

desktop applications, although they may be uncomfortable to wear. They are necessary since close-speaking microphones are necessary for good speech recognition performance on large vocabulary dictation. For pen input, the feel of editing on paper should be imitated. In the course of developing the prototype, several display and pen-input technologies were tried. A touch-screen provided the best trade-off among currently available hardware. Since audio input is considered standard and needs no further description in this dissertation, the following sections focus on hardware for pen input, in particular, flat-panel and touch-sensitive display technologies.

5.4.1 Flat-Panel Displays

Flat-panel displays can be divided into two categories: *active displays*, which emit light, and *passive displays*, which reflect lighting coming from other sources. Figure 5-6 shows a taxonomy of flat-panel technologies that is based both on a comprehensive (yet somewhat outdated) survey [Tannas 1985], and on a more recent survey [Harding, Martin et al. 1996].

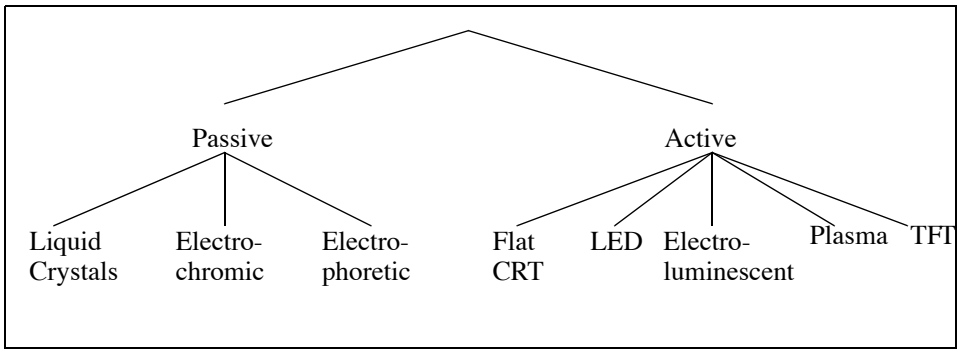


Figure 5-6. *Taxonomy of flat-panel displays*

Table 11 compares the main characteristics of most current flat-panel display technologies (adapted from [Kankaanpaa 1988]). This paragraph discusses pros and cons of each of these technologies. Electrolumniscent panels operate at low voltages, thus decreasing the overall power consumption of a system. This issue is particularly important for small portable devices, which are becoming increasingly popular. The main problem with LCD is low con-

trast and thus poor legibility, whereas the major obstacles of using plasma displays are high cost and unattractive appearance. Recently, TFT displays have been widely used for laptops. Color has become standard, and resolutions comparable to much larger CRT displays (1200x1025) have become available.

Table 11: *Characteristics of important flat-panel displays*

Characteristic	CRT	Plasma	EL	Active Matrix LCD	Passive Matrix LCD
Max. physical Size	25'' diago-nal	40'' diago-nal	17'' diagonal	21'' diago-nal	10'' diago-nal
Max. Resolution	2048 x 2048	1200 x 1600	1024x800	1280x1024	640 x 480
Thickness	>> 2''	0.5''	0.25''	0.25''	0.25''
Weight	heavy	light	very light	light	light
Contrast	good	medium	good	medium	poor
Lifetime	10,000 h	> 40,000 h	30,000 h	> 30,000 h	>30,000 h

5.4.2 Touch-Sensitive Panels

Touch-sensitive panels are still under development. Figure 5-7 classifies the most important technologies for display pointing devices.

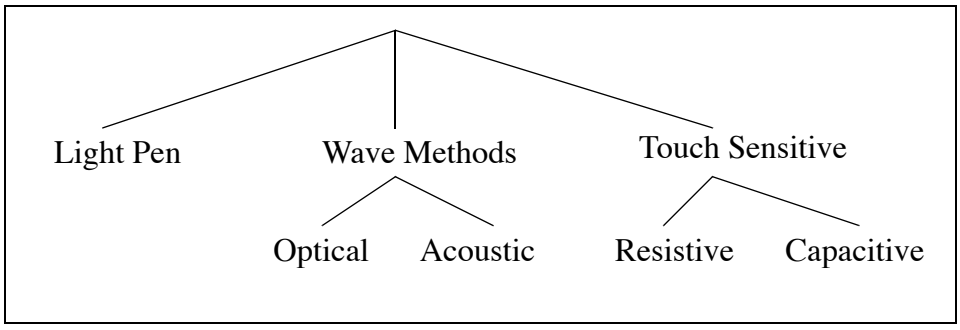


Figure 5-7. *Taxonomy of pen input devices*

This paragraph discusses some advantages and disadvantages of the different pen-input devices (based on [Kankaanpaa 1988], p. 73). With optical-resistive sensors, any stylus can be used. Capacitive methods work with any conductor, including a finger. Most capacitive technologies cannot distinguish between the touch of a stylus and the touch of parts of the hand;

whenever several points touch the display, the average position is returned as sensor value. This leads to severe usability problems since a user cannot rest the wrist while handwriting or gesturing on the display. Light pens and resistive sensors eliminate this problem because they ignore input other than the pen, therefore neither fingers nor wrist can corrupt pen input. However, the data-transfer rate of current light pens is a too small to ensure a smooth trace of the trajectory in handwriting, which is very disturbing for handwriting input[1].

Table 12: *Comparison of touch-sensitive display technologies*

Characteristic	Resistive	Capacitive	Optical	Acoustic	Light Pen
Resolution	4000 x 4000	100 x 100	< 100 x 100	2000 x 2000	1000 x 1000
Optical Clarity	medium	medium	good	good	good
Pointing Device	stylus	any conductor	any	any	stylus
Usability for Handwriting	good if data transfer rate from stylus high enough	good if wrist can be rested on display	(not tried)	(not tried)	bad: delay between movement and display
Calibration Procedure	not necessary	not necessary			can be awkward
Cost	medium	medium	medium	high	medium

5.4.3 Hardware Decisions for the Multimodal Dictation System

Which hardware decisions were made for the multimodal dictation system prototype? There was no alternative to using headsets with close-speaking microphones for audio-input. This section discusses alternatives evaluated for pen-input.

Finding an appropriate pen input device proved to be a challenging hardware problem in the development of the multimodal dictation system prototype. A good pen-input device and writing-sensitive display are crucial to minimize usability problems with handwriting and gesture

1. We tried a product by Interactive Computer Products, Inc.

input. In particular, the writing-sensitive display panel may not have a parallax between the writing-sensitive surface and the actual display surface. A parallax occurs when the display has considerable thickness and the pointing device is not perpendicular to the display.

A number of pen-input devices and writing-sensitive displays were tried in the course of this thesis work. We enumerate them and briefly describe the problem(s) encountered with each:

- Touchscreen mounted on CRT: Wrist cannot be rested on screen while writing. For all user studies reported in this dissertation, this hardware was employed to capture pen input.

- WACOM PL-300 LCD Tablet: Screen size is very small (400x600 pixels), and data-transfer rate of resistive technology is too low to ensure smooth handwriting trajectories.

- National Display Systems Touch-Laptop: Capacitive touchscreen that can ignore touches by the writer's wrist (good). However, no touchscreen driver available for LINUX, and performance problems with sampling of pen input when front-end runs in a web browser on a laptop.

- FUJITSU STYLISTIC 1200: Appears to be a much more adequate (yet very expensive) input device, but was not available for the user studies in this dissertation work.

In summary, this chapter described how the various pieces of interactive multimodal correction were put together and integrated with a state-of-the-art large vocabulary dictation recognizer to build a prototype multimodal dictation system. We presented a method to automatically highlight likely recognition errors in the output hypothesis, the set of supported editing gestures, and algorithms to automatically classify different types of audio and pen input. We described usability engineering problems that were encountered during the iterative design process of building the user interface for the multimodal dictation system, and issues in

the choice of hardware. All these problems are not specific to a multimodal dictation system, and may be useful for the design of other multimodal applications.

Part 2: Evaluation

Baber/Hone, among the first researchers to address the problem of error correction in speech user interfaces, noted that "... it is often difficult to compare the (correction) techniques objectively because their performance is closely related to their implementation. Furthermore, different techniques may be more suited to different applications and domains." (from [Baber and Hone 1993]). A number of user interface evaluation methodologies, including acceptance tests, expert reviews, surveys, usability tests, and field tests are accepted in the field of human-computer interaction [Shneiderman 1997]. For research on novel user interfaces, two approaches have predominated: modeling and user studies.[1] The former focuses on quantitative models of (specific aspects of) human-computer interaction that are based on fundamental properties of human cognitive capabilities. Examples include cognitive models (e.g., GOMS [Card, Moran et al. 1980], SOAR), but also performance models that may be more specific to certain kinds of human-computer interaction.

While usability tests with human participants and real speech recognition systems present a rigorous methodology for evaluating efficiency of error-correction techniques, an evaluation approach based on *user studies* has the following limitations:

- *External validity.* Experimental results depend on the specific speech recognizer used, as well as the task (vocabulary) and the participants (experience and training).

- *Internal validity.* Controlling the occurrence of errors is impossible using real recognizers.

While *model-based evaluation* has the advantages of low cost, abstraction from implementation details, and the possibility to iterate design cycles quickly, its usefulness for the design of

1. [Sweeney, Maguire et al. 1993] proposes a taxonomy of evaluation techniques for interactive systems with three main categories: user-, theory-, and expert-based approach. The user-based approach corresponds to user studies, and the theory-based approach to modeling. The expert-based approach applies mainly to product development, but not to evaluating research issues in novel interaction techniques, as we discuss in this context.

concrete speech user interfaces can be questioned:

- Using *average values for the important model parameters* (timings and recognition accuracies) is problematic because they are known to have a large variation across different users. In particular, if different input modalities are employed, the "outliers" become more relevant, because they are likely to be different across modalities. While some modality may be inferior on the average to another (e.g., handwriting vs. speech for input speed), it may be the only effective option for a subset of users (e.g., handwriting as correction modality for all users with very poor speech recognition performance, such as speakers with foreign accents).

- The assumption of *perfect human performance* is questionable, because the design of a correction method determines whether or not users have problems using it.

However, both methodologies - model-based evaluation and empirical studies - complement each other: lack of external validity of user studies can be compensated with predictions from model based evaluation, and the impact of high variation in recognition accuracy, as well as user errors, can be analyzed based on the rich data from user studies. The evaluation of multimodal interactive error correction presented in this dissertation includes both approaches. Chapter 7 presents a simple performance model of multimodal human-computer interaction, which is applied to multimodal interaction error correction in an automatic dictation system. Chapter 8 presents an extensive empirical evaluation of interaction multimodal error correction. As an introduction, Chapter 6 reviews previous empirical studies relevant to dictation systems.

6. Previous Studies on Dictation

The way that people generate text in written form has evolved over time, from handwriting, to dictating to machines, and using typewriters or word processors for transcription. Automatic dictation systems (or listening typewriters) enable users to dictate text using voice instead of a keyboard, and the text is available immediately in machine-readable format. This section reviews previous studies of text-production methods, including dictation. Knowledge of these studies provides important background information, and makes it possible to compare them to the empirical evaluation of the multimodal dictation system, presented later in Chapter 8.

The two sections of this chapter clarify terminology associated with quantitative and qualitative measures and review results from empirical studies of different text production methods, including handwriting, dictating to a secretary or tape recorder, text editors, simulated and real[1] listening typewriters.

6.1 Terminology: Quantitative and Qualitative Measures for Dictation

Most previous studies on dictation adopted task completion time as the main quantitative measure. Task completion time can be decomposed into various time measurements. Such a decomposition is described, along with a terminology that we adopted for the evaluation of the multimodal dictation system.

Task completion time, in the context of dictation tasks, means the total time required to produce a certain text. A decomposition of the dictation task suggests finer grained time measures. A dictation task can be decomposed into three parts: generation of new text, reviewing

1. Prior to this dissertation, only listening typewriters (dictation systems) that require users to pause briefly between every word have been formally evaluated (e.g., in [Alto, Brandetti et al. 1989]).

(editing), and pauses. Figure 6-1 shows a decomposition of the task completion time that corresponds to this decomposition: time necessary to generate the raw text (*generation time*), time for correcting and editing (*review time*), and *pause times*. The generation time can be subdivided further into time to compose text in the user's mind (*composition time*), and time to dictate the text (dictation time). The review time can be subdivided into time necessary to correct recognition errors (*correction time*), and time to revise and modify the text (*revision time*).

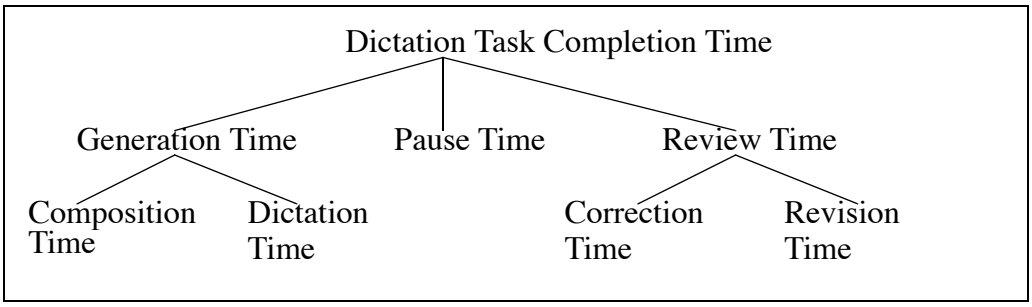


Figure 6-1. *Decomposition of dictation task completion time*

Depending on the type of dictation task used in an empirical study, some of these time measurements may not apply. For instance, in text composition tasks (such as composing a letter or a report), there are composition and revision times. By contrast, in text reproduction tasks, dictation task completion time decomposes solely into dictation, correction, and pause time. The latter decomposition will be relevant for the empirical evaluation of the multimodal dictation system presented in the next chapter.

Besides time as main quantitative measure, some studies investigate qualitative measures. These measures attempt to capture the process and the quality of the text produced, including:

- Ratio of time spent on correction, as a percentage of error-free time.
- Preference ratings, e.g., for modalities or methods employed.
- Quality ratings of the produced text, e.g., final errors (incorrect words left after the revision phase) or how convincing the text is.

In summary, various time measurements are commonly used in the evaluation of listening typewriters; however, other measurements such as quality of the output and user preferences may be equally important, especially for composition tasks. The remainder of this dissertation evaluates interactive error correction in the context of text reproduction tasks. Therefore, we have to deal only with some of the times from Figure 6-1, and they relate to measures defined in the next chapter as follows: "Dictation Time" as the time to actually dictate the text will be measured as *input speed* V_{Input}(dictate), "Pause Time" will be part of the *overhead time* $T_{Over-head}$, and "Correction Time" will be measured as *correction speed* $V_{Correct}$.

6.2 Studies on Dictation

Before reviewing of studies on dictation, this section comments briefly on the kinds of evaluation methodologies and potential confounding factors when comparing results across studies. Previous studies on dictation are reviewed, beginning with Gould's study on the way experts (business executives) dictate. This study evaluates handwriting, dictating to a machine, and speech, as dictation methods. It provides important baseline numbers on handwriting and text composition speeds. An important insight into the nature of dictation tasks is that the main factor limiting composition speed may not be input speed but dictation skill. This study was later extended to include a simulated listening typewriter. The follow-up study concluded that imperfect listening typewriters would be useful in composition tasks.

Roberts and Card introduce another line of work by presenting a methodology for evaluation of text editors. The methodology employs a set of benchmark tasks, which is relevant to interpret some other studies presented (since they adopted or extended this methodology). One such study evaluated the use of gestures for editing in text editors - an idea that this dissertation extended by using gestures as one of the multimodal error-correction methods. Finally, results from studies that used real (not simulated) listening typewriters are reviewed; an early study by IBM compared text reproduction by voice and keyboard input. The results of this study are not relevant for today's *continuous* dictation systems because voice input in this

study required the user to pause between every word. However, the experiment design of this study served as a starting point for the evaluation of the multimodal dictation system prototype. A later study evaluated the usefulness of IBM's continuous speech dictation product for creation of medical reports, concluding that dictation systems are not acceptable for highly trained professionals (such as radiologists) unless recognition accuracy is almost perfect.

6.2.1 Comments on Comparing Dictation Studies

This section points out issues that are important in interpreting the studies presented in the next section. Each study differ in its basic evaluation methodology and in a number of other ways, which makes comparisons among them difficult and which must be taken into consideration when interpreting the results.

The kind of task defines two main evaluation methodologies: a set of *benchmark* tasks vs. *text composition* tasks. All previous studies of text editors are based on performance measurements on benchmark correction tasks, applying Roberts and Card's evaluation methodology [Roberts and Moran 1983]. By contrast, Gould's studies on how experts dictate, and all studies on simulated listening typewriters, employed text composition tasks.

Caution must be exercised when interpreting the results of these studies, because various confounding variables must be considered. The most important potential confounding variables include:

- *Task*: Some studies evaluate text *reproduction* tasks (people reading some given text), and others investigate text *composition* tasks. Although both tasks involve text input into a computer system, the importance of text input speed differs for each; for text reproduction it is the main limiting factor, whereas for composition tasks, composition skill appears to have at least equal importance.

- *Typing speed* and *speaking rate*: Many studies do not control for either typing speed or speaking rate (which differ considerably across speakers, cf. [Pallett, Fiscus et al. 1994]).

- *Correction time*: Most studies exclude errors that take only a short time to recover (e.g., less than 15 seconds). This assumption is reasonable when time measurements are taken with stopwatches by human experimenters (such as in many of the earlier studies), because such measurements are not accurate for short intervals. However, ignoring all short errors also means that all typing errors are excluded from the analyses. Borenstein [Borenstein 1985] quantifies the inaccuracy introduced by this shortcoming. He estimates how much correction time is lost, across several error times that are being ignored. For example, ignoring all corrections less than 15 seconds long eliminates as much as 50% of the total time spent on corrections; cutting off at 5 seconds leads to an estimated 12.5% error.

- *Simulation issues* (applies to studies on listening typewriters only): Slow speed of listening typewriter can severely limit the potential gain in speed by a listening typewriter. Furthermore, the method that is used to simulate recognition errors is frequently questionable (unless real recognition errors are used).

6.2.2 Review of Experimental Results

The first formal evaluation of text-production methods was Gould's study on the way in which experts dictate [Gould 1978]. Business executives were instructed to compose two types of business letters (routine and complex) using four different methods: handwriting, handwriting without visual feedback, dictating to a secretary, and speaking (which means in this context pretending to speak in front of an audience). There were four key findings. First, dictating does not take a long time to learn (assuming the skill to compose text is already well developed). Second, the improvement in dictation speed through years of practice is only small, and far below the difference between novice dictation speed and speaking rate of up to 200 wpm

(words per minute), measured for reproducing some given text. The speaking rate while simultaneously composing the dictated text is much lower, in the order of 40 wpm. Third, there are significant qualitative differences between dictating and writing, primarily in the quality of the produced text. And finally, an easily reviewed external record, for example, in the form of writing appearing on the paper, appears to be important. This study investigated dictation in a realistic setting, and the participants represent users for which dictation systems are intended (highly trained professionals). The study provided baselines on input and composition speeds for handwriting and dictation. The study also addressed the more theoretical issue of whether dictation can be considered a skill. Furthermore, since even experts in dictation do not come close to the maximum speaking rate, composition skill is probably the main limiting factor for dictation performance.

Table 13: *Text production and composition times [Gould 1978]*

Method	Production Speed (wpm)	Composition Speed (wpm)	Time spent on planning
Handwriting	17	13	67%
Dictating to Secretary	35	19	67%
Simulated Isolated Word Listening Typewriter	36	8	78%
Simulated Continuous Speech Listening Typewriter (from [Gould, Conti et al. 1983])	57	16	72%

Table 13 summarizes the most important quantitative results of this study: a comparison of input (production) and composition speeds, and how much time is spent on planning (i.e., composition time in the terminology introduced in the previous section). The *composition speed* is the main performance variable in this table. The production speed of dictation to a secretary includes waiting for the secretary to catch up, clarifications, and instructions; therefore, listening typewriters can achieve significantly higher productions speeds. Isolated word listening typewriters slow dictation down since they require the user to briefly pause between every word, thus the significant lower production and composition speeds, compared with

continuous speech listening typewriters.

As a follow-up to this study, Gould investigated listening typewriters [Gould, Conti et al. 1983]. This study suggested that an imperfect listening typewriter would be useful in composing letters; listening typewriters were at least as efficient as traditional methods (handwriting, typewriting), and participants preferred the listening typewriter, despite the slow speed of the simulated listening typewriter. The study cautioned that speed and accuracy are a major concern, especially for experienced users.

That accuracy may be the most important factor in determining user acceptance of speech recognition applications was recently confirmed in a study of a commercial "real" listening typewriter on a report creation task [Lai and Vergo 1997]. Having the radiologists dictate the reports directly into a listening typewriter, rather than using transcription services, reduced the turn-around time (from when the radiologists dictates until he receives a written draft for revision) considerably overall. But the radiologists had to spent more of their time, because they had to correct the recognition errors. Consequently, the radiologists preferred the old method of dictating into a tape recorder and waiting for a typist to type the report. The study concluded that recognition is the most important factor determining acceptance of speech recognition technology by highly specialized professionals. However, delegating the correction of recognition errors to less trained personnels would reduce the time that the radiologists spend on creating a report considerably, and would probably lead to a much more favorable evaluation of dictation by the radiologists.

Roberts [Roberts and Moran 1983] developed a methodology to evaluate text editors and presented results from an empirical comparison of seven editors. The methodology compares text editors on a set of benchmark correction tasks. As dependent measures, time, errors, learning, and functionality are used. Since all of the editors still had command-line interfaces, instead of today's WIMP-based (Windows, Icons, Menus, Pointing) word processors, the results are no longer relevant. Moreover, the results on error performance are difficult to interpret since

all "small" errors were excluded from the analysis (cf. the remarks at the end of the introductory section in this chapter). The most valuable contribution of Roberts' work was the first formal evaluation methodology for text editors. Borenstein provided a critical review of Roberts' evaluation methodology and presents quantitative data on the effect of various confounding variables [Borenstein 1985]. Borenstein's study included more recent WIMP-based editors. He pointed out that the functionality dimension in Roberts' methodology will soon become out-dated as vendors continue to expand word-processor functionality.

In an empirical evaluation of a text editor, Kankaanpaa [Kankaanpaa 1988] applied Roberts' approach of measuring completion time for a set of benchmark editing tasks. The text editor used standard editing marks which the user draws directly on the display of a flat-panel touch-sensitive display. Instead of having real participants perform real tasks, the study predicted completion times using a keystroke-level GOMS model [Card, Moran et al. 1980]. Kankaanpaa concluded that editing using pen-drawn gesture is intuitive, and that the GOMS predictions matched well with his study.

IBM conducted an early experiment with a "real" (not simulated) listening typewriter [Alto, Brandetti et al. 1989]. The experiment evaluated text production speeds using keyboard and voice, including the time necessary to correct errors. The results of this study are outdated as well, because the system required the user to pause briefly between each word (*isolated word* dictation). However, the study is an important starting point for the design of experimental evaluation of a multimodal dictation system. The results suggest that document production is faster using voice rather than keyboard, and that large-vocabulary speech recognition can offer a very competitive alternative to traditional keyboard input. A confounding factor of the study is the slow typing speed of the chosen subjects: 25 wpm, which is slower than what is considered fast non-secretarial typing - 40 wpm.

Table 14 presents a summary of the most important quantitative results from all these studies, along with stipulations resulting from the experiment design. The fourth column indicates the

time spent on correction, as percentage of the time spent to compose/generate an initial version of the text (therefore, >100% is possible).

Table 14: *Summary of text production performance variables, gathered from various relevant studies*

Method	Study	Text Production Speed (wpm)	Time spent on Correction	Comments (Task, Problems)
Handwriting	[Gould 1978]	17	not measured	composition task
Dictating to Machine	[Gould 1978; Gould, Conti et al. 1983]	25	not measured	composition task; doesn't include time to transcribe
Dictating to Secretary	[Gould 1978; Gould, Conti et al. 1983]	30-35	not measured	see above
Text Editor	[Roberts and Moran 1983]	n./a.	12%	correction benchmark tasks; outdated editors, typos excluded
	[Alto, Brandetti et al. 1989]	20	20-25%	subjects slow typists
Isolated Word Listening Typewriter	[Alto, Brandetti et al. 1989]	21-29	70-130%	real system, experiment description not detailed enough to replicate
	[Gould, Conti et al. 1983]	36	23%	simulation, composition task
Simulated Continuous Listening Typewriter	[Gould, Conti et al. 1983]	57	18%	composition task

In summary, it is important to distinguish between text reproduction and text composition when considering dictation tasks. Text composition includes the creative act of putting some content into well-formed words and sentences, getting these words into machine readable format (to produce a nice layout using standard word processors, and for simplified revision) is only the secondary task. Concerning text production, handwriting achieves production speeds of less than 20 wpm, and traditional dictation to a secretary (with or without the help of a "dic-

tation machine", which is basically a tape recorder) achieves around 30 wpm. Isolated word listening typewriters reach input rates of 20-30 wpm, which is just about as fast as the traditional techniques. Consistently with that observation, commercial isolated word listening typewriters (i.e., automatic dictation systems from IBM, Dragon, and Kurzweil until 1997) were targeted only to user populations that didn't have a choice, such as people with Carpal Tunnel Syndrome or Repetitive Stress Injury (RSI). Simulation studies predict that continuous speech listening typewriters could increase text production speed to 50 wpm and more. Whether these speeds can be achieved depends on dictation accuracy of the underlying large vocabulary speech recognizer (determining the number of recognition errors to be corrected), and the efficiency of the offered correction methods. In anticipation of results presented in Chapter 8, current keyboard-less correction methods are too inefficient to realize this potential productivity gain, possibly one reason why commercial dictation products have not yet been a sweeping success.

7. Performance Model

Recognition-based human-computer interaction is defined in [Rhyne and Wolf 1993] as interaction method where user input must be recognized automatically prior to further interpretation. Examples for such modalities include the communication modalities that people naturally use, for example, speech, handwriting, and gestures. One main characteristic of recognition-based interfaces is that automatic interpretation of user input is usually imperfect. This is in contrast to most current computer interfaces that employ keyboard input and direct manipulation using a pointing device, where no recognition errors occur - except for human error. This chapter adopts Rhyne's terminology, because it allows us to discuss multimodal interaction in very general terms.

Given different input methods in recognition-based interfaces (that are feasible with current technology), an important challenge for the designer of such applications is to be able to predict which method users prefer. This chapter proposes a *performance model* of multimodal human-computer interaction that predicts input speed. *Input speed* was chosen as the main performance variable, because a rational user prefers methods that minimize time and effort spent on interacting with the system.

In related work, Mellor and Baber proposed a model of speech-based user interfaces that predicts task completion times using a critical path analysis technique [Mellor and Baber 1997]. Although their model addresses imperfect recognition performance and can also be applied to multimodal situations, it does not explicitly model the dependency of task completion time on modality, recognizer, and implementation-specific factors.

The first section presents our performance model of recognition-based multimodal interac-

tion. The second section applies the model to error correction in a multimodal dictation system. Predictions of correction speeds, input speeds (dictation system throughput), and the accuracies required for multimodal correction to be faster than correction by typing. In Chapter 8, we apply the model to extrapolate results from our empirical evaluation of the multimodal dictation system prototype, and to estimate the impact of recognition technology improvements on correction speeds and the productivity of future dictation systems.

7.1 Performance Model of Recognition-Based Human-Computer Interaction

This section presents a performance model of input in recognition-based interfaces that is based on a few basic and easily measurable parameters, including input rate and recognition accuracy of different input modalities. Since the model is inspired by the work on multimodal interactive correction, it can easily be applied to interactive correct; but it could be extended to recognition-based multimodal interaction in general. We first clarify what factors determine user effort, which in turn determine a rational user's preferences. The performance model of recognition-based multimodal input is presented in the second subsection. The model is validated by comparing predictions of correction speeds and dictation system throughputs with the empirical values determined during user evaluations.

7.1.1 Factors Determining User Effort in Recognition-based Interfaces

Effort of user input in recognition-based interface is determined by the following three primary factors:

- *Time* required by the user to provide the input (dictation time), and by the system to process it (response time)
- *Accuracy* of automatic input recognition
- *Naturalness* of interaction

While the first two factors are intuitive, the third factor requires some clarification. "Natural-

ness" is meant to capture how intuitive some method of human-computer interaction is for a user. These factors depend on user and task. For instance, some users have trouble typing and prefer keyboard-free input methods, and some tasks lend themselves better to speech input than others.

7.1.2 The Performance Model

Early work of Baber and Hone on modeling error correction pointed out that correction techniques are difficult to compare because their performance is closely related to their implementation, specifically to the performance of the recognition technology [Baber and Hone 1993]. These remarks apply generally to input in a recognition-based interface. The performance model presented here overcomes dependence on implementation, by factoring out different implementation-specific factors in the following way. The first subsection introduces input speed as main performance variable used in the model. Following subsections describe the model parameters and how these parameters can be estimated, the decomposition of input speed into the model parameters, and two refinements of the model. The final subsection validates the performance model on data from the user evaluation of the multimodal dictation system. Although the model is motivated by its origin in a dictation application, we formalized it for input in recognition-based interfaces in general.

7.1.2.1 Input Speed as the main Performance Variable

As the main performance variable, the performance model combines time and accuracy into one single measure: the *input speed (or system throughput)*. This definition of input speed incorporates recognition accuracy by measuring the total time until successful *completion* of input, including the time necessary to correct recognition errors (which may require several correction attempts). More generally, the main performance variable of the model is the number of items that the interaction can advance per unit of time. The type of input item depends on the application; for dictation, words are typically the input item, but for other applications, input items may correspond to semantic units or completed transactions. Similarly, the input

speed refers to different measures depending on the application. In a dictation application, the input speed (or better, *dictation system throughput*) refers to the number of words that the user can enter per minute (*words per minute*, wpm); for error correction, the input speed (or better, *correction speed*) denotes the number of errors that can *successfully* be corrected per minute (*corrections per minute*, cpm); and for a service transaction task, the input speed indicates how many transactions can be completed successfully per unit of time (e.g., *transactions per minute,* tpm).

7.1.2.2 Performance Model Parameters

The performance model includes explicitly the factors that determine input speed (as defined above), including user, modality, recognizer, and interface implementation. Interaction with a recognition-based interface using an input modality *m* is modeled by the following four parameters.

The first parameter is the *accuracy WA(m)* of a single attempt at communicating an item to the application. In a dictation system, the user attempts to input words, and WA(m) is the standard word accuracy[1] (hence its acronym). In the context of error correction, we use the term *correction accuracy CA(m)* instead of word accuracy, to distinguish between initial input and input that occurs while correcting recognition errors. Both word and correction accuracies are measured in percent (alluding to their interpretation as the probability of correct recognition).

The second parameter is the average *input time* $T_{input}(m)$ that it takes to communicate one item using modality *m* to the application. The input time is the inverse of the input rate, for example, speaking rate for speech, and writing speed for handwriting. The input time is measured in seconds per input item (word).

The third parameter captures the system response time; the *real-time factor R(m)* indicates

1. We use the term *word accuracy* as commonly defined in the speech recognition field, as the quotient of the number of correct items in the recognition hypothesis, diminished by the number of insertion errors, and the number of items in the reference.

how many times longer than real-time it takes to automatically recognize (and interpret) user input. In the following, R(m) = 1 means that there is no delay at all, i.e. recognition finishes at the same time as user input.[1]

Finally, the *overhead time* $T_{Overhead}(m)$ summarizes all other times that are necessary to complete an interaction in modality *m*, which is measured in seconds per correction attempt. The overhead time includes the time the user needs to plan or select an appropriate interaction method (if there is a choice), and the time the user spends on initiating an interaction in *m,* (e.g., the time to press a button, to pull down a menu, or to move the hand to the screen to write or gesture on it). Hence, the overhead time depends both on modality and interface implementation.

How can the model parameters be estimated?

- **Input time T_{Input}***:* In most applications, commonly known standard estimates can be used. Although input rates vary across people, they are specific to a certain input modality, so they do not change across applications. The following estimates for input rates (in words per minute, wpm) were taken from the studies on dictation methods (as presented in the previous chapter): handwriting 15-20 wpm, continuous speech 100-200 wpm, (fast non-secretarial) typewriting 40 wpm. For some modalities, the input rates depend on their usage within an application. As an example, the input rate of gestures depends on the size and complexity of the gestures. Furthermore, even some of the above "standard" input modalities (speech, handwriting) may be employed in novel interaction methods in such a way that the input rate changes, e.g., for handwriting and spelling in partial-word corrections.

- **Word/Correction Accuracy WA(m)/CA(m)**, **Real-time Factor R(m)**: The accuracy and real-time factor depend on the recognition system, but estimates can eas-

1. Achieving R=1 is obviously a very challenging goal. However incremental recognition algorithms make it possible to start recognition while input is still being generated.

ily be derived using benchmark recognition tasks for each new release of a recognizer. Assuming a constant word accuracy across repeated input attempts (a simplifying assumption, as Chapter 8 will show), the random variable describing the number of interaction attempts until success has a geometric distribution, and Equation 7-1 shows the simple formula for the expected mean *E[.]* of such a geometric distribution.

$$E[N(m)] = \frac{1}{CA(m)}$$

Equation (7-1): *Estimates for average number of interactions until success (assuming CA is constant across multiple correction attempts)*

- **Overhead Time $T_{Overhead}(m)$**: The overhead time depends on the task, the available interaction methods, and the implementation. Therefore, the overhead time may be different for each new version of an interface. Instead of measuring the overhead, reasonable guesses could be used.

Table 15 in Section 7.1.2.5, page 156, shows estimates for all model parameters as measured in the user study of the multimodal dictation system.

7.1.2.3 Decomposition of Input Speed in the Performance Model

The core of the performance model is how input speed (as defined in the first subsection) is expressed as a function of the four model parameters (presented above). The first equation presented in this subsection describes how the input speed depends on the time for each input attempt and the number of attempts. The second equation models the time per input attempt with a simple affine relationship. As a notational comment, all parameters are modeled as random variables, because they vary across users. For simplicity, the following description uses the expected means as if they were regular variables.

Assuming statistical independence between time per interaction and accuracy, the *total time* to successfully complete an interaction using modality *m* is the product of $T_{Attempt}(m)$, the time

necessary for one interaction attempt, and $N(m)$, the number of attempts until the input item has been communicated successfully (i.e., the first attempt plus any correction attempts that may be necessary). The input speed $V_{Input}(m)$ in items (words) per minute is the quotient of 60 seconds (one minute) and the total time:

$$V_{Input}(m) = \frac{60}{N(m) \cdot T_{Attempt}(m)}$$

Equation (7-2): *Factorization of input speed into time per attempt and number of attempts*

To further decompose the time per attempt, we model recognition-based multimodal interaction by the following steps: the user plans the interaction, chooses an interaction method (modality), provides the necessary input, waits for the system to interpret the input, and finally decides whether correction is necessary or whether to proceed in the task at hand. How much time does such a multimodal interaction require? The steps of planning, choosing the modality, and the preparation of the actual input correspond to one unit of overhead time (per correction attempt). Then, user input in modality m and its automatic interpretation take $R(m)$ times $T_{input}(m)$ seconds. The time necessary for one attempt at entering an item in modality m follows:

$$T_{Attempt}(m) = T_{Overhead}(m) + R(m) \cdot T_{Input}(m)$$

Equation (7-3): *Basic decomposition of time per attempt into overhead, input and system response time*

The model is generally applied by replacing some of its parameters with appropriate estimates, while other parameters correspond to the independent variable of the problem under question. For example, in Section 7.2.1 we predict the correction speed (of unimodal and cross-modal correction by repeating input) as a function of correction accuracy. The input rates are replaced by standard estimates, overhead times and real-time factors are set to certain values, and the correction accuracy is the independent variable.

7.1.2.4 Refinement of the Performance Model

The previous section presented the performance model in its simplest form, suitable for recognition-based multimodal correction by repeating input in one modality. With appropriate refinements, the model can be applied to more complex situations. This subsection refines the model in two ways, appropriate for its application to multimodal correction on a dictation task. First, correction accuracy does not stay constant in repeated correction attempts, but it deteriorates; second, applications typically provide heterogeneous sets of interaction modalities, which may include modalities that are not interpreted with imperfect recognition technology, rather than a single modality.

The first refinement of the model modifies how the average number of input attempts is estimated. Equation 7-1 assumed that recognition accuracy stays constant in repeated interaction attempts. However, in anticipation of an important result of the next Chapter 8, we know that correction accuracy deteriorates when input is repeated in the same modality. To model this deterioration of correction accuracy in a simple way, we assume that the accuracy for the first correction attempt is CA_1, while the accuracy of following correction attempts (in cases of repeated misrecognitions) is only CA_2. With this assumption, the expected mean number of attempts until success is:

$$E(N) = CA_1 + (1 - CA_1)\left(1 + \frac{1}{CA_2}\right)$$

Equation (7-4): *Estimate for average number of input attempts until success (the correction accuracy is CA_1 for the first correction attempt, and CA_2 for further correction attempts)*

The second refinement generalizes the model from interaction in a single modality to heterogeneous sets of interaction modalities, some of which may not be interpreted using imperfect recognition. For example, multimodal interactive correction, as implemented and evaluated in this dissertation, includes both recognition-based modalities (speech, spelling, handwriting, and gestures) and other, not recognition-based interaction methods (e.g., correction by choos-

ing from alternative words). We extend the model to include heterogeneous correction methods as follows:

- Correction by choosing from a list of alternatives is not interpreted using imperfect recognition. We model correction by choosing from a list as one correction attempt that is successful with chance *CA(list)%*.

- Correction using editing gestures addresses different types of correction tasks than correction by respeaking, spelling, or handwriting; words are deleted or the cursor is positioned typically *before* the user corrects by respeaking, spelling, or handwriting. Therefore, the refined models editing gestures separately as $N(gest)T_{Attempt}(gest)$, in addition to corrections by repeating input.

- Finally, multimodal error correction offers users a choice between different correction modalities, for example, respeaking, spelling, and handwriting. We model user choice between these different modalities *m* by empirical usage frequencies *freq(m)*.

Thus, to model correction by choosing from a list of alternative, editing using pen-drawn gestures, and an additional set *M* of correction modalities to insert or replace words, the average correction speed is estimated as:

$$V_{Correct}(M) = \frac{60}{CA(list)T_{At_t}(list) + (1 - CA(list))\left(N(gest)T_{Attempt}(gest) + \sum_{m \in M} freq(m)N(m)T_{Attempt}(m)\right)}$$

Equation (7-5): *Decomposition of correction speed for choosing from alternatives ("list"), pen-drawn gestures ("gest"), and a set M of correction modalities.*

The empirical user evaluation of interactive error correction - which we will use to validate the performance model - compared *interactive error correction methods* (i.e., two sets of correction modalities): *Conventional correction* by choosing from alternatives, editing using key-

board and mouse, and typing (hence M={typing})[1], and *multimodal correction* by choosing from alternatives, editing using pen-drawn gestures, and repeating using speech, spelling, and handwriting (thus M={speech, spelling, handwriting}).

7.1.2.5 Validation with Interactive Multimodal Error Correction

The performance model presented here was validated on data from the user evaluation of the multimodal dictation system. We derive predictions for the speed of conventional and multimodal correction (as introduced in the previous paragraph) and compare them with the correction speeds measured during the user evaluation. Section *7.2.3* later in this chapter presents further evidence validating the performance model, by comparing predictions of dictation speed with data from the user study.

To apply the standard method for model validation - estimating model parameters on a training set, and evaluating the accuracy of model predictions on a separate test set - we divided the data from our fifteen participants - five users from each of the three categories of typing skill. The training set consists of three participants in each category of typing skill (for a total of nine participants), and the test set consists of two in each category (for a total of six).

Using the training data, we estimated the following model parameters: input rate for editing with gestures and correction by spelling, word accuracies and recognition response times for all modalities, and overhead times. Table 15 shows the estimates for word-level correction modalities with 95% confidence intervals in parentheses. The input speeds are shown in words per minute (wpm), assuming each word is counted as separate error. Chapter 8 presents a more detailed analysis, which distinguishes between whole-word and partial-word corrections, and describes details on how the data was collected.

1. Here, we model typing as input modality in the same way as introduced for recognition-based modalities: typing is characterized by an input speed, a correction accuracy, and an overhead time.

Table 15: *Performance model parameters for interactive error-correction modalities, as estimated from training data (95% confidence intervals in parentheses)*

	Choosing from List	Respeaking	Spelling	Handwriting	Editing Gestures	Typing	Keyboard Editing
T_{Input} (wpm)	58(25)	47 (5)	26 (6)	18 (4)	36 (6)	17 (7)	n/a
Correction Accuracy CA	21% (8%)	36% (23%)	80% (17%)	71% (8%)	86% (6%)	84% (5%)	82% (8%)
Real-time Factor R	1	2.6	1.5	1.3	1.0	1.0	n/a
$T_{Overhead}$ (sec/correction)	4.6 (0.5)	5.4 (2.1)	4.3 (0.7)	3.5 (1.1)	5.0 (0.8)	2.6 (0.7)	4.3 (1.0)

To derive predictions for correction speeds using the refined model, as expressed in Equation 7-5, the following estimates are still missing: estimates for the usage frequencies *freq(m)* for different correction modalities, and estimates of the input rate of correction by typing, given the typing speed[1]. Table 16 shows estimates for the relative usage frequencies obtained during the user study, and Figure 7-1 shows a linear regression analysis that predicts the input of rate of correction by typing, as a function of typing speed. The standard error is 8.0 (N=16), and r^2=0.24; i.e., 24% of the variance is accounted for by the linear relationship. Given the small number of available samples, this simple linear model is sufficiently accurate for the purposes of this dissertation.

Table 16: *Usage frequencies of repeating using speech, spelling, and handwriting*

Modality	Relative Frequency
Respeaking	55.8%
Spelling	10.6%
Handwriting	29.1%
Spelling PWC	1.4%
Handwriting PWC	3%

1. The input rate for correction by typing is significantly lower than the speed of typing a whole piece of text, because the user has to position his hands on the keyboard before each correction.

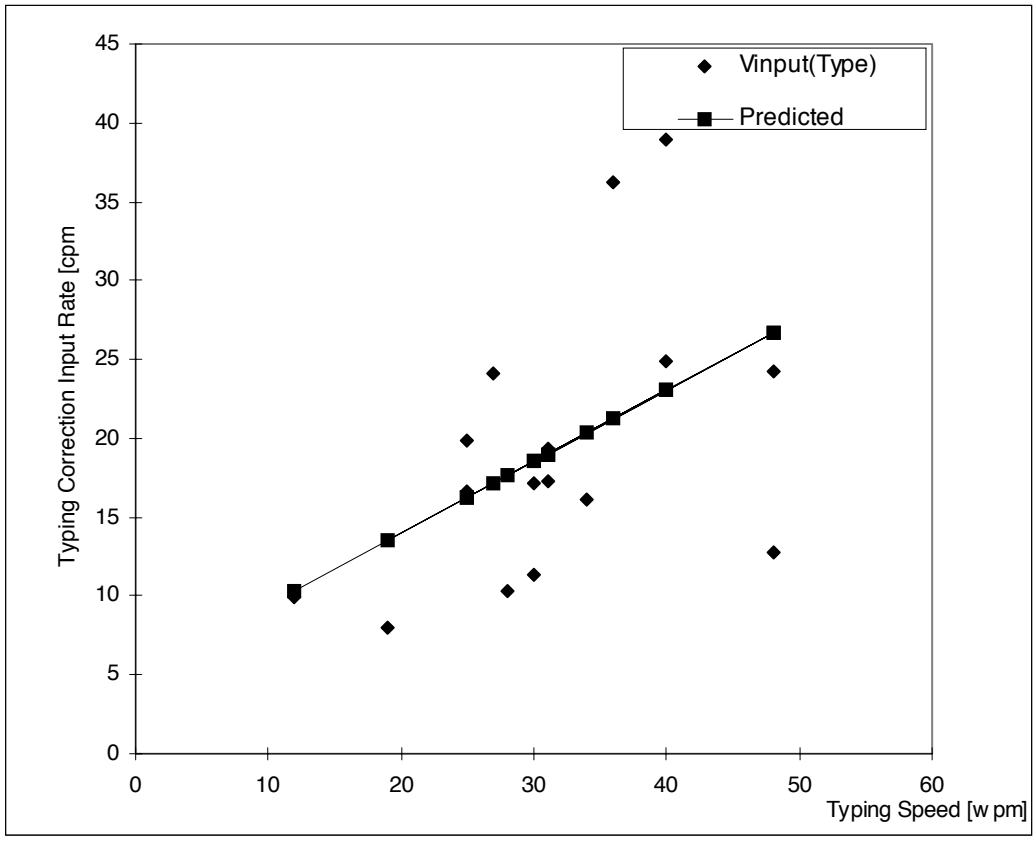


Figure 7-1. *Linear regression analysis predicting the input rate of correction by typing $V_{Input}(type)$, given the typing speed*

Now that all necessary pieces have been assembled, we can predict correction speeds using Equations 7-4 and 7-5, and the parameters as shown in Table 15. Table 17 compares the correction speed estimates obtained from the performance model with the actual measured correction speeds (averaged across the 6 participants of the test set). The average absolute deviation between model predictions and actual values is 17% for multimodal correction (N=12, 6 participants of the "test set" in both "multimodal" and "multimodal & error highlight condition, as described in Chapter 8), and 12% for correction using keyboard & list, across different categories of typing skill (N=6, two participants in each of the three categories of

typing skill). This deviation is within the reasonable range for such empirical models.[1]

Table 17: *Validation of the performance model, comparing measured correction speeds (averaged across participants of test set) with model predictions*

Correction Method	Average measured Correction Speed [cpm]	Predicted Correction Speed [cpm]	Signed Model Error
Multimodal	4.5	3.7	-18%
Keyboard & List ("slow" typing)	5.9	6.2	5%
Keyboard & List ("average" typing)	6.2	7.0	13%
Keyboard & List ("fast" typing)	7.3	7.2	-1%

As can be seen, the performance model of input in recognition-based interfaces can successfully be applied to modeling multimodal interactive error correction, and model predictions match empirical data sufficiently well.

7.2 Application to Multimodal Interactive Correction and Dictation

This section applies the performance model to the following three important questions about interactive error correction in a multimodal dictation system:

1) How does correction speed depend on recognition accuracy and modalities, and how will future performance improvements affect the comparison of unimodal and multimodal correction?
2) What recognition accuracy is necessary so that multimodal correction is faster than correction by typing?
3) What is the total system throughput of a multimodal dictation system, as a function of dictation accuracy and error correction methods?

To further validate the performance model, the final subsection will also compare predictions for dictation speed with data from the user evaluation. The next chapter will apply the model

1. A discussion of goodness-of-fit measures and justification for using average absolute error of prediction can be found in [Kieras, Wood et al. 1997].

to extrapolate the results from the user study, for example, to real-time recognition in all modalities.

7.2.1 Correction Speeds with Imperfect Recognition

Correction speed depends on the modality and the performance of available recognizers. To predict speed of correction by repeating in modality m, $T_{Attempt}(m)$ in Equation 7-2 is replaced by Equation 7-1 (assuming correction accuracy were constant across repeated attempts), and the input rates $T_{Input}(m)$ are replaced by the estimates shown in Table 15. In anticipation of faster computers in the future, recognition in real-time is assumed for all modalities ($R(m)=1$), and the overhead time $T_{Overhead}(m)$ is set to 3 seconds to normalize for implementation-specific differences[1]. Figure 7-2 plots the total input speed for correcting by repeating in continuous speech (respeaking), spelling, and handwriting, over the recognition accuracy in each of these modalities.

Figure 7-2 shows that at best, with 100% recognition accuracy, correction by respeaking would achieve 24 corrections per minute (cpm), and correction by handwriting 15 cpm. This compares favorably to correction by typing for users with good typing skills, who achieved 12 cpm in our user evaluation.

Furthermore, we can use Figure 7-2 to predict under what conditions (unimodal) correction by respeaking could be as efficient as multimodal correction. Since speech is the fastest modality for text input, speech would also be the most effective correction modality, if recognition was accurate enough. At which level of accuracy would corrections by respeaking outperform multimodal correction? For example, multimodal corrections by spelling are 80% accurate

1. The variation of overhead times $T_{Overhead}$ in our empirical data (cf. Table 15) is due to implementation specific differences in how the modalities are triggered (e.g., pressing a button to initialize spoken correction takes longer than starting to write on the screen for handwriting and gesture corrections), and modality choice patterns (e.g., users preferred to try speech first, and therefore the overhead time for speech corrections contains relatively more time spent on searching for errors, compared with other correction modalities). For the present consideration, we want to abstract from these differences, and assume the same overhead time for all modalities, near the lower end of empirically observed values.

with current recognizers (cf. Table 15). The arrows in Figure 7-2 reveal that correction by respeaking would be faster if they were more than 60% accurate. (Note that the assumptions of this prediction require that this accuracy remains constant across repeated correction attempts.) By comparison, Table 15 shows that corrections by respeaking currently achieve only 36% accuracy; explaining why multimodal correction is faster than unimodal correction by respeaking in the multimodal dictation system prototype. More generally, similar predictions can help the designer of (multimodal) speech user interfaces to decide which correction methods are most efficient.

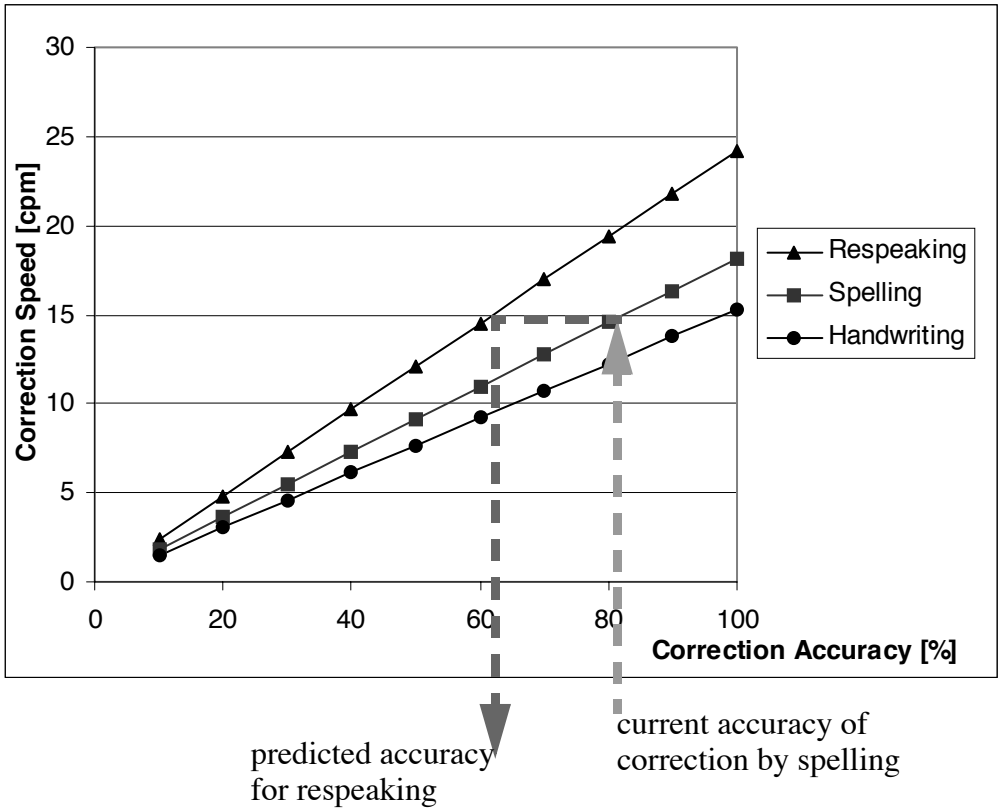


Figure 7-2. *Predicted correction speed for multimodal interactive correction*

7.2.2 Is Multimodal Correction faster than Typing Correction?

Since multimodal input and correction can be seen as an alternative to (traditional) keyboard input, the comparison to typing speed is particularly interesting when discussing correction in a dictation system. To compare multimodal correction and correction by typing, we predict

which accuracies are necessary (across different correction modalities) to outperform correction by typing (across different typing skills). This question can be answered easily by comparing predictions for correction speeds, as a function of modality and correction accuracy (as derived in the previous subsection), with speeds of correction by typing (such as estimated in Figure 7-1). Figure 7-3 shows the correction accuracies (shown on the y-axis) necessary to achieve certain speeds of correction by typing (shown on the x-axis).

In anticipation of a result from the user study (see Section 8.3.3, page 185, in the next chapter), average computer workers (who generally are fast non-secretarial typists) can correct up to 15 errors per minute using keyboard and choice from the N-best list. To reach this correction speed, accuracy for corrections by respeaking would have to be recognized at more than 60% accuracy. This accuracy may appear quite low, but poses a challenge when maintained across repeated correction attempts.[1] To reach the same correction speed, corrections by spelling would have to be 85% accurate, and corrections by handwriting almost 100% accurate - assuming the whole-word correction.

If advanced correction techniques were used, which require only partial input for correction (e.g., partial-word corrections as described in Section 4.3.3), and if overhead times could be further reduced (e.g., by reducing the time spent on locating errors), the accuracy required to correct faster than by typing would be correspondingly lower. Hence, multimodal correction is not hopelessly slower than keyboard correction, and model predictions allow the designer of future applications to determine whether multimodal correction is actually faster than correction by typing.

1. With the recognizer used for this dissertation work, correction by respeaking were 42% accurate in the first correction attempt, which deteriorated to 20% and 0% in the second and third attempt (for more details, see Figure 8-2 in the next chapter).

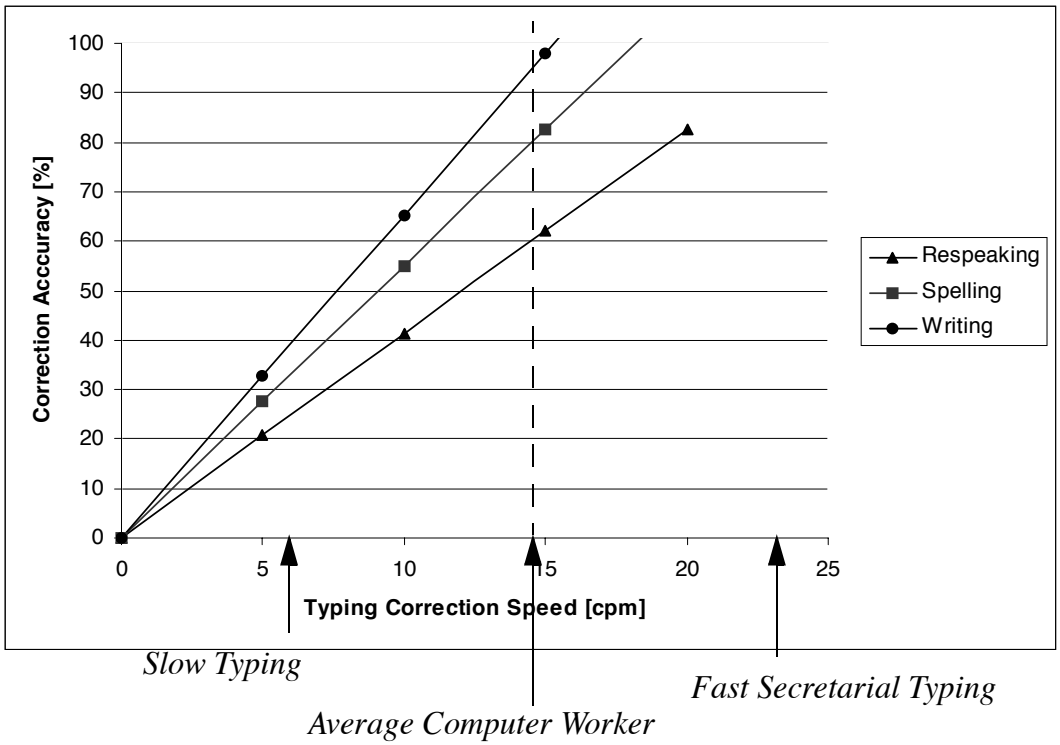


Figure 7-3. Predicting correction accuracies necessary to beat typing in correction speed

7.2.3 Dictation Speed

In addition to comparing correction speeds, the performance model allows predictions of the *dictation system throughput*, i.e., the dictation speed including time necessary to correct errors. The dictation speed can be predicted based on word accuracy *WA(dictate)*, recognition speed *R(dictate)* of the dictation recognizer, and the correction speed $T_{Correct}(m)$ of a correction method *m*, as described in the following paragraph.

Text production with a dictation system consists of three steps: dictation, automatic interpretation of spoken input, and correction of recognition errors. How much time do these steps require? A user with speaking rate $V_{Input}(dictate)$ *(wpm)* dictates $wordN=V_{Input}(dictate) * 1 min$. words in one minute. The speech recognizer needs $T_1=R(m)*1min$ to interpret this dictation input. During automatic interpretation of the dictation input at accuracy *WA(dictate)*, on the average *errorN=wordN*(1-WA(dictate))* recognition errors occur. The correction of these

recognition errors requires $T_2=errorN*T_{Correct}(m)$, where $T_{Correct}(m)$ is the inverse of the correction speed $V_{Correct}(m)$ using method m. The total time to enter *wordN* words including correction time is thus $T=T_1+T_2$, leading to the following formula for the dictation system throughput, as a function of correction method m and dictation accuracy $WA(m)$:

$$V_{Dictate}(m) = \frac{V_{Input}(dictate) \cdot 1min}{R(m) \cdot 1min + V_{Input}(dictate) \cdot 1min \cdot (1 - WA(dictate)) \cdot T_{Correct}(m))}$$

Equation (7-6): *Estimation of dictation speed based on dictation accuracy, recognition speed and correction speed*

For example, with a state-of-the-art large vocabulary dictation recognizer that performs at 90% accuracy in real-time, and multimodal correction that achieves a correction speed of 8 errors per minute, a dictation speed of $V_{Dictate}$=49 words per minute can be expected. This technique of predicting system throughput will frequently be used in the evaluation chapter.

Table 18: *Validation of dictation speed predictions*

Correction Method	Average measured Dictation Speed $V_{Dictate}$ (wpm)	predicted Dictation Speed $V_{Dictate}$(wpm)	Signed Model Error
Multimodal	16	14.8	-8%
Keyboard & List ("slow" typing)	16.3	16.7	+2%
Keyboard & List ("average" typing)	18.4	18.1	-2%
Keyboard & List ("fast" typing)	25.0	18.4	-26%

The empirical evaluation of the multimodal dictation system offers the opportunity to further validate the performance model, and in particular the Equation 7-6. Table 18 compares empirical dictation speeds with predictions from Equation 7-6, using the following parameters that were measured in the user study: $V_{Input}(dictate)=133$ *wpm,* $WA(dictate)=75\%$, and correction speeds of various methods as presented in Table *17*. Across the different correction methods and participants, there is an average absolute deviation of 18% (N=18). The high deviation for "Keyboard&List" with "fast" typing is probably due to the small number of independent "test"

samples (only two participants in the "fast" category).

In summary of this chapter, the performance model of input in recognition-based interfaces is a first step towards formalizing multimodal interaction. We applied the model to interactive multimodal error correction, and validated it using data from the empirical evaluation of the multimodal dictation system prototype. Several interesting predictions were derived, including how multimodal correction compares to conventional correction using keyboard and mouse input, how multimodal can unimodal correction may compare with future improved recognizers, and how the system throughput of a dictation system depends on dictation accuracy and available correction methods.

How does the presented performance model relate to cognitive models in the field of human-computer interaction, such as the well-known keystroke-level model [Card, Moran et al. 1980]? Similar to the keystroke-level model, it is a model for a specific kind of user input to a computer system; the model is tailored to recognition-based human-computer interaction. Unlike the keystroke-level model, it is derived from intuition and not backed by a cognitive theory of human-computer interaction. Its usefulness for modeling error correction in recognition-based user interfaces has been shown; however, it is unclear how well it generalizes to other kinds of recognition-based interaction. With the deployment an increasing number of speech recognition (and multimodal) applications, there is a need for an evaluation methodology that goes beyond standard benchmark evaluations in the speech recognition and related fields. The model presented in this dissertation is a first step towards a more general evaluation methodology for future multimodal interfaces.

8. Experimental Evaluation

In contrast to model-based evaluation (as presented in the previous chapter), user studies examine how users interact with either a real or a simulated application. Empirical data on the human-computer interaction is gathered using interaction logs, videotapes, informal observations, interviews, and questionnaires.

This chapter presents a user study that evaluates multimodal interactive error recovery in the context of the multimodal dictation system prototype, which was developed in this dissertation. Section 8.1 describes the research questions that the study addressed, including when and why multimodal interactive error correction is beneficial, and how various error-correction methods compare. The subsequent Section 8.2 describes the experimental design, and discusses potential confounding variables and how they were addressed. Section 8.3 presents quantitative and qualitative results from the user study and the post-experimental questionnaires. The final section discusses the results and explains some key observations based on performance and cognitive variables.

As primary evaluation measures, correction accuracy and speed are used. *Correction accuracy* was defined in the previous chapter as word accuracy (as commonly used in the speech recognition field) on correction input. *Correction speed* was defined the rate at which corrections can be successfully completed, and *input speed (or system throughput)* as the rate at which items can be entered, including the time necessary for corrections.

The chapter includes many details of the experimental design. The reader can gain a basic understanding of the experiment and the results from Sections 8.1, 8.2.1, and 8.3.

8.1 The Research Questions and Hypotheses

To develop the research questions for the user study of the multimodal dictation system, the reader may remember the research question of this dissertation, which was stated in Chapter 1: given unreliable speech recognition technology, how can the user's effort necessary to recover from interpretation errors be minimized? This dissertation proposed multimodal interactive error recovery as solution for speech recognition applications with a graphic user interface. What is the role of an empirical evaluation of multimodal interactive correction? It has to provide evidence for the effectiveness of multimodal interactive correction, and determine relative strengths and weaknesses of different correction modalities. With that knowledge, a designer of speech recognition applications can make educated decisions on how to address the problem of error correction, and reduce the user's effort necessary to correct errors.

The remainder of this section elaborates how evidence for the effectiveness of multimodal interactive correction modalities can be provided, and how their relative strengths and weaknesses can be identified. Each subsection starts out with a research question and hypotheses, followed by some explanations. The first three subsections address the fundamental issues of multimodal versus unimodal correction, and the usefulness of a multimodal dictation system. The final subsection discusses additional research questions that address issues specific to multimodal interactive error correction in the context of dictation applications.

8.1.1 Ineffectiveness of Unimodal Correction and Effectiveness of Multimodal Correction

Research Question (1): *Why is unimodal correction ineffective, and why is multimodal correction effective?*

Hypotheses (1): Recognizing corrections in the same modality is difficult, and recognition performance of (most) current recognizers deteriorates on corrections in the same modality. Switching modality significantly increases correction speed, compared with correction in the same modality.

These key issues of interactive error correction have already been discussed several times. The reasons why unimodal correction by respeaking may be ineffective were identified in Section 4.3.1. However, no previous research has provided empirical evidence for either the claim that unimodal correction is ineffective nor the claim that multimodal correction could speed up correction. The user study on multimodal error correction will test both these claims.

8.1.2 Comparison of Interactive Correction Methods

Research Question (2): How do multimodal correction methods compare with conventional methods, and how do different multimodal correction methods compare with each other?

Hypotheses (2): With current recognition technology, multimodal correction methods are faster than (unimodal) correction by respeaking, and slower than correction by typing for users with good typing skills.

As identified in our informal survey of interactive correction methods, conventional correction methods are limited to correction by respeaking, typing, or choosing from a list of alternatives. Based on intuition and on predictions derived from the performance model, we expect that with current recognition technology, multimodal correction is faster than (unimodal) correction by respeaking, and that keyboard correction remains the fastest correction method. Multimodal correction should be particularly attractive for applications without a keyboard (e.g., small mobile devices) and for users with poor typing skills.

8.1.3 Usefulness of the Multimodal Dictation System for Text Reproduction Tasks

Research Question (3): *Is the multimodal dictation system useful for text reproduction tasks?*

Hypotheses (3): Dictation (including correction time) with the multimodal dictation system compares favorably to average non-secretarial typing.

Looking at the whole text production process, a multimodal dictation system allows the user to produce text at a rate that compares favorably to typing the whole text. A multimodal dicta-

tion system exploits high input rates on continuous speech in the text generation phase, and avoids time loss in repeated recognition errors by offering to switch modality.

8.1.4 Issues in Multimodal Interactive Error Recovery in the Context of Dictation Applications

The previous subsections stated the fundamental research questions of multimodal interactive error correction. But a number of more specific issues occurs in the context of dictation. The discussion of multimodal interactive error recovery in Chapter 4 proposed a number of methods that could potentially improve the error-correction process in the context of dictation applications, but no evidence was presented. This section introduces each of these issues, in the style of the previous subsections, as pair of research question and hypothesis, followed by brief discussions.

Research Question (4): Does partial-word correction help?

Hypothesis (4): Partial-word corrections increase input speed, correction accuracy, and overall correction speed.

Since partial-word corrections require less user input, input speed should increase; and since the vocabulary reduction algorithm for partial-word corrections dramatically reduces the number of possible alternatives, correction accuracy should also increase. However, it is unclear whether users sufficiently learn to identify situations where partial-word corrections are beneficial, and whether the hardware allows the user to execute partial-word correction with sufficient accuracy.

Research Question (5): Are gestures and pointing (as described in Section 4.3.2) more efficient for simple editing tasks, compared with editing with keyboard and mouse input?

Hypothesis (5): Editing gestures and pointing outperform traditional keyboard and mouse based methods.

Multimodal interactive error recovery proposed pen-drawn gestures and pointing for simple editing tasks including selecting errors, positioning the cursor, and deleting words. It is not clear whether such gesture-based interaction is more efficient than conventional direct manipulation interaction techniques for a wide range of manipulation tasks.

Research Question (6): Can system-initiated location of errors (as described in Section 4.2) speed up error correction?

Hypothesis (6): Imperfect automatic locating of errors is useful for dictation tasks if the method ensures sufficiently high accuracy.

As mentioned earlier, several researchers have suggested that automatic flagging of recognition errors (e.g., based on confidence measures) could facilitate error correction. The multimodal dictation system prototype offers an opportunity to test this hypothesis on a dictation task. The crucial issue is what accuracy is necessary so that highlighting errors automatically is useful.

8.2 Experimental Design

Given so many research questions, how can one experiment possibly answer all them? There is no single such experiment. Instead, the experiment design for the user study of the multimodal dictation system addresses several research questions in several sub-experiments. In addition, some research questions were sufficiently answered in an extensive pilot experiment that preceeded the final evaluation. The first subsection presents the basic experimental design. The second subsection discusses potential confounding variables and how the experimental design avoids them. The final subsection presents alternatives for the experimental design, and it motivates the choices made in the design of our user study.

8.2.1 Method

The method of a user study specifies the main elements of an experimental design: task (what participants do during the experiment), experimental conditions (the dependent and indepen-

dent measures, and how the independent measure are varied systematically across the conditions), procedure (the different steps of the experiment), who participated in the study, data coding, and data analysis (what measures are taken during the experiment, and how the obtained data is statistically analyzed). The following subsections specify each of these elements for the user study of the multimodal dictation system. All written materials and forms used for the user study can be found in Appendix A. This material includes instructions to the participants, a quick tutorial on the use of the multimodal dictation system, the various dictation tasks, the post-experimental questionnaire, and the consent form.

8.2.1.1 Task

This section describes the experimental task and a few technical details of the recognizers used during the experiment.

Participants were instructed to dictate either one or more sentences, which were chosen from the Wall Street Journal, and to correct all recognition errors, using different sets of correction modalities. After reading a sentence, the multimodal dictation system displayed the current text on the writing-sensitive display (touchscreen). Recognition errors were not simulated, but the current text contains the actual errors that a state-of-the-art large vocabulary dictation recognizer made on the participant's speech. Then, participants visually located recognition errors, selected them by tapping on the screen, and corrected them using one of the available correction modalities. The goal was to get every sentence correct word by word, and as fast as possible.

Figure 8-1 shows a snapshot of the prototype's GUI during tutorial mode, where an additional area contains instructions for the next practice task, along with a button to move from one task to the next.

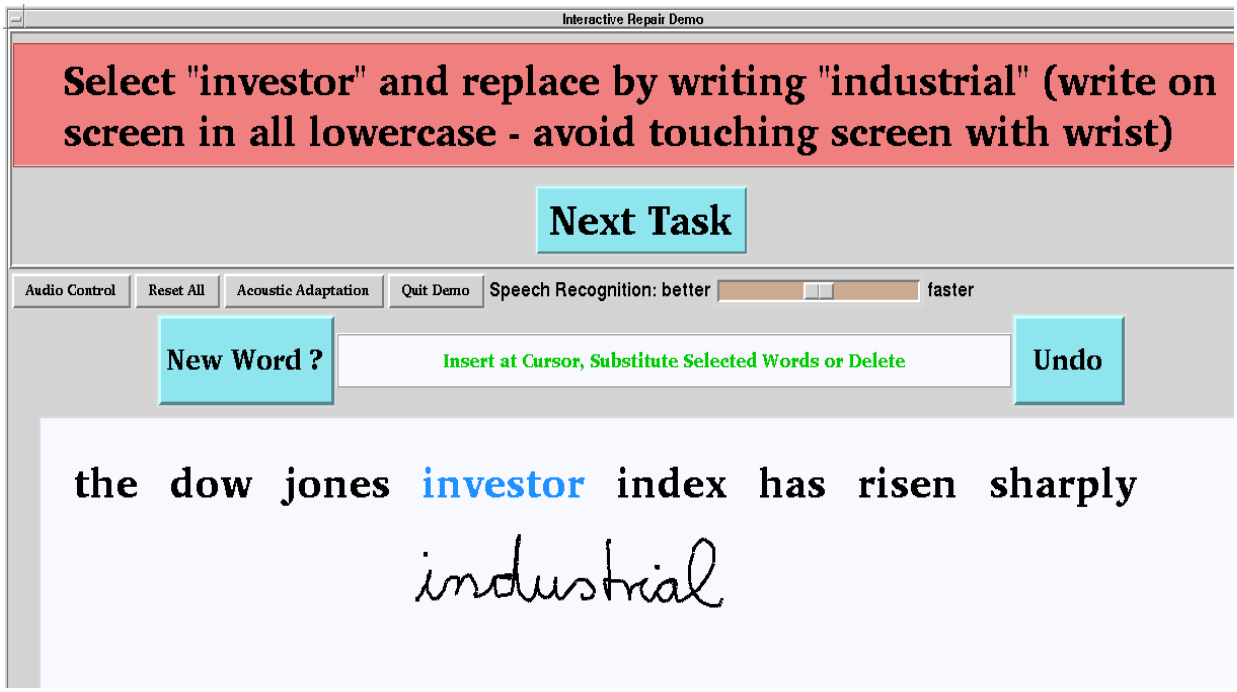


Figure 8-1. Snapshot of multimodal dictation system prototype in tutorial mode. The bottom frame contains instructions for practice tasks. The current text input (recognition hypothesis) is displayed in the main frame. User pen-input ("industrial") is displayed visually.

Some technical details of the recognizers that go beyond the description of the multimodal dictation system prototype in Chapter 5. The vocabularies were based on the standard 20,000 vocabulary from the last official evaluation on the Wall Street Journal speech recognition benchmark test in November 1994 [Pallett, Fiscus et al. 1994]. To eliminate any out-of-vocabulary words[1], all words that occur in the experimental tasks were included in the vocabulary. For continuous speech recognition and N-gram context modeling, the standard '95 60K language model (distributed by NIST and the LDC) were used.

8.2.1.2 Experimental Conditions

The experimental conditions differed in the set of modalities that are available to locate and correct recognition errors. In all conditions, the participant had to locate recognition errors visually and select them. However in one condition, the system highlighted words that were

1. The problem of out-of-vocabulary words is another important problem of speech user interfaces, which was not addressed in this thesis work.

likely to be incorrect.

The experiment conditions compared keyboard-free multimodal correction with conventional correction methods, with and without keyboard input.[1]

- *Conventional correction methods* ("*Keyboard & List*" in descriptions below): Selecting words and positioning the cursor by mouse-click, replacing and inserting words by typing, replacing words by choosing from a (visually presented) list of alternative words, deleting words by hitting the backspace or delete key.

- (Keyboard-free) *Multimodal correction methods* ("*Multimodal*" in descriptions below): Selecting words by tapping on screen, positioning the cursor and deleting by editing gestures; replacing words by choosing from a visually presented list of alternatives; replacing/inserting words using continuous speech, spelling, and handwriting; and partial-word corrections using gestures, spelling, or handwriting.

Table 19 shows the experimental conditions used in the *final* user study. The rows indicate which of the correction modalities are available in each experimental condition. In "Keyboard & List", the user located errors and corrected them using the set of conventional correction methods. In the two multimodal conditions, errors were corrected using the set of (keyboard-free) multimodal modalities. However, they differed in how errors are located: in "Multimodal", the user located errors, whereas in "Multimodal with imperfect error-locate", the system highlighted errors automatically. Highlighting errors was imperfect, too, because automatic location of errors is based on unreliable measures of confidence measures that have only about 87% accuracy in classifying words as correct versus incorrect.

1. Strictly speaking, since in a dictation task the primary input is typically provided by dictating using continuous speech, correcting errors by typing is actually "multimodal" (in the sense that modality is switched for correction), and correcting by respeaking is "unimodal". The terminology chosen for the sets of correction modalities is motivated by the fact that the "conventional" correction methods typically appear in today's speech recognition applications, and require mouse and keyboard input, whereas the "multimodal" methods allow the user to correct errors "multimodally", without keyboard input.

Table 19: *Experimental conditions for the final user study*

Experimental Condition	Keyboard & List	Multimodal (user locates errors)	Multimodal (system highlights likely errors)
Choose from list of alternatives	X	X	X
Respeaking		X	X
Spelling		X	X
Handwriting		X	X
Typing/Mousing	X		
Editing Gestures		X	X
Partial-Word Correction		X	X
Imperfect Auto-Locate of Errors			X

We chose a within-subject, repeated measures experimental design. That means each participant performed all experimental conditions, and the experimental measures were compared across conditions with the same participant. A repeated measures design was chosen to limit the impact of the known high variance of recognition performance across users as confounding variable.

The *pilot* experiment used other experimental conditions than the final study, as shown in Table 20. The goals of the pilot experiment was different from the final study: examine the relative effectiveness of different multimodal correction modalities, and establish empirically that (unimodal) correction by respeaking is ineffective. Correction by respeaking or choosing from a list ("Respeak&List") was compared with several multimodal correction methods, including multimodal correction that allows the user to switch modality only once ("Spelling&List", "Handwriting&List"), and multimodal correction that allows choice among all correction modalities, except for keyboard input ("multimodal, no PWC (Partial-Word Correction)", "multimodal"). Note that correction using keyboard and mouse did not include the possibility of choosing from the list of alternatives, unlike the "conventional correction" condition

in the final study.

Table 20: *Experimental conditions for the pilot experiment*

Experiment Condition	Respeak & List	Spelling & List	Handwriting & List	Multimodal no PWC	Multimodal	Keyboard & Mouse
Choose from list of alternatives	X	X	X	X	X	
Respeaking	X			X	X	
Spelling		X		X	X	
Handwriting			X	X	X	
Typing/ Mousing						X
Editing Gestures				X	X	
Partial-Word Correction					X	

8.2.1.3 Procedure and Participants

An experimental session consisted of three phases: tutorial, experiment, and debriefing phase:

- *Tutorial Phase (45-60 minutes):* First, the experimenter demonstrated the multimodal dictation system using the quick tutorial (see Appendix A). Then, the participant practiced using the multimodal dictation system on a set of simple practice tasks (cf. Figure 8-1). In more practice tasks, the participant learned to master each correction modality, gaining experience in the relative effectiveness of different modalities. Finally, the participant performed a set of trial tasks, using all multimodal correction modalities. After this session, all participants showed sufficient familiarity with the different correction modalities.

- *Experimental Phase (60-90 minutes):* The experiment itself was subdivided into four sessions. In the first three sessions, the participant performed three experimental conditions: "keyboard&list", "multimodal", and "multimodal with imper-

fect auto-locate". The order of experimental conditions and dictation tasks varied randomly across participants. In the fourth session, the participant's typing speed was measured.

- *Debriefing:* The participant filled out the post-experimental questionnaire. Then, the experimenter explained the purpose of the experiment and answered any questions from the participant.

Participants were recruited from the campus community of Carnegie Mellon University. Posts were distributed on general electronic newsgroups and physical boards in selected buildings. A monetary incentive of $25 was given to reach a more diverse participant population. Participants included students and administrative staff, they were balanced in gender, and most participants did not have any prior experience with speech-recognition software. The typing speed of the participants was systematically varied according to self-reported low, average, and high typing speeds.

8.2.1.4 Measures and Data Coding

We measured performance at the level of a single input modality using the following three measures: input rate (i.e., how man word can a user enter per minute), system response time (i.e., how much time does automatic recognition require), and recognition accuracies. These measures correspond to parameters of the performance model - input time, real-time factor, and accuracy. Performance at the task-level was assessed using the correction speed $V_{Correct}(m)$ and the total input speed (or system throughput) $V_{Input}(m)$, as defined in the introduction of this chapter. In addition to these quantitative measures, a post-experimental questionnaire (see Appendix B) assessed some qualitative variables, including ease of learning, perceived user preference, and subjective variables such as intuitiveness and perceived strain of a correction modality.

During experimental sessions, data was collected in two ways. First, the prototype multimodal

dictation system created a time-stamped record of all spoken, written, and typed user interactions. This record was later manually annotated with the correct system response for each interaction, to assess recognition accuracies. For analysis of modality choice patterns, the record also contained for each recognition error the sequence of modalities used, until successful correction. All sessions were videotaped - the second method of data collection.

8.2.2 Experimental Design Alternatives

The experimental design as described in the previous sections evolved in a longer process, and many design decisions were made along the way. This section makes these design decisions explicit, and defends the chosen design against possible alternative designs. Thus, the reader may gain insights for future user studies of multimodal applications.

8.2.2.1 Basic Experimental Design

In designing a user study of any novel method or system, there are two basic approaches: First, an experiment that compares the novel method with previously known methods. For that purpose, previous systems (or methods) either have to be used directly in the experiment, or they have to be reproduced with sufficient accuracy. The former causes difficulties since capabilities of previous methods and the new experimental task have to be reconciled, and this is notoriously difficult. Second, an experiment can compare different methods referring to previously published or commercially available systems, but not claiming an explicit comparison. In the case of the multimodal dictation system, any direct comparison with other systems would be severely confounded by the differences in the baseline accuracies of the recognition systems (which this dissertation did not attempt to improve). The second approach was therefore chosen.

8.2.2.2 Task Variables.

- *Choice of dictation task*: Dictation tasks are either the reproduction of some given text, or the composition of new text. While the latter is more realistic, it implies a

shift in the experiment's focus from the problem of correcting errors to the problem of supporting text composition. This dissertation focuses on error-correction methods, therefore, a text reproduction task was chosen.

Furthermore, the experimental design made a conscious choice for the modality of the initial input, namely for continuous speech with an eye on dictation applications. In other applications for which multimodal error correction is applicable, the typical modality for initial input may be different, for example handwriting in personal assistants (like Apple's Newton, or 3Com's Palm Pilot).

Another choice regarding the dictation task is the type of text that is given to participants. Since the continuous speech recognizer that was available for building the multimodal dictation system was trained on the Wall Street Journal database, the dictation tasks consisted of sentences from the Wall Street Journal, to maximize recognition accuracy.

- *Choice of correction task*: To investigate error correction in an automatic dictation system, the correction task can be constructed in two ways, either a set of benchmark correction tasks (e.g., as in Roberts' evaluation methodology for text editors [Roberts and Moran 1983]), or (simulated or real) recognition errors while participants perform a dictation task. Since simulations of recognition errors is difficult, and to be able to assess the overall usefulness of a multimodal dictation system as tool for dictation, this user study evaluated error correction within a dictation task.

- *Task realism:* The choice of reproduction of some given text limits the validity of the experimental results for dictation applications, since many dictation tasks are composition tasks. Other types of tasks could have been included, in particular typical tasks for other applications that belong to the category of non-conversational speech recognition applications with a graphic user interface, for example numerical data entry. This dissertation defers such experiments to future work.

  Another trade-off in task realism was taken with respect to the issue of out-of-

vocabulary words: a more realistic dictation task would include out-of-vocabulary words, instead of eliminating them by adding all words within the experimental tasks to the recognition vocabularies. However, the new word problem is a separate research challenge beyond the scope of this dissertation. As rudimentary solution, the multimodal dictation system prototype had an "ADD WORD" button which allowed the user to type in new words that would automatically be added to all recognition vocabularies. However, since this solution does not address the (more important) issue of deciding whether a recognition error is due to a new word or to something else, this feature was not formally evaluated (and remains a challenge for future research).

8.2.2.3 Experimental Conditions.

Experimental conditions, and the inclusion of a control group are important choices in any experimental design. Since a basic design that compares different correction method was chosen for the multimodal dictation system study, rather than comparing to a specific existing system, a control group was not necessary. However, other choices were made regarding the experimental conditions.

The pilot experiment compared different multimodal correction methods explicitly. Separate experimental conditions offered different sets of multimodal correction modalities (i.e., correction only by respeaking, only by spelling, or only by handwriting).

After significant improvements of recognition performance on initial dictation and corrections by respeaking, the final user study examined more high-level issues, in addition to validating results from the pilot experiment. Such high-level issues include relative strengths and weaknesses of multimodal and conventional methods, and user preferences. Therefore, the experimental conditions compare only two correction methods: one with free choice among keyboard-free multimodal correction modalities, and the other with free choice among "conventional" correction modalities (with keyboard and mouse)

8.3 Results

This section presents the results of the user evaluation of interactive error correction within a multimodal dictation system. The first subsection summarizes statistics describing the data collected during the evaluation. The following subsections discuss results for each research question, in the same order as when they were introduced in Section 8.1.

8.3.1 The Data from the User Study

In the final user study, data from fifteen participants was collected, five participants in each category of (self-reported) typing speed (slow, average, fast); whereas the pilot experiment included only six participants. By pooling the data across experimental conditions for every input modality (continuous speech, spelling, handwriting, gesture, typing), we built a database of multimodal corrections. Table 21 shows important statistics of this database, which includes data from both pilot experiment and final study. The rows are: initial dictation using continuous speech; word-level corrections using continuous speech, spelling, handwriting, choosing from the list of alternatives, and typing; editing tasks (deleting and placing the cursor) using gestures and mouse/keyboard; and partial-word corrections using spelling, handwriting, and gestures. Note that speech input allows the user to enter multiple words at a time, therefore, the table shows the number of words in speech interactions as separate number. This database was used in previous chapters to evaluate algorithms that improve correction accuracy (see Section 4.4), and to estimate the basic correction modality parameters for the performance model (see Section 7.1.2).

Table 22 characterizes the dictation performance and the absolute amount of errors during the final study. The word accuracy on the initial dictation was with below 80% significantly lower than 90+% current commercial dictation systems claim. However, unlike those systems, we did not adapt the acoustic models of the dictation recognizer to the current user's voice. Such speaker adaptation requires 100-200 sentences as training data, and reading these sentences

would have extended the duration of the experiment excessively.

Table 21: *Database of dictation and multimodal corrections*

Modality	Number of Interactions
Continuous Speech, Initial Dictation	503 Sentences (9750 Words)
Respeaking	515 Repairs (1778 Words)
Spelling (word level)	816 Words
Handwriting (word level)	1301 Words
Choose from list of alternatives	478 Words
Typing	685 Words
Gestures (word level)	747 Corrections
Editing with Mouse/Keyboard	431 Corrections
Spelling (character level)	40 Corrections
Handwriting (character level)	65 Corrections
Gestures (character level)	206 Corrections

Table 22: *Dictation and error statistics (final study)*

Experimental Condition	Word Accuracy on Initial Dictation	Errors
Keyboard & List	73%	867
Multimodal Correction	77%	593
Multimodal with Imperfect Error-Locate	78%	739

8.3.2 Ineffectiveness of Unimodal Correction - Effectiveness of Multimodal Correction

This section answers research question 1 - why unimodal correction is ineffective, and why multimodal correction is effective. The results of the study confirm that unimodal correction by respeaking is ineffective, and that multimodal correction is effective. Empirical evidence is provided in two ways: first, by comparing the recognition accuracy on initial and correction input, and second, by comparing the correction speed of unimodal with multimodal correction.

Comparing correction speed of unimodal with multimodal correction is more costly, because it requires to isolate unimodal and multimodal correction for each modality in separate experimental conditions. This design was chosen for the pilot experiment. Correction by respeaking ("Respeak &List" condition) is compared with different multimodal correction methods: multimodal methods that allow switching modality once ("Spelling&List", "Handwriting&List" conditions), or free choice among all modalities ("Multimodal" condition).

Table 23: *Correction speeds (cpm=corrections per minute)*

Correction Method	Correction Speed $V_{Correct}$ (cpm)
Respeak & List	2.3
Spelling & List	5.3
Handwriting & List	5.2
(Free choice among) Multimodal	4.5
Multimodal, no PWC	4.9
Keyboard & List (depending on typing skill)	6.0 - 7.3

Table 23 shows the (empirical) correction speed of different correction methods. As can be seen, correction by respeaking ("Respeak&List") is slower than any multimodal correction method ("Spelling&List", "Handwriting&List", "Multimodal, no PWC", "Multimodal", "Keyboard &List"). An analysis of variance reveals that unimodal correction by respeaking is significantly slower that any multimodal correction method ($F(5,25)=27.33$, $p<0.05$).

Comparing recognition accuracies across repeated correction attempts offers a second way to establish ineffectiveness of unimodal correction and effectiveness of multimodal correction. The recognition accuracies in different modalities can be tabulated across correction attempts *in the same modality*. Figure 8-2 shows the correction accuracies assuming that the original input was in speech. Note that the counter for the correction attempt is reset after each switch of modality, even if the same recognition error is corrected. For example, if some recognition error required three correction attempts, the first two in speech, and the final attempt in hand-

writing, this final attempt is assigned to category "1", because it was the first attempt after a switch of modality.

First, if users repeat input in speech, correction accuracy is much lower (only 40%) than if users switch to a different modality (75% for handwriting, 80% for spelling). If multiple correction attempts are necessary, correction accuracy in successive attempts remains high if the user switches modality after each attempt. In terms of Figure 8-2, this means staying within category "1". An analysis of variance confirms that correction accuracy is significantly lower if repeated in the same modality ($F(2,6)=36.2$, $p<0.01$).

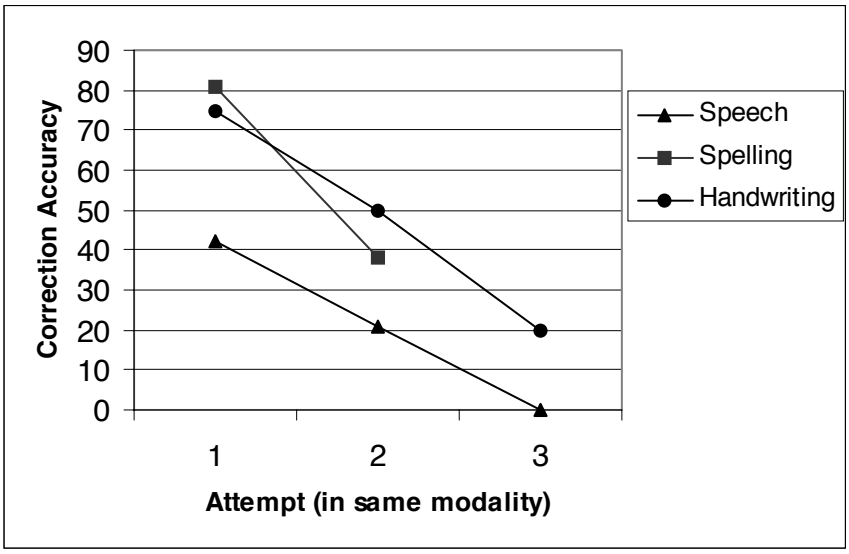


Figure 8-2. *Deterioration of correction accuracy on repeated attempts in the same modality*

Second, Figure 8-2 reveals that the correction accuracies are significantly lower than word accuracies known from standard benchmark tasks (which are in the 80-90% for the modalities reported here). Recognizing correction is more difficult than standard benchmark tests (remember: subset of more difficult to recognize words, short words, hyperarticulation). Despite the difficulty of recognizing correction input, correction accuracy is higher if modality is switched. For example, the figure below indicates around 80% for multimodal correction using spelling or handwriting, instead of 40% for unimodal correction by respeaking.

8.3.3 Comparison of Interactive Correction Methods

Addressing research question 2 - how multimodal correction methods compare to conventional methods - this dissertation focuses on two measures: task completion time and user preferences (as indicated by modality choice patterns). Different correction methods can be compared using their correction speed, combining input speed and recognition accuracy into a single measure, as described in Chapter 7. Alternatively, we can look at what correction methods users prefer, given free choice. In both cases, every instance of an error and its correction constitutes a data point for a correction method. Thus, statistics for both correction speed as well as user choice of correction method can be accumulated across an experimental condition.

This section compares interactive correction methods quantitatively and qualitatively:

- Quantitatively on the level of single correction interactions, by deriving basic modality parameters (input rate, recognition accuracy, recognition speed, and overhead times), and on the level of completed correction interaction sequences, by comparing overall correction speeds.

- Qualitatively by examining modality choice patterns and preference ratings (as participants indicated in the questionnaires).

In summary of the results, the user study shows that multimodal correction is faster than unimodal correction by respeaking, and slower than correction by typing for users with good typing skills. Regarding user preferences between modalities, there is evidence that correction accuracy has a significant influence on user preference between different modalities. However, speech is preferred initially.

The section contains a lot of material; Tables 24, 26 and Figure 8-4 contain the main results.

8.3.3.1 Quantitative Comparisons

As described before, the basic parameters for multimodal correction (as defined in the perfor-

mance model) can be estimated by pooling the data of all repair interactions across all experimental conditions. Table 24 shows estimates derived from the database of multimodal corrections, for input rate V_{Input}, word accuracy *WA*, real-time factor *R*, and overhead time $T_{Overhead}$. The width of 95% confidence interval is indicated in parentheses.

The following bulleted list discusses the data concerning word-level corrections (i.e., the upper half of this extensive table). The interpretation for the partial-word data (bottom section of table) is done in Section 8.3.5, and for the different modalities for performing editing tasks (middle section of table) in Section 8.3.6.

Table 24: *Dictation and correction modality parameters (final study)*

Modality	Input Rate V_{Input} (wpm)	Accuracy WA/CA (%)	Real-time Factor R	Overhead Time $T_{Overhead}$ (sec)
Continuous Speech, Initial Dictation	133 (9)	75% (5)	2.6	3.5 (0.5)
Correction by Respeaking	45 (4)	35% (13)	2.6	5.0 (1.4)
Spelling (word level)	24 (4)	79% (16)	1.7	4.0 (0.8)
Handwriting (word level)	17 (2)	75% (6)	1.4	3.5 (0.9)
Choose from list of alternatives	48 (15)	24% (6)	1.0	4.8 (0.4)
Typing	19 (5)	87% (3)	1.0	2.8 (0.5)
Gestures (word level)	39 (4.3)	86% (4)	1.0	4.9 (0.5)
Editing with Mouse/Keyboard	20 (8)	80% (7)	1.0	3.8 (0.8)
Spelling (partial-word correction)	41 (14)	95% (8)	1.5	2.5 (0.8)
Handwriting (partial-word correction)	25 (3)	81% (15)	1.2	2.5 (0.8)
Gestures (partial-word correction)	31 (5)	46% (11)	1.0	5.5 (1.1)

- *Input rate:* The input rates V_{Input} shown are consistent with those in the literature, in particular, for continuous speech dictation and handwriting [Gould 1978].

- *Correction by respeaking:* The accuracy on corrections by respeaking is fairly low due to a very low accuracy on isolated word corrections. The recognizer used in this study performs at 10% on isolated word recognitions, as opposed to 51% on multiple word corrections. The poor performance on isolated word corrections

appears to be a weakness of the specific recognizer used. However, the significant decrease in accuracy from dictation to corrections by respeaking generalizes across different recognition systems (as shown in Section 4.3.1, page 83).

Respeaking is an input rate of 46 wpm significantly slower than dictating with 133 wpm ($p<0.01$). People elongate pauses between words and enunciate more clearly in speech repairs [Oviatt, Levow et al. 1996], which explains this slow-down in speaking rate on corrections. This observation limits the head start of continuous speech as correction modality, compared with other modalities such as handwriting and spelling.

- *Correction by spelling and handwriting*: Our data shows clearly why switching modality for corrections is so effective: instead of 35% accuracy for unimodal correction by respeaking, cross-modal correction by spelling and handwriting is 75-80% accurate. Cross-modal partial-word corrections are even more accurate. However, recognizing correction is a more challenging task than standard benchmark tests for spelling and handwriting recognition, because the accuracy is well below the 95% that both recognizers achieve on benchmark data.

- *Correction by choosing from a list of alternatives:* The low accuracy of choosing from a N-best list of alternatives indicates that the correct alternative is frequently not among the top choices. The magnitude of this effect is specific to the recognizer used in this study, but the effect appears to generalize across different recognition systems [Huang 1998].

  The input rate for choosing from a list of alternatives is determined by two factors: first, the time necessary to trigger the display of the list (in the multimodal dictation system by holding down on a word for a second), and second, the time necessary to scan the list and decide whether the correct alternative is among them.

- *Correction by typing:* Participants made a surprisingly high number of typing errors, hence the accuracy of correction by typing is far below 100%. The speed of correcting by typing at 19 wpm is an average across participants of different typing skills.

  Table 25 shows the measured average typing speeds and speed of correction by typing in more detail. As can be expected, correction by typing is slower than typing consecutive text, because it takes time to position the hands over the keyboard and to focus on what to type as correction for each correction. Furthermore, the "slow" category did not cover people with really poor typing skill, because 23 wpm is still a quite decent typing speed. This deficiency in the distribution of the participants can be compensated by predictions for really slow typing speeds using the performance model. Refer to Figure 7-1 in the previous chapter for a linear regression between typing speed and speed of typing correction that was derived from our experimental data.

Table 25: *Typing skills and speed of correction by typing (final study)*

Category	Typing Speed (wpm)	Input rate V_{input} of Correction by Typing (cpm)
"slow"	23	9.1
average	35	20.0
fast	40	29.3

Interactive correction methods can also be compared at the level of successfully completed repair interactions, by adding the times required to successfully correct an error, across different correction methods. Table 26 shows the correction speeds $V_{Correct}$ as measured in the experiment, based on data from the pilot and the final study as indicated. As can be seen, correction by respeaking is significantly slower than any method that allows the user to switch modality for correction (i.e., all other methods). Correction with keyboard input remains unchallenged in speed for users with good typing skills. Surprisingly, the experimental condi-

tions that gave free choice among non-keyboard correction methods ("multimodal, no PWC" and "multimodal") do not perform best. The reason is the low accuracy of corrections by respeaking, combined with the tendency of participants to try speech first, which frequently fails, and thus they have to spend time on an additional correction attempt.

Table 26: *Comparison of correction speeds (from pilot and final study)*

Correction Method	Correction Speed $V_{Correct}$ (cpm)
Respeak & List (Pilot)	2.3
Spelling & List (Pilot)	5.3
Handwriting & List (Pilot)	5.2
Multimodal with Imperfect Error-Locate	4.0
Multimodal (Pilot and Final Study)	4.5
Multimodal, no PWC (Pilot)	4.9
Keyboard & List ("slow" typing)	5.9
Keyboard & List ("average" typing)	6.2
Keyboard & List ("fast" typing)	7.3

To get an estimate of the influence of learning on the speed of multimodal correction, we measured the correction speed that the developer of multimodal correction achieves: 6.8 cpm. Learning therefore plays a significant role, and the participants in the experiment still applied multimodal correction in a suboptimal manner. Experienced users can achieve 7 cpm correction speed, which compares favorably to correction by typing for users with good typing skills.

8.3.3.2 Qualitative Comparisons

The design of the multimodal error-correction user studies allows us to analyze qualitative issues based on user preferences in the "Multimodal" conditions (which gave the user free choice among all keyboard-free correction methods), and based on answers in the post-experimental questionnaires. We are going to address the following issues: What correction modal-

ities are used most frequently? What drives the choice of correction modality? And finally, how do users perceive multimodal correction in comparison with conventional correction methods?

Table 27: *Empirical usage frequencies of modalities (final study)*

Correction Modality	Multimodal	Keyboard & List
Respeaking	27%	n/a
Spelling (whole word)	5%	n/a
Handwriting (whole word)	14%	n/a
Choose from List	10%	12%
Gesture	34%	n/a
Spelling (partial words)	1%	n/a
Handwriting (partial words)	1%	n/a
Gesture (partial words)	1%	n/a
Typing	n/a	50%
Mousing / Keying	n/a	38%

Table 27 addresses the issue of modality choice, showing the usage frequencies of the various correction methods across the experiment conditions in the final user study. The following explanations may help the reader understand this data. First, gesture and Mousing/Keying are necessary for most correction in both conditions (to delete words and to place the cursor), therefore they have to be used frequently. Second, the high usage frequency of correction by respeaking is noteworty, in view of repeated experience that respeaking is quite inefficient (35% average accuracy). This provides evidence for our explanation why multimodal correction is slower than correction by spelling/handwriting & list, as seen in Table 26, namely that participants continually try to correct by respeaking, wasting time since respeaking frequently fails. Third, participants use handwriting more often than spelling as correction modality - although spelling is faster and more accurate (cf. Table 24), suggesting that handwriting is a more intuitive (or "natural") correction modality. Finally, partial-word corrections are used only very infrequently. Apparently, it is difficult for users to judge when partial-word correc-

tions may be beneficial, and most users initially ignore the possibility of partial-word correction.

What factors drive user modality preferences? The multimodal approach assumes that correction accuracy is a major factor; if correction accuracy is higher in one modality, users are willing to switch to it. Do users learn which modalities are more efficient? Do all users converge on the same modality, or does the preferred modality differ across users? To address these questions, we analyzed modality choice patterns in the "multimodal" experiment condition, and how they develop over time (at least during the experimental session).

Figure 8-3 shows estimates for the usage frequency of different modalities. One time unit corresponds to forty correction interactions, and the whole x-axis to one experimental session (which lasted for about one hour). The figure illustrates that participants learn to prefer effective modalities and avoid ineffective modalities. Over time, the usage frequency of effective modalities increases, while that of ineffective modalities decreases. The most effective modality is different for the two participants. Therefore, it is beneficial to offer multiple modalities even though some modality may be inferior to another on average, such as handwriting compared to spelling.

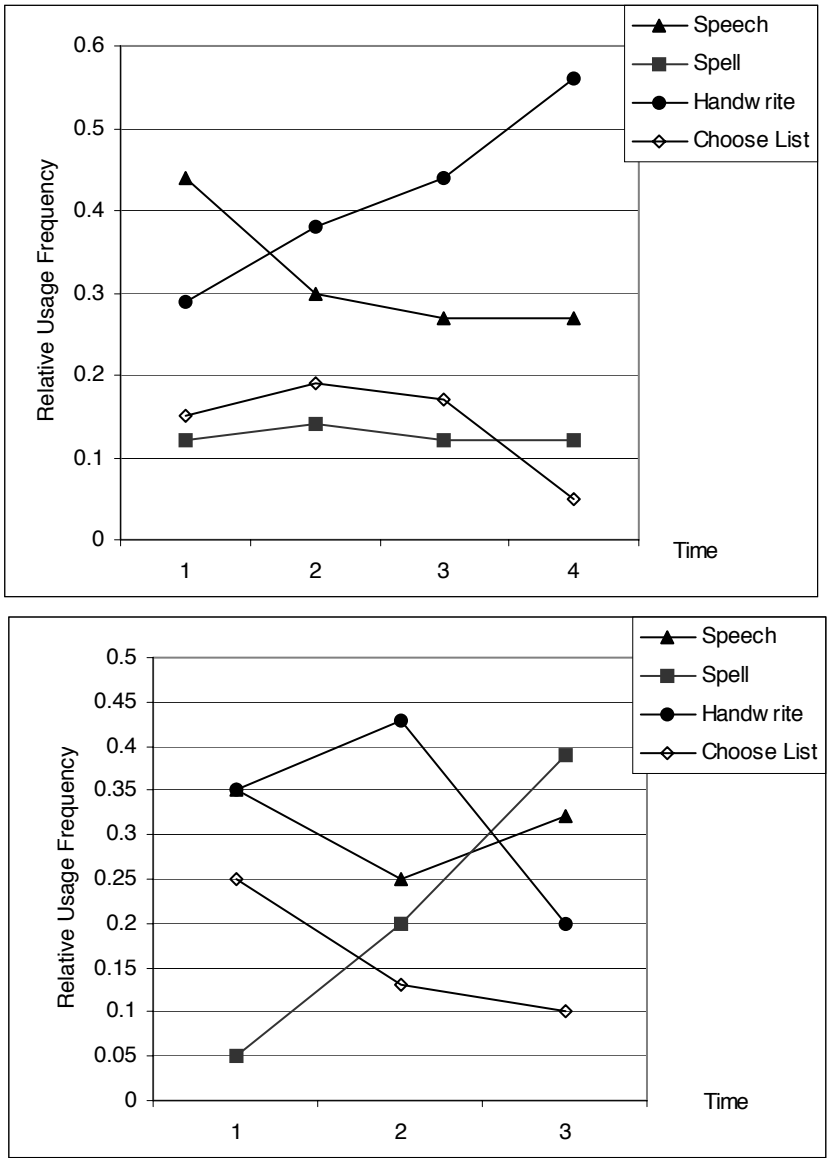


Figure 8-3. *Usage frequencies of different modalities for two typical users. Speech corrections were 32% / -32% (upper/lower graph) accurate, spellings 40% / 85%, handwritings 68% / 85%, and choosing from list 18%/7%. In the upper case, the user learns to avoid speech and spelling and favor handwriting; in the lower case, the user learns to avoid choosing from list, preferring spelling.*

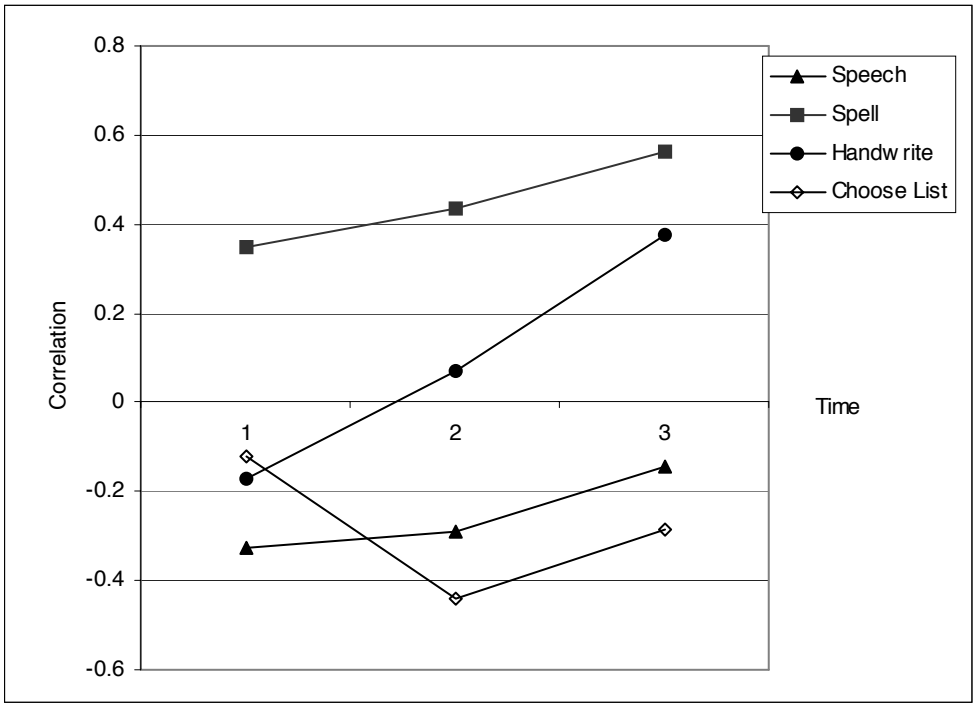


Figure 8-4. *Correlation of usage frequency with effectiveness of correction increases with practice, but a bias towards correction by speech remains.*

Figure 8-4 suggests that these observations are not incidental. To show that users prefer effective modalities, we computed the correlation of modality usage frequency with modality effectiveness (high correction accuracy means high effectiveness score). A positive correlation means that users prefer effective modalities. Except for the "choose from list" modality, the correlation between usage frequency and effectiveness increases over time, and this effect is significant ($F(2,4)=7.25$, $p<0.05$). The correlation for spelling and handwriting is positive, indicating that users follow the rational choice for these modalities (of preferring the more effective modality). However, user behavior for the speech modality is different. The negative correlation indicates that users choose speech as correction modality despite evidence that speech is ineffective for correction. The fact that the correlation gradually becomes less negative suggests that eventually, users learn that speech is ineffective.

Figure 8-5 provides further evidence for the user bias towards speech, by showing modality usage frequencies in the first correction attempt. As can be seen, speech is preferred in the

first correction attempt, and only very slowly speech is used less frequently in favor of spelling or handwriting. The modality differences in this figure are significant ($F(3,9)=28.1$, $p<0.01$).

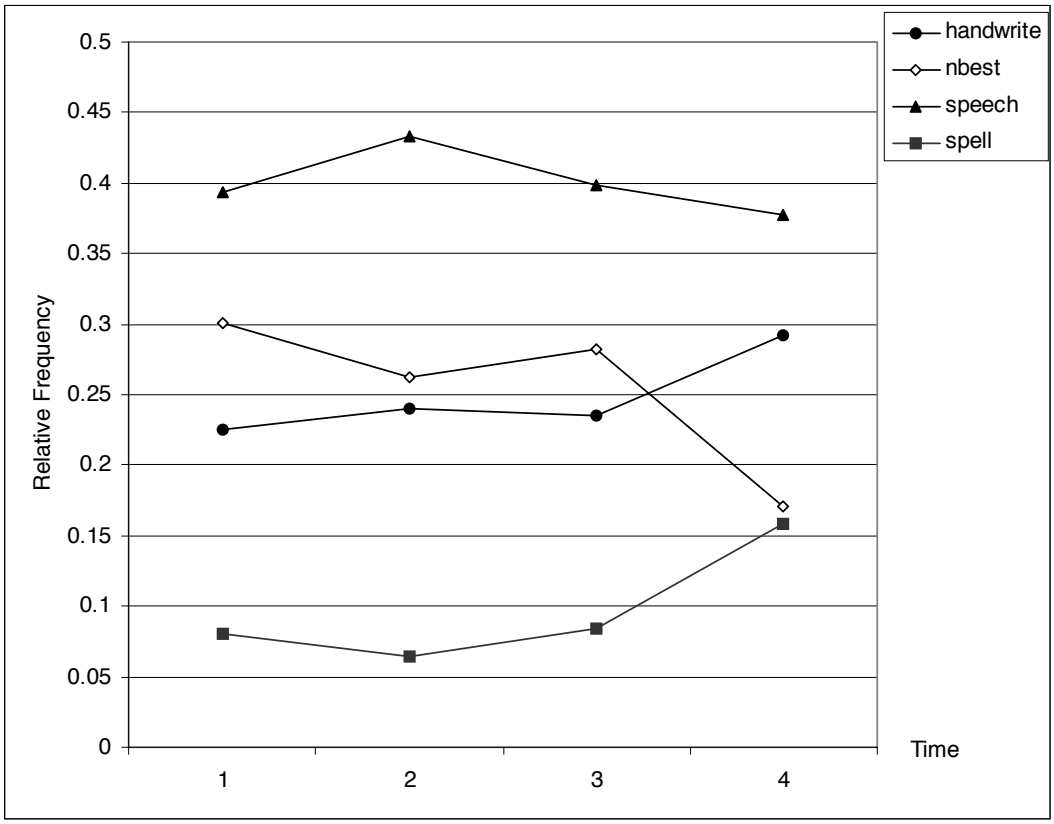


Figure 8-5. *Modality choice in the first correction attempt. Users slowly learn to prefer effective modalities (spelling and handwriting) over ineffective modalities (speech and choosing from the Nbest list.)*

How do users perceive multimodal correction? To assess this question, a post-experimental questionnaire was designed and completed by each participant. In summary, participants feel that multimodal correction is easy to learn, and judge the overall performance quite positively. However, multimodal correction does not score significantly better than correction using keyboard and list of alternatives ("Keyboard & List" experimental condition) in liking-scores along different dimensions. Finally, most participants prefer speech as correction modality if it was as accurate as other modalities. The results are presented in detail in the following list. Likert-scales of 1.0 - 5.0 were used, where 1.0 meant "good".

- Is multimodal correction easy to learn? Score: 2.0
- Overall performance of multimodal correction? Score: 2.3
- Is multimodal correction ... than conventional (keyboard&list) correction?

  a) more intuitive? - No ($p>0.1$)

  b) more pleasant? - Yes ($p<0.05$)

  c) takes less concentration? - No ($p>0.1$)

  d) causes less physical fatigue? - No ($p>0.3$)

  e) would prefer it for text editing? - No ($p>0.25$)

The fact that conventional "keyboard & list" correction was more efficient (faster) for most participants in the user study might explain why most participants did not perceive multimodal correction more positively in the direct comparison. Additionally, unsolved hardware and technical problems (e.g., unwieldy touchscreen) took away from the positive user experience with multimodal error correction. The following comments from participants on the post-experimental questionnaire support this explanation: "very difficult to get the pen to mark/gesture correctly, especially within words", "it gets tiring writing on the screen", "it was difficult to select from the list of alternatives". In fact, seven out of fifteen participants made suggestions to improve the hardware setup. This may explain why it didn't receive better preference scores than conventional "keyboard&list" correction.

Users are quite aware of how well different methods work for them. Table 28 below provides evidence for this point, by showing the correlation between actual correction accuracy (across different participants) and the subjective efficiency rating, which the participants assigned to the correction modalities in the questionnaire. Positive correlation between correction accuracy and self-reported efficiency of a correction modality means that the modality which users believe to be efficient, actually is efficient. As can be seen, users perceive the modalities of speech, spelling, and handwriting as efficient, when they are accurate - although a standard test for positive correlation reveals that the evidence presented here is not statistically signifi-

cant (on a 0.05 level - with the exception of choice-from-list).

Table 28: *Correlation between correction accuracy and self-reported efficiency of various correction modalities*

Correction Modality	Correlation
Respeaking	0.39
Handwriting	0.21
Spelling	0.29
Choice from List	0.73
Gesture	-0.2
Typing	-0.33

8.3.4 Usefulness of Multimodal Dictation System for Text Reproduction Tasks

Moving beyond the issue of error correction, this section discusses implications of this dissertation work on the overall text production process, answering the third research question - whether the multimodal dictation system is useful for text reproduction tasks. To assess the potential productivity gain of multimodal input methods, we compare the total system throughput (or *dictation speed*) of a *multimodal dictation system* (i.e., first dictating, then correcting multimodally) to that of a *conventional dictation system* (i.e., first dictating, then correcting using keyboard and choice from list) and a standard *text editor* (i.e., typing the entire text). Note that we use the term throughput differently from some commercial vendors of dictation systems who exclude the time necessary for correction.

The user study allows us to measure the total system throughput of the multimodal dictation system used in the experiment. However, that result depends on the specific recognizers available at the time of the experiment. To abstract from recognizer-specific issues, and to extrapolate to future scenarios (e.g., faster recognition at higher accuracy), this section applies the prediction method outlined in the previous chapter (see Section 7.2.3). Predictions confirm the hypothesis that the system throughput of a multimodal dictation system can exceed 40 wpm (assuming 90% dictation in real-time) and compares favorably to non-secretarial typing. More

details follow below.

Figure 8-6 illustrates the system throughput of a multimodal dictation system, a conventional dictation system, and a text editor. The throughput for the dictation systems assumes 90% recognition accuracy in real-time. Since typing speed has a significant impact on this comparison, the results are tabulated across different typing skills. Since the experiment did not really cover "slow" typists, results for the slow category are derived using predictions from the performance model. Furthermore, "poor" refers to novice users and "good" refers to experienced users for the multimodal dictation system

As can be seen, a multimodal dictation system compares favorably to fast (non-secretarial) typing of 40 wpm - without requiring any keyboard input. The dictation speed of conventional dictation systems varies significantly depending on the user's typing skill, and this is because they rely on keyboard input for correction. For users with good typing skills, a conventional dictation system is currently the most efficient text production method.

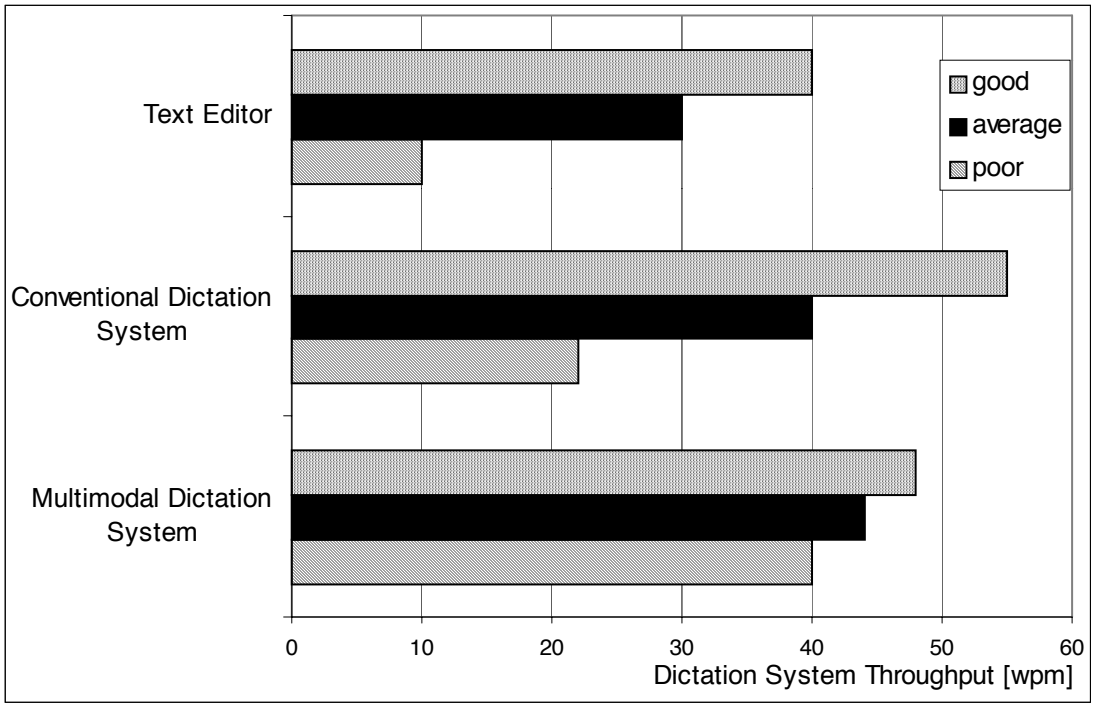


Figure 8-6. *Predicted throughput for different text production methods, across typing skills, for 90% dictation accuracy*

Table 29 provides more detailed information on this comparison. The first column shows the system throughputs $V_{Dictate}$ as measured in the experiment, with 75% accurate initial dictation at 2.6 x real-time. The second column contains the predictions on which Figure 8-6 is based. The last column indicates the accuracy of the text after completion of the task, i.e. how many errors were overlooked.

The measured system throughputs in column 1 are quite low, due to the comparatively low accuracy and slow recognition of initial dictation. Regarding the quality of the produced text document, although participants appear to have been slightly more careful in correcting multimodally, compared to the other conditions, the differences in text accuracy (shown in the last column) of the final document are not statistically significant.

Table 29: *Comparison of text input rates (including correction time) and text accuracy*

Text Production Method	measured $V_{Dictation}$ [wpm]	predicted $V_{Dictation}$ [wpm]	Text Accuracy [%]
Multimodal Dictation System (no keyboard input)	16	40	99
Conventional Dictation System (slow typist)	not covered by participants	22-34	97
Conventional Dictation System (average)	16-18	40-48	98
Conventional Dictation System (fast)	25	>52	98
Text Editor (slow typist)	5-15	n/a	not covered
Text Editor (average typist)	23-35	n/a	95
Text Editor (fast typist)	>40	n/a	97

8.3.5 Effectiveness of Partial-word Correction

This section analyzes the effectiveness of partial-word corrections, providing an answer to research question 4. Summarizing the discussion below, the research hypotheses regarding this question were confirmed in part and rejected in part. Our study confirms that partial-word corrections have higher input rates and correction accuracy, compared to whole-word correc-

tions. Overall, however, partial-word correction decreased correction speed. Due to problems with the available touchscreen hardware, selecting characters with pen-drawn gestures caused many difficulties, and any gain in input rate and correction accuracy was lost in additional gestures.

The pilot experiment compared multimodal correction with and without partial-word correction in two separate experimental conditions. Comparing the rows "Multimodal" and "Multimodal, no PWC" in Table 26 reveals the disappointing result that partial-word corrections decreased overall correction speed. However, the following more detailed analysis reveals that partial-word corrections had some positive effects.

Table 24 indicates correction accuracies and input rates V_{Input} for word-level corrections and partial-word corrections by spelling and handwriting. A standard t-test confirms our original hypothesis (see number (4) in 8.1) that partial-word corrections are significantly more accurate than word-level corrections ($p<0.05$), and that input rate for partial-word corrections is significantly higher ($p<0.05$). So why does partial-word correction decrease correction speed, if it increases correction input rate and correction accuracy? The explanation is that selecting and deleting at the level of characters is very difficult with the touchscreen used in the experiments, it is very thick and has a significant parallax. Therefore, users typically needed several attempts to selecting or deleting characters in a word, thus losing the saved input time in additional gesture interactions.

8.3.6 Effectiveness of Gestures/Pointing for Editing Tasks

This subsection reports results relevant to research question 5 - whether (pen-drawn) gestures and pointing are more efficient for editing than keyboard and mouse input. This question cannot be answered adequately, because of the limitations of the experimental design. Weak evidence can be obtained by comparing the overhead times in the experimental condition "Keyboard&List" to the "multimodal" conditions; the time spent on selecting is included in the overhead time. Table 30 shows that overhead times for conventional keyboard- and

mouse-oriented correction methods are lower than for multimodal correction, suggesting that gesture-based interaction using marks on a touchscreen is less efficient than traditional mouse-based interaction. However, this result may be confounded by the hardware set-up; with a full-sized 21-inch touchscreen and extremely large fonts to display words (as in the prototype multimodal dictation system), gestures to delete words and position the cursor require large hand movements. By contrast, the standard mouse works on a small mouse-pad and is quite efficient. The situation would be quite different with small, hand-held devices. Their display area is much smaller, making gesture-based interaction more efficient, but currently available pointing methods for such devices are rather inefficient.

Table 30: *Overhead times for keyboard/mouse-oriented correction versus multimodal correction*

	Keyboard & List	Multimodal
Respeaking	n/a	5.0
Spelling	n/a	4.0
Handwriting	n/a	3.5
Gestures	n/a	4.9
Choose from List	4.2	5.0
Typing	2.8	n/a
Mousing / Keying	3.8	n/a

8.3.7 Usefulness of Automatic Error Highlighting

The "Multimodal, with Imperfect Error-Locate" experimental condition allows us to address research question 6 - whether system-initiated location of errors speeds up error correction. Within the experimental design, this question can be answered only for the level of accuracy that the current implementation of automatic error highlighting achieved (87% classification accuracy). Comparing rows "Multimodal" and "Multimodal, with Imperfect Error-Locate", in Table 26, reveals that imperfect highlighting of likely recognition errors actually decreases correction speed ($p<0.05$). Being unable to trust in the error labels, participants scanned the entire text for errors anyway, and they became confused by incorrect error labels.

This result suggests that automatic highlighting of likely recognition errors must attain high accuracy levels to be useful. Since users currently spend a significant time locating and selecting recognition errors (as captured in the still fairly high overhead times), error correction should benefit from automatic locating of recognition errors at some level of accuracy.

8.4 Chapter Summary and Discussion

Leading back to the "bigger picture" from the detailed results presented in the previous section, this section summarizes the main results, discusses some high-level issues, and comments on some surprising results.

Multimodal error correction circumvents the drawback of many current non-conversational speech recognition applications that depend on keyboard input for efficient correction (research question 1). For applications where keyboard input is acceptable and for users with good typing skills, typing remains the most efficient correction method for text input (research question 2). However, with improvements discussed below, multimodal correction could eventually beat even fast typing in correction speed. A dictation system with multimodal correction can achieve text production rates of more than 40 wpm, which compares favorably to fast unskilled typing (research question 4). Our study did not provide conclusive evidence for the remaining research questions. Partial-word corrections (research question 4) could speed up correction even further, but hardware problems prevented a gain in overall correction speed, despite decreased input time and increased correction accuracy. Editing with gestures (research question 5) appears to be less efficient than keyboard and mouse input with current desktop hardware; however, editing with gestures remains an attractive alternative that can be expected to beat keyboard input in effectiveness for small or hand-held devices. Finally, the time spent by users locating recognition errors ought to decrease if the system were to automatically highlight recognition errors (research question 6); however, our study showed that the automatic classification algorithm must be highly accurate (more than 90%) in order to realize this gain.

The knowledgeable reader was probably surprised that the recognition rates on dictation, handwriting, and spelling input were much lower than what might be found in other publications. Table 31 provides evidence that explains the significantly lower recognition rates observed in the user study of the multimodal dictation system prototype. First, tuning all recognizers to real-time performance without speaker adaptation (unlike all current commercial dictation systems) deteriorated performance, compared to benchmark results. Second, reiterating a point made previously, correction input is more difficult to recognize than standard benchmark data. Standard benchmark data is evenly distributed across easy and difficult-to-recognize words, while repair is limited to the subset of difficult-to-recognize words. Third, with limited recognition performance across different recognizers, users need significant time to learn optimal correction strategies. The fact that the speed of multimodal correction increases from 4.5 cpm to 6.8 cpm for an expert user provides evidence for this point.

Table 31: *Explanation of performance losses, compared to benchmark results*

Modality	Benchmark, anytime	Benchmark, real-time	Dictation in User Study	First Correction	Repeated Correction
Continuous Speech	93%	83%	75%	42%	18%
Spelling		94%		81%	44%
Handwriting		93%		75%	44%
Gesture		93%		88%	86%

This table also suggests that there remains much room for improvement. Obviously, multimodal correction will benefit from improvements in any of its component technologies. For example, using predictions from the performance model, we can predict what improvements would cause multimodal correction to be faster than correction by typing for users with good typing skills:

- Achieving real-time performance in all modalities, and cutting all word error rates in half

- Reducing the overhead time per correction interaction from 3.5-5.0 seconds (as in the prototype) to 2.5 seconds. However, that may be more difficult to achieve than it appears; for example, imperfect highlighting of possible recognition errors actually increased overhead times, decreasing correction speed.

- Achieving a 70% accuracy for corrections by respeaking (across multiple correction attempts), and in real-time

The user preferences in multimodal correction, and the answers in the post-experimental questionnaire suggest that users preferred speech, if it was as accurate as other modalities. Why would speech be preferred, despite repeated evidence that speech works less accurately? These observations seem to confirm the claims of many advocates of speech recognition technology: that speech is more "natural" or "intuitive" as input modality, or to communicate in general.

The data on usage frequencies indicated that handwriting was chosen as correction modality much more frequently than spelling. Why would handwriting be preferred over spelling? One reason is that spelling long words is difficult. Why? Basic facts on human cognitive processing suggest the following explanation. For spelling, the letter sequence and the "index" of how far one has progressed in spelling a word must be kept in short-term memory. With words longer than 5-7 letters, this information exceeds the capacity of short-term memory. For long words, information must be retrieved from long-term memory during spelling, and this increases cognitive stress. By contrast, for handwriting, we can rely on visual feedback: seeing what is written so far eliminates the need to keep an index in short term memory, and prevents decay of the letter sequence. Therefore, although it is slower, correction by handwriting may actually be preferred to correction by spelling.

Comments made by participants on the post-experimental questionnaire suggest overall that, despite performance and hardware problems with current technology, they felt that multimodal correction is attractive, and if limitations due to current recognition performance and hard-

ware were eliminated, multimodal correction would be faster, even for users with good typing skills. The following quotes by participants on what they felt was best about multimodal correction support this point: "it is easier to correct minor mistakes without typing - similarly, the handwriting recognition makes mistake correction much easier", "very easy, not as frustrating as writing, should become even better as recognition improves. Does not distract from task of dictating as much as pop-ups", "You have a choice of methods, each of which is more useful for different situations", "fast, accurate, and little concentration needed", "When it didn't understand my speech multiple times, I could just write the word", "the spelling is fantastic".

9. Conclusions

This final chapter summarizes the dissertation and its main contributions. Section 9.3 discusses limitations of the interactive approach to error correction. Section 9.4 outlines directions for future research - first obvious extensions of this dissertation work, followed by broader issues that would lead to new research projects. The chapter closes with some concluding remarks.

9.1 Thesis Summary

This dissertation investigated interactive error correction for speech user interfaces. It is widely acknowledged that speech input is preferred over other input methods, such as keyboard and mouse, in applications where hands or eyes are busy, where mobility is required, and where speech is faster or more convenient. In applications which keep hands or eyes busy (e.g., inspections or car navigation), non-speech modalities are not available for error correction, and applications which favor interaction as a dialog with the user (e.g., translation or automated directory assistance) suggest the use of conversational error-correction techniques. This dissertation focussed on *non-conversational* speech recognition applications with *graphic user interfaces*, for example mobile personal assistants, form filling, and dictation applications.

An informal survey of commercial speech recognition products and published research systems revealed that error correction is limited to variations of the following three methods: respeaking (repeating using continuous speech), choosing from a list of alternatives, and using mouse and keyboard. With current speech recognition technology, *respeaking* is ineffective because most speech recognizers often deteriorate on repetitions, especially when hyperartic-

ulated. For two state-of-the-art large vocabulary recognizers, the word error rate on corrections by respeaking was more than double the word error rate on the original input. *Choosing from a list of alternatives* is not effective in continuous speech recognition applications because the correct sequence of words is frequently not among the top alternative choices, especially if several consecutive words are misrecognized. For the recognizer used in this dissertation work, the correct word was among the 6 next best alternatives only in about 25% of the cases. Finally, *using mouse and keyboard* defeats the purpose of employing speech as an alternative to keyboard as input modality. Hence, current correction methods for non-conversational speech recognition applications are either ineffective or require keyboard input.

This dissertation proposed *interactive multimodal correction* as an alternative. The user can efficiently correct errors by switching modalities for repetitions (for example, from continuous speech to isolated words, handwriting, or oral spelling), and by using intuitive pen-drawn gestures to delete words and to position the cursor. Interactive multimodal correction is effective because the user provides a signal which is redundant with the original speech input. For example, with the recognizers available in this research (without modifications to handle isolated or hyperarticulated speech), multiple word corrections by speech are only 50% accurate, whereas multimodal corrections by spelling or handwriting are 80-90% effective on the first attempt.

Achieving such high correction accuracies was challenging because recognizing corrections is difficult. Short words are hard to recognize in any modality, and continuous speech recognizers frequently misrecognize them. Similarly, the accuracy of recognizing spelled or handwritten corrections is lower on short words. To achieve 80-90% correction accuracy, three algorithms were developed in this thesis and integrated with the overall system. These algorithms improve the effectiveness of interactive multimodal correction by *correlating correction input with the context*.

- First, phrase and sentence level context are utilized by constraining the search space for correction input with a language model. Using the trigram word context of an error to rescore lists of alternative hypotheses obtained from decoding isolated word repair input decreases the error rate for corrections by up to 26% (relative).
- Second, the interpretation of correction input can be biased towards words which are frequently misrecognized by the continuous speech recognizer. For instance, applying a unigram prior to the interpretation of isolated word correction reduces the error rate by up to 20%.
- Finally, methods which correct characters within a word were also developed. Such *partial-word corrections* increase speed of correction by increasing input speed and correction accuracy. If the recognition vocabulary is limited to those sequences of characters that could follow the partial word, the error rate on corrections decreases by 50%, compared with word-level corrections.

Input speed is the most important quantitative evaluation measure for input modalities. To compare various multimodal correction methods, and to extrapolate results to future recognition performance, this dissertation presented a performance model of multimodal human-computer interaction. The model predicts that correction by respeaking would be faster than typing if recognition accuracy on spoken corrections exceeds 70%[1], and that full word handwriting will always be slower than typing, even if 100% accurate recognition were available.

To empirically evaluate interactive multimodal correction, state-of-the-art large vocabulary recognizers for continuous speech, spelling, handwriting, and pen-drawn gestures were used to build a prototype *multimodal dictation system*. In user studies, two categories of correction methods were compared: methods available in current dictation systems (respeaking, choosing from a list, or typing), and the new multimodal methods. The results show that interactive

1. Recognition rates of current continuous speech recognizers on spoken corrections are closer to 50% - without modifications to handle isolated word input and hyperarticulated speech.

multimodal correction almost doubles correction speed, compared with respeaking and choosing from a list of alternatives. Typing currently remains the fastest correction method for users with good typing skills. For applications where a keyboard is not desirable or not possible, or for users with poor typing skills[1], state-of-the-art continuous speech recognizers with multimodal correction achieve text input rates of 40-50 words per minute (including the time necessary to correct errors), which compares favorably to fast non-secretarial typing. Analyses of the modality usage patterns when users were given free choice among correction methods showed that user choice between modalities is driven by correction accuracy, although there is a bias towards using speech initially. Users learn to prefer the more accurate correction modalities and avoid inaccurate ones.

In summary, multimodal error correction bridges the gap between the promise of speech recognition technology to deliver fast input without keyboard, and the problem of losing the potential productivity gain in correction of recognition errors.

9.2 Contributions

This dissertation contributes to the speech recognition field by developing multimodal interactive techniques for correcting speech recognition errors without using a keyboard, in non-conversational speech recognition applications with a graphic user interface. Previously, although speech recognition technology offered an alternative to the keyboard as input modality, the time gained by using speech as fast input modality was largely lost in correcting recognition errors. With multimodal error correction, and with algorithms that increase correction accuracy by correlating repair input with context information (as proposed in this dissertation), text input without keyboard input can be more efficient than text input by typing. This advantage will be even more pronounced with further improvements in recognition technology and interface hardware.

1. Typing speeds of less than 20 wpm can be considered poor, more than 40 wpm are generally considered fast unskilled typing, and more than 50 wpm fast secretarial typing.

A second contribution lies in breaking the focus on word accuracy as single performance measure, which is still widespread in the speech recognition field. Applying the methodologies of model-based and empirical evaluation to speech user interfaces, the evaluation of multimodal correction addresses important task-level and qualitative issues: that not only dictation accuracy, but also efficiency of error correction is crucial to realize productivity gains with speech recognition applications, and that user choice between modalities is driven by recognition accuracy.

Third, this dissertation contributes to the speech recognition and human computer interaction field by empirically confirming the hypothesis (proposed by other researchers in both fields) that multimodal flexibility can improve error correction in speech recognition applications. Multimodal correction is generally more efficient than conventional keyboard-free correction (by respeaking and choosing from lists of alternatives), but less efficient than correction with skilled keyboard input. With further improvements in recognition accuracy, implementation of multimodal correction, and hardware, multimodal correction will eventually compare favorably even to correction by fast typing.

Fourth, the performance model of recognition-based human-computer interaction presented is useful beyond its application to the problem of error correction. As the speech recognition field matures, performance predictions that abstract from current recognition performance and interface implementation will be extremely useful in the realization of more and more speech recognition applications. The model could be a first step towards a formal framework for multimodal interaction.

Finally, this dissertation also suggests which factors determine user preference in speech-enabled interfaces: both naturalness of the medium (users generally prefer speech) and accuracy of interpretation. The developer of speech recognition applications can thus circumvent limitations of speech by offering other, more accurate modalities to compensate for speech recognition errors.

9.3 Limits of Interactive Multimodal Error Correction

As pointed out in the introductory chapter, interactive multimodal error correction as presented in this dissertation does not try to solve the problem of error correction in speech user interfaces in general. What follows is a discussion of the most obvious limitations of interactive multimodal correction.

First of all, since handwriting and gestures (editing marks drawn on the screen) were explored as main alternative modalities to speech, interactive multimodal correction is applicable only to speech recognition applications with graphic user interfaces. However, the more general concept of offering alternative input modalities obviously applies to any kind of speech recognition application, because alternatives include variations of one modality, for example speech as continuous speech, disconnected speech, and isolated words. Furthermore, many conversational applications also contain subdialogues for which multimodal correction is appropriate, hence multimodal correction is not limited to only non-conversational applications. Overall, the benefits of multimodal correction are highest for tasks that favor multimodal interaction (e.g., handwriting for numerical data entry, or gestures for manipulation of graphical data), when performance on spoken corrections is particularly poor (e.g., in noisy environments), or when keyboard input is not available or not efficient (e.g., on small hand-held devices).

Second, whether multimodal correction is faster than unimodal correction obviously depends on how poorly unimodal correction works. This dissertation showed that on a dictation task, multimodal correction is faster than unimodal correction by respeaking - assuming the same speech recognizer for both initial and correction input. Other work [Soltau and Waibel 1998] suggests that recognition accuracy on spoken corrections can be significantly increased by modifying the recognition algorithm appropriately. Performance model predictions can help to decide whether multimodal correction is beneficial or not. For example, if speech corrections were more than 60% accurate (across multiple correction attempts), they would outperform multimodal correction as implemented in the multimodal dictation system prototype.

Interactive multimodal correction as presented in this dissertation does not solve the problem of recognition errors due to new words. The new-word problem has not received sufficient attention in the speech recognition field in recent years, because the ability to handle larger and larger vocabularies appeared to reduce the problem considerably; with a 60,000 word vocabulary, the rate of new words on Wall Street Journal texts is less than 1%. However, in more realistic application settings the new-word problem remains significant. For example, internal tests of a speech recognition software developer suggest that the new-word rate for not matching texts is as high as 4% [Acero 1998]. Also, new words tend to contain important content; for example, new words are frequently names [Suhm 1993]. Deciding whether a recognition error is caused by a new word, and graceful and efficient ways to incorporate new words into an existing system remain a challenge for future research.

Finally, this dissertation evaluated text *reproduction* only, and most dictation involves text *composition*. Some research suggests that for text composition, not input rate, but composition skill is the main limiting factor [Gould 1978], and that for highly specialized professionals, dictation software is only acceptable if recognition accuracy is almost perfect [Lai and Vergo 1997]. However, speech-enabled dictation and text editing tools could still be attractive for highly specialized professionals, for example, by delegating the effort required to correct recognition errors to less skilled assistants.

9.4 Future Research

9.4.1 Extensions of Multimodal Interactive Correction

The results from the user study of the multimodal dictation system prototype suggest a number of improvements and extensions that would increase its usefulness as data entry and text editing tool. What follows is a discussion of some obvious next steps.

There are several measures to improve system throughput of the multimodal dictation system. First, the performance of the dictation recognizer was 75% accuracy in 2-3 x real-time, instead of more than 90% accuracy in real-time claimed by current commercial dictation sys-

tems. To achieve this performance, commercial systems adapt the recognizer to the current user's voice. Second, since users seem to prefer speech corrections and adopt a more staccato-like speaking style in corrections by respeaking, a disconnected speech recognizer could be used for spoken corrections. Disconnected speech recognition is easier than continuous speech recognition, therefore accuracy of corrections by respeaking in disconnected speech mode should exceed 80%. Third, how different input and correction modalities are triggered could be improved. An obvious limitation of the prototype was the use two separate buttons to distinguish between continuous speech and spelling input. If automatic classification of various speech modalities was sufficiently accurate, interaction time could be reduced by having the system listen continuously for speech input, and process speech input appropriately as soon as the users speaks, without having to press a button.

More functionality is necessary to cover all tasks that typically occur during text editing. Currently, the only supported editing tasks are deleting words and placing the insertion cursor. Text editing involves many more editing tasks, including formatting, moving items, and importing tables and graphics. Also, the prototype supports text input only; a system usable for document production would need to support all kinds of input, including punctuation and digits. Basically, the functionality of a typical word processor would need to be covered.

Interactive multimodal correction should be explored in other data entry tasks. For example, the input speeds of different modalities are different for digit input, or input including graphic information. For those data entry tasks, handwriting and gestures may be faster than speech input.

Finally, longitudinal studies of modality usage patterns are needed to determine whether the observed bias towards using speech diminishes with increasing practice, or whether it persists. Within the experiment, this bias could be explained as a novelty effect, because most participants had no prior experience with speech recognition technology.

9.4.2 New Research Directions

Beyond improving interactive multimodal correction, and developing the multimodal dictation system into a tool useful for document production, this dissertation suggests several directions for substantial future research: recognition algorithms that can handle hyperarticulated speech better, interactive correction for conversational speech recognition applications, interactive correction using several modalities simultaneously, and a framework for multimodal human-computer interaction.

Other research confirms that hyperarticulation of speech corrections is one reason for poor performance of (unimodal) correction by respeaking [Oviatt, Levow et al. 1996; Soltau and Waibel 1998]. Users appear to transfer hyperarticulation as an effective strategy to resolve communication problems in human-human conversation to error resolution in speech-enabled interfaces. In contrast to hyperarticulation in human-human conversation, hyperarticulated speech corrections are even more difficult to recognize by a machine - at least with current speech recognition algorithms that are trained for normally pronounced speech. To account for people's natural tendency to hyperarticulate speech corrections, speech recognition algorithms that are good at recognizing hyperarticulated speech must be developed. After all, there is the potential to do better on hyperarticulated speech, compared to normally pronounced speech, since at least the human hearing system is able to exploit the additional information in hyperarticulated speech. Initial research on modifying the recognition algorithm for isolated word input and hyperarticulated speech shows that accuracy on spoken corrections can be significantly increased[1].

Second, for many interesting speech recognition applications, conversational interactions are preferable (see also the taxonomy of speech recognition applications presented in Chapter 1). Research in dialogue systems addresses conversational error correction in part; some systems allow the user to correct recognition errors in the context of a spoken dialogue (so-called *clar-*

1. On a database of German hyperarticulated speech, modifications of the recognition algorithm increased the word accuracy on isolated word input to around 80% [Soltau and Waibel 1998].

ification dialogues). However, a framework for development and evaluation of error correction in such applications is still missing.

Third, this dissertation did not explore two approaches to interactive correction that could potentially increase correction accuracy and efficiency further: spoken correction commands (as already implemented in some commercial dictation systems, such as Dragon's Naturally Speaking), and simultaneous or combined use of several modalities for correction.

Finally, the performance model of multimodal, recognition-based interaction presented in this dissertation is only a first step towards formalizing multimodal interaction. Future research could generalize the model to cover interaction that does not occur sequentially (e.g., concurrent multimodal input), and interaction that is not necessarily repetition of previous input. Furthermore, current research on multimodal interfaces is limited to case-studies that show benefits of multimodal interaction in a certain context. A framework of multimodal interaction would need to characterize situations where multimodal interaction helps, based on some clearly defined criteria.

9.5 Final Remarks

Important speech recognition applications that include data-entry tasks and afford a graphic user interface include text composition, service transaction systems, and information retrieval. For such applications, this dissertation shows how keyboard-free input using speech and other modalities can be more efficient than traditional keyboard and mouse input. This dissertation thus contributes a piece to the puzzle of where human-computer interaction can benefit from multimodal interaction, and how to go about implementing such multimodal applications. Exploring the differential strengths and weaknesses of different input modalities remains a research challenge, both for the fields that develop the necessary component technologies (including the speech recognition field), and the human-computer interaction field. Both of these fields have gone mostly separate ways during the past decade. Early attempts of HCI researchers to use speech as alternative input modality were disappointed by low speech rec-

ognition performance at the time, and the speech recognition field has traditionally focused on improving recognition performance. Now, as the technology matures, both fields will benefit from pooling their expertise to make speech recognition a success in enhancing human productivity. As Newell said: "there is much more to designing a useful speech-driven word processor than simply attaching a speech recognition front end to a standard word processor. For such a system to be acceptable, the performance requirements are very high" ([Newell, Arnott et al. 1991], p. 131). This dissertation provided further evidence that performance means more than just speech recognition accuracy.

Appendix A: Experiment Materials

This appendix contains all the written materials that were used during the experimental evaluation of multimodal error correction: the consent form that was signed by each participant (to ensure participation was voluntary, and to reaffirm to the participant that the experiment had been reviewed by the standard human subject review process of Carnegie Mellon University), the experiment instructions that were handed out to each participant at the beginning, the quick tutorial of the multimodal listening typewriter (which can serve as user manual for future reference), the various dictation tasks that were used during the tutorial phase (the trial tasks) and during the experiment (one for each of the three experiment sessions), and finally, the post questionnaire that each participant filled out upon completion of the experiment. The experiment instructions are inspired by [Gomoll 1990].

A.1 Participant Consent Form

Project Title: Interactive Multimodal Error Correction

Conducted by: Bernhard Suhm, Computer Science Department

I agree to participate in experimental research conducted by students or staff under the supervision of Dr. Alex Waibel. I understand that all experiments have been reviewed by the University's Institutional Review Board and that to the best of their ability they have determined that the experiments involve no invasion of my rights of privacy, nor do they incorporate any procedure or requirements which may be found morally or ethically objectionable. I understand that my participation is voluntary and that if at any time I wish to terminate my participation in this study I have the right to do so without penalty. I understand I will be paid for my participation when I have completed the experiment. I further have the right to contact the following people and report objects of any kind, either orally or in writing to:

Susan Burkett - Associate Provost - Carnegie Mellon University

(412) 268-8746

Alex Waibel - Principal Investigator - Carnegie Mellon University

(412) 268-7676

Purpose of the Experiment: This research involves the evaluation of interactive methods to correct recognition errors. I understand that I will carry out certain tasks which include dictating and correcting using speech, spelling, handwriting, pen-drawn gestures and typing. Additionally, I will answer questions and fill out a questionnaire concerning my opinion about this computer program. I also understand that some sessions may be videotaped for later analysis by the researchers.

Anonymity of Participation: I understand that all participants will be given code names, and that videotapes and transcripts will be labelled with theses code names only. The videotapes will be shown only to research directly working on this project. However, if the need arises to show any portion of the videotapes of my participation, I understand that the researchers will obtain my written consent before doing so.

_____ _____

Name (Please Print) Signature and Date

_____

E-Mail (optional)

A.2 Experiment Instructions for Participants

The Nature of the Experiment

- You are helping me by trying out an advanced speech recognition interface.
- I am testing the system; I am not testing you.
- This is a prototypical system. If you have trouble with some of the tasks, it is the system's fault, not yours.
- Remember, participation is voluntary. If you should become uncomfortable or take any objection, feel free to quit any time.

Experiment Phases

- *1. Phase: Tutorial*

  You learn how to use the speech recognition interface, and how to correct errors using different correction methods. At the beginning of the tutorial, the experimenter demonstrates the different features of the interface to you. Then, you get familiar with the different correction methods in a series of simple practice tasks. Finally, to master each method, you have the opportunity to practice more guided by a series of drill tasks.
- *2. Phase: Experiment*

  1. Step: To measure your typing speed, you type in several sentences from the Wall Street Journal.

  2. Step: You dictate three sets of sentences from the Wall Street Journal and correct the errors which occured using different sets of correction methods, as instructed by the experimenter: "Keyboard/Mouse" and "Multimodal" (explanation see below), "Keyboard/Mouse with Autohighlight".
- *3. Phase: Debriefing*

  You will be asked to fill out an evaluation questionnaire. Your critical comments will be helpful in improving the system.

Correction Methods:

- *Correction Method A*: You can replace words by typing or consulting the list of alternatives (click on word and hold until pop-up menu appears, or a message in the status line stating there is currently no alternative available for the word you selected). You can insert words by typing, and delete words by selecting them, and hitting the backspace or delete key. Place the cursor by clicking between two words (similar to a standard text editor).
- *Correction Method B*: You correct errors choosing among: respeaking, spelling, handwriting and partial word correction. Delete words and place the cursor using pen-drawn gestures.
- *Correction Method C*: The difference from method B is that the system automatically highlights words (in red) which may be incorrect. However, this flagging of errors is not 100% reliable: the system will fail to flag some errors, and also mistakenly flag correct words. Therefore you have to double check to ensure you do not miss any misrecognized words. You can use the "Select Next Error" button to select the next consecutive sequence of red words following the currently highlighted word or position of the cursor.

Special Instructions:

- All words are displayed in lowercase - do not worry to get upper/lowercase correct.
- *Punctuation:* Although the prompts are punctuated (mainly to help reading), punctuation is currently not supported by the multimodal listening typewriter.
- *Abbreviations:* Slightly inconsistent across modalitites. In continuous speech, they will appear as separate words (e.g., f. b. i.), in spelling as one word, all lowercase (fbi), and in handwriting as one word (f.b.i.)
- *Numbers*: In spelling and handwriting, you have to write numbers out, for instance spell "f-i-v-e" for "5".

- *Contractions*: Supported in all modalities, but in spelling, just ignore them. For instance, spell "d-o-n-t" to input "don't".

A.3 Interactive Multimodal Correction Quick Tutorial

The multimodal listening typewriter combines large vocabulary continuous speech recognition with interactive multimodal correction. You can switch between providing new input and correcting/editing at any time. The supported modalities are: continuous speech, spelling, handwriting, typing and editing gestures (drawn on the display). You can use any of the supported modalities for both new input and for corrections. However continuous speech is currently the only modality which allows multiple word input.

How to provide input in different modalities:

Table 32. *How to initiate input in different modalities.*

Modality	To initiate input	Input termination criterion
Continuous Speech	Press *Dictate/Respeak* button, listen for beep, start speaking	Stop speaking - recording will stop after ~1 sec. of silence
Spelling	Press Spell button, listen for beep, start spelling (just one word)	as above
Handwriting	Start writing on the display - avoid writing over displayed words, and resting the wrist on display	Stop writing - processing will begin after ~1 sec. of no further pen input
Editing Gestures	Mark gesture on the display - avoid resting the wrist on display	as above
Typing	Start typing	Press *Spacebar* or *Return*, or initiate input in another modality

Basic Correction Methods

- *Replacing Words:* Select one or more words, and replace the selection by providing input in any modality. In addition, you can replace one word by choosing from the list of alternatives (see below).
- *Inserting Words:* Place the cursor between the words where you want to insert, and then provide input in any modality.

- *Deleting Words:* Use one of the supported deletion gestures to delete one or more words. You may delete several subsequent sequences of words with a single gesture.

How to select words

Tap with your finger on a word to select that word. If this word is next to another, already selected word, the word will be added to the selection; otherwise the previous selection will be deleted, and only the word that you tapped on will be selected.

Potential Problem: You tap a word, and it is deleted. – Reason: You may have "smudged" the finger a bit while tapping, and the interface interpreted it as editing gesture.

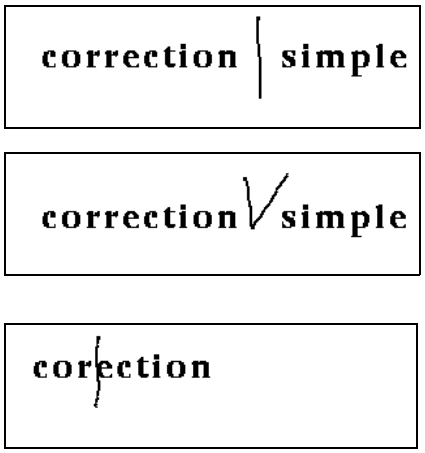
How to deselect

- Tap on the screen "far" (2-3'') away from any displayed word OR
- Place the cursor somewhere, using one of the place-cursor gestures OR
- Use the *Undo* button

How to choose from the list of alternative hypotheses

This correction method is supported only for replacing isolated words. Tap on the word and hold for ~1 sec. If there is no alternative for this word, a message will appear in the status line. However if there are alternatives available, they will be displayed as pull-down menu. To select any of the alternatives, move your finger over the alternative you want (do not release your finger in the meantime) until the chosen alternative appears highlighted, then release your finger. If none of the alternatives is correct, release your finger over any other part of the display.

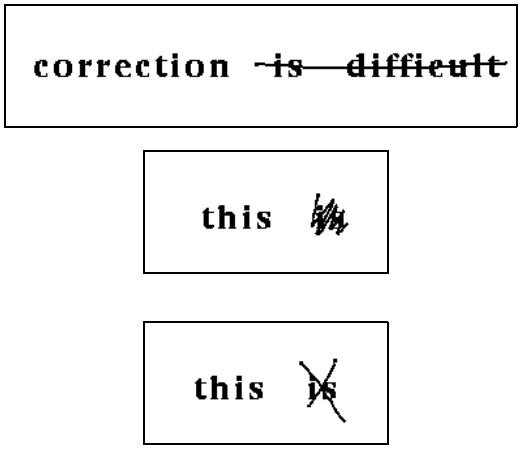
Gestures to Place the Cursor



Potential Problems:

- You mark one of these above gestures between two words, however nothing happens. – The interface probably failed to recognize your gesture. Try again, or use a different gesture.
- A word is deleted instead. – Probably your mark covered a word partly, and the interface misinterpreted it as a deletion gesture.

Gestures to Delete Words



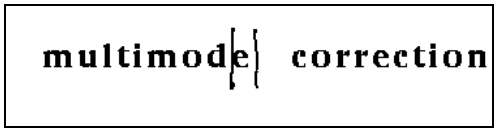
Potential Problems:

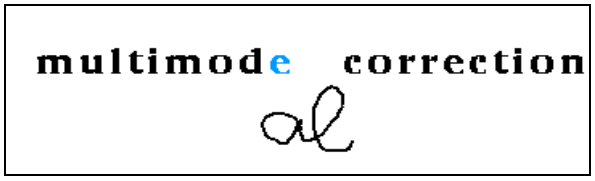
- You use one of these gestures, however nothing happens. – The interface probably failed to recognize your gesture. Try again, or use a different gesture.
- More words are deleted than I intended. – If your gesture extends across multiple words, the interface will delete all words which are (partially) covered. Try again, and be careful not to cover any word that you do not want to delete.
- Only part of the word is deleted. – You have to cover the whole word, otherwise the interface will interpret your gesture to delete only the characters you covered (see below under "partial word corrections"). As mentioned above, you do not need to cover the whole word if you delete multiple words.

Partial Word Corrections

Beside corrections on the level of whole words, you can also correct on the level of characters within a word. This may be beneficial if a recognition error consists only of a few added (or substituted) characters. Instead of replacing the whole word, you can select the characters you want to replace (demarcating the first and last character with a gesture consisting of two vertical bars as shown below) or delete them (using the same gestures as for deletions of whole words). Then, you can correct by providing only the missing characters. Such partial word corrections are supported for spelling and handwriting.

Gesture to select characters, and partial word correction by handwriting:





Potential Problems:

- The characters I intend to select are not selected, instead, they are deleted, or nothing happens at all. – With many current touchscreens, due to the thickness of the screen it is difficult to select characters. The gesture to select is sometimes misinterpreted as gesture to delete, or as handwriting input. Please try again - otherwise resort to correction on the word level.
- The correction would result in a valid English word, but the system keeps misrecognizing it. – Maybe the word is not in the system's vocabulary. Check by pressing the *Add Word* button, and type the word. A message will appear in the status line, stating whether the word was already in the vocabulary, or whether it was out-of-vocabulary and has been added to the vocabulary.

Autohighlighting of Errors

- In this mode, the system will highlight words in red which probably have been misrecognized.
- Use the *Select Next Error* button to select the next sequence of highlighted (red) words, in preparation of replacing these words in the following repair interaction.
- Note that the autohighlighting will mistakenly highlight some correct words, and also mistakenly fail to highlight some errors. Therefore double check all words.

How to correct using mouse and keyboard

- *Select a word* by clicking on it
- *Place the cursor* by clicking between two words
- *Delete word(s)* by selecting and pressing *Delete* or *Backspace*

A.4 Experiment Tasks

Trial Experiment Tasks

Digital technology has eliminated the difference and something new is emerging

Apple computers next generation of machines may look like a Mac, beep like a Mac and greet you with a smiling icon but don't be fooled

These Macs will act as telephone answering machine television and V. C. R.

Broadcasters and producers filled a large auditorium to hear Clinton's presentation

Since nineteen eighty two (1982) Lotus has grown into a publicly traded company that has nearly one billion dollars in annual revenues

Dictation Task #1

Insurance and risk management professionals say the terrorist bombings in Sacramento and Oklahoma city did not surprise them

Such attacks around the globe have been rising for years they said

But after Oklahoma city it's a safe bet that executives everywhere realize they are not immune from the threat of terrorists

The global market for environmental goods and services will surge in the next two decades

The report was prepared for businessmen at a seminar being held as part of a global forum conference a follow up to the nineteen ninety two (1992) u. n. conference on environment and development

The Supreme Court Monday declared constitutional California's worldwide unitary method of taxation deciding a long legal battle

The nation's court by a seven to two vote rejected a challenge against the controversial taxing method

But the case still was important because a number of mainly foreign based firms still sought tax refunds

Why do health insurers businesses not reimburse health care providers at Medicare rates rather than suffer the effect of government cost shifting

Several employer groups testified against the administration plan saying it isn't needed

Others said faster funding rules could put many firms in financial jeopardy

As automakers position themselves for California's nineteen ninety eight (1998) deadline for zero emission cars american Flywheel Systems Incorporated announces a deal

Dictation Task #2

Unfortunately we live in a society where terrorism is a part of our lives and our futures

Companies historically have been very lax when it comes to security, said a spokesman for the Risk and Insurance Management society

Pension and profit sharing spending rose only three percent (3 %) in nineteen ninety two (1992) from nineteen ninety one (1991) as outlays for health care increased ten percent (10 %)

Employers effectively have been robbing pension funds to help pay for climbing health care costs

More than two thirds of these employers also contributed to the cost of telecommuting

The statistics were compiled by the Employee Benefit research institute a study group

They would rather wait for the manager to put together a track record that can be compared to

those of other funds

Some of the new arrivals have performed well

He found that fifty one percent (51 %) of the aggressive growth funds that have been appeared since nineteen seventy two (1972) trailed their peers during their first year of operation

When O. J. Simpson brutally beat his wife five years ago and got off with only a slap on the wrist it wasn't because he was a celebrity

Video would be shot and the production house would make the ad

Dictation Task #3

The favorite targets of terrorists have long been oil companies financial institutions, utilities, and nuclear power plants

Usually we only hear about the large scale episodes, Harris said

The trend is definitely upward for sabotage and terrorism, said Steven Harris vice president of a San Francisco based company

Barry Diller chairman of Q. V. C. Network Incorporated and Laurence Tisch chairman of C. B. S. are negotiating a deal in which C. B. S. would merge with Q. V. C.

A nineteen ninety two (1992) Roper organization poll showed fifty eight percent (58 %) of respondents recording programs when they were unable to watch T. V.

Q. V. C. which earned fifty nine (59) million on revenue of one point two (1.2) billion in the most recent fiscal year operates two home shopping channels that sell consumer products

New mutual funds are being formed every day, but does it make sense to put money in a fund without a track record?

Investors will have to wait four weeks to bid for U. S. treasury coupon securities again

I have an office full of examples where Medicare payments to hospitals are only seventeen percent (17 %) to eighteen percent (18 %) of charges

Richard Darman former director of the federal office of management and budget and now an investment banker with the Washington D. C. based Carlyle group will join a board of advisers

A.5 Participant Post-Evaluation Questionnaire

1. How easy were the correction modalities to learn?

difficult 1 2 3 4 5 easy

2. How well did the different correction modalities work for you?

Respeaking	poor	1	2	3	4	5	good
Spelling	poor	1	2	3	4	5	good
Handwriting	poor	1	2	3	4	5	good
Editing Gestures	poor	1	2	3	4	5	good
Partial Word Corrections	poor	1	2	3	4	5	good
Keyboard and Mouse	poor	1	2	3	4	5	good
Choosing from Alternatives	poor	1	2	3	4	5	good

3. Assuming all modalities had equal accuracy, what method(s) would you prefer, and why?

4. Indicate your opinion on the following statements with respect to correction method A:

"Correction is intuitive"	agree	1	2	3	4	5	disagree
"Correction is pleasant"	agree	1	2	3	4	5	disagree
"takes little concentration"	agree	1	2	3	4	5	disagree
"causes little physical fatigue"	agree	1	2	3	4	5	disagree
"would use it for text editing"	agree	1	2	3	4	5	disagree

5. Indicate your opinion with respect to correction method B (choosing from a list of alternatives, respeaking, spelling or handwriting):

"Correction is intuitive"	agree	1	2	3	4	5	disagree
"Correction is pleasant"	agree	1	2	3	4	5	disagree
"takes little concentration"	agree	1	2	3	4	5	disagree
"causes little physical fatigue"	agree	1	2	3	4	5	disagree
"would use it for text editing"	agree	1	2	3	4	5	disagree

6. How do you judge the automatic highlighting of errors in correction method C?

7. What are the best aspects of correction method B?

8. What are the worst aspects of correction method B? What were the most common mistakes the system made?

9. How would you rate the overall performance of correction method B?

poor 1 2 3 4 5 good

10. What would you suggest to improve the system?

Demographic Information

Age:

Sex:

Education (completed):

How do you rate your typing skills:

poor 1 2 3 4 5 good

Do you have prior experience with speech recognition applications? If so, what kind?

Any other comments:

Appendix B : Theory of Repair in Human-Human Dialogue

Communication problems and the strategies people use to resolve them have been studied extensively both in linguistics and in medicine. The theory of repair in human-human dialogue provides useful analogies for the investigation of repair in human-machine dialogue for two reasons: to serve as a measure for what kinds of repair can be considered "intuitive" and "natural", and to help build theoretical underpinnings for the design of repair in human-machine dialogue.

Conversation analysis, a subfield of linguistics, investigates how people communicate. Research in conversation analysis forms the basis of the theory of repair in human-human dialogue. The first section B.1 reviews research on the structure of natural language dialogue. The goal of natural language dialogue is to extend the shared knowledge (common ground) of the conversation partners. This goal is achieved collaboratively in a sequence of conversation turns. Linguistic conventions determine when a dialogue partner has ended a turn and when dialogue control is passed to someone else.

The following section B.2 looks at what kinds of communication problems occur in human-human dialogue. Several taxonomies of errors in human-human dialogue are reviewed. Communication problems can occur on different linguistic levels (lexical, syntactical, semantic) and at different stages in the communication process (from vocalization of the sequence of words to an understanding of the intended meaning).

Just as in natural language dialogue in general, repair of communication problems in such dia-

logues is structured by turns. Section B.3 reviews the *repair strategies* used in human-human dialogue. Conversational repair can be categorized into self- and other- repair according to who initiates and who executes the repair (the speaker versus the listener). Repair strategies include repetitions, paraphrases, and additions. Repetitions are preferred over more complex repair strategies unless multiple interactions force the conversation partners to employ the more complex strategies, such as paraphrases or additions.

B.1The Structure of Human-Human Dialogue

The foundation for the theory of *conversational repair* is the research in conversation analysis that investigates the structure of human-human dialogue [Sacks, Schegloff et al. 1974; Clark and Wilkes-Gibbs 1986; Clark and Schaefer 1989; Clark and Brennan 1991]. This research shows that conversations progress by reaching a state of mutual understanding about what was said and what was meant. This process of extending the knowledge shared by the conversation partners (their *common ground*) is called *grounding* [Clark and Schaefer 1989]. Since it is impossible to objectively determine when mutual understanding on some aspect of conversation has been reached, conversation partners assume mutual understanding as soon as a *grounding criterion* is fulfilled. Furthermore, conversation partners strive to minimize the total effort spent by both partners - an observation which Clark calls the *principle of least collaborative effort* [Clark and Wilkes-Gibbs 1986]. For example, it may take more effort to generate a perfect utterance than to produce a flawed utterance which is easy to repair. In summary, conversation partners collaborate on extending their common ground. Dialogue is structured as a sequence of turns. For more details on turn-taking in conversation, see [Sacks, Schegloff et al. 1974]. But how does this relate to human-computer interaction, in particular multimodal systems?

Increased interest in and availability of multimedia has initiated research to understand grounding in multimodal contexts. Clark and Brennan [Clark and Brennan 1991] point out that different media provide certain resources for grounding, associated with specific con-

straints and costs. A trade-off between resources, cost and constraints determines the preferred medium in a given (multimodal) communication setting. Applying the principle of least collaborative effort, different styles of grounding can be predicted. However, only a few case studies limited to artificial tasks have explored grounding in multimodal environments (see for instance [Traum and Dillenbourg 1996]). The subsequent sections will review research on the nature of errors in human-human dialogue and the strategies people employ to recover from them.

B.2 Taxonomies of Error in Natural Language Dialogue

Several taxonomies of errors in natural language dialogue have been proposed. Understanding categories of errors is valuable since they form the basis of the theory of repair in human-human dialogue, and they help to structure approaches to repair in human-computer dialogue via natural language. Two taxonomies are presented which differ in the dimension used to categorize errors: the linguistic level on which a communication problem occurs, and the communication stage when the problem occurs. Finally, the taxonomies are extended to include communication problems in natural language dialogue between a human and a computer.

The taxonomy proposed by Véronis [Véronis 1991] classifies errors according to the linguistic level on which they occur: either at the level of words (*lexical*), at the level of word sequences (*syntactic*), or at the level of meaning (*semantic*). Hirst [Hirst, McRoy et al. 1994] further subdivides the semantic category by distinguishing nonunderstanding and misunderstanding. *Nonunderstanding* is defined as an error where a conversation partner fails to find any complete interpretation of an utterance, and *misunderstanding* is defined as an error wher the listener does not arrive at the interpretation intended by the speaker. Figure B-1 illustrates this taxonomy.

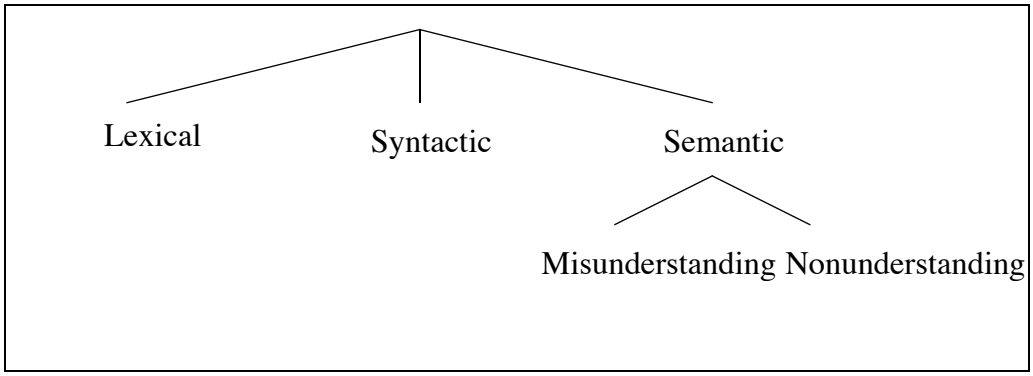


Figure B-1. *Taxonomy of errors in natural language dialogue according to linguistic level on which an error occurs*

Clark classifies communication problems in natural language dialogue according to the stage of communication when they occur [Clark 1994]. He distinguishes three stages: First, the stage of *vocalization* and *attention*: the speaker vocalizes an utterance, and the listener pays attention to what the speaker says. Second, the stage of *identification*: the listener identifies the sequence of words. Third, the stage of u*nderstanding*: speaker and listener reach the mutual belief that the listener understood what the speaker actually meant.

Figure B-2 shows an extension of Clark's taxonomy adopted by researchers on repair in dialogue systems. The taxonomy can help to identify different opportunities for prevention and resolution of errors in human-machine dialogue (e.g., [Brennan and Hulteen 1995; LuperFoy and Loehr 1997]). Compared with Clark's taxonomy, one communication stage is added (verbalization) and three high level categories are introduced (generation, channel, interpretation). The high level categories summarize the stages occuring on the sides of the speaker and listener, respectively; *generation* of an utterance by the speaker, transmission of the utterance from speaker to listener (the *channel*), and understanding of the utterance by the listener (*interpretation*). The stage of verbalization is added under the "generation" category. It is defined as the process of forming an appropriate sequence of words, before actually saying anything.

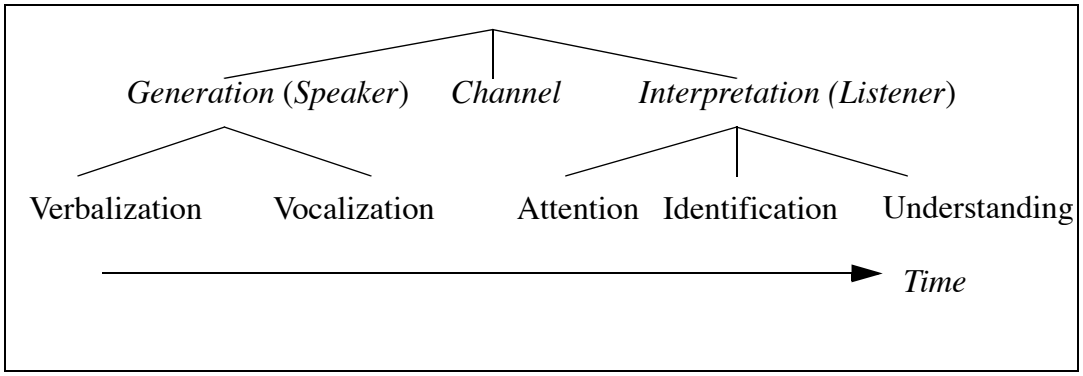


Figure B-2. *Error Taxonomy according to the communication stage when the error occurs*

Véronis extended her taxonomy to the situation of natural language dialogue between a human and a machine [Véronis 1991]. Errors are classified along two different dimensions: by who caused the error, and by the type of error. Errors in human-machine dialogue can be caused by either the system or the user. There are two error types: competence and performance errors[1]. Ignorance of the commonly accepted linguistic rules of well-formed language causes *competence* errors, whereas the lack of skill in applying these rules results in *performance* errors. These two types of errors are analogous to Norman's distinction between errors and slips [Norman 1988]. In summary, Véronis's taxonomy of errors in human-machine dialogue consists of a 2x2 matrix of error categories: user competence, user performance, system competence, and system performance.

To summarize, communication problems in natural language dialogue can occur on different linguistic levels, and during different stages of communication. However what strategies do we employ in dealing with communication problems? How do we recover from errors? What repair strategies are preferred?

1. The concepts of competence and performance error were originally introduced by [Chomsky 1965].

B.3 Repair in Human-Human Dialogue

Conversational techniques to deal with communication problems are so common that we frequently do not consciously notice when a communication problem has occurred and is being repaired. This section on repair in human-human dialogue begins by identifying the basic strategies people employ in dealing with communication problems. Then, research on the structure of conversational repair is presented. Repair can be categorized according to the sequential position of the turn in which the repair is initiated, and according to who initiates and performs the repair. People prefer to correct errors themselves, frequently still within the same turn in which the communication problem occurs. The final section reviews research in the medical field on how the hard-of-hearing overcome communication problems. This research identifies repetitions and paraphrases as preferred communication repair strategies. Repetitions are generally preferred over more complex repair strategies, unless multiple repair attempts are necessary to successfully complete a repair.

B.3.1 Strategies to deal with Communication Problems

Conversation analysis identifies three basic strategies that people employ in dealing with communication problems: preventing communication problems, monitoring conversation for potential communication problems, and collaborating on recovering from communication problems (e.g., [Sacks, Schegloff et al. 1974; Clark and Schaefer 1989]). Conversational techniques aimed at achieving the former two goals are described in the following two paragraphs. The much more developed research on repair in human-human dialogue is topic of the remainder of this appendix.

Numerous techniques for preventing communication problems have been identified. In noisy environments people commonly speak more loudly or simply move to a quieter room. In communication when communication problems are more likely to occur (e.g., when speaking with the hearing-impaired), a common technique is to speak more clearly. Other more generally useful techniques of rhetoric include repeating important facts, rephrasing, and elaborating

from the general to the specific.

Back-channel utterances play a crucial role in monitoring conversation for communication problems. Back-channel utterances include a variety of non-speech sounds (such as "uh huh", "um", "ah") and non-verbal cues (e.g., facial expressions, gestures). These back-channel cues provide feedback to the speaker about whether an utterance has been understood and when it has not.

The following two subsections review research on error recovery in human-human dialogue: what is the structure of conversational repair, and what strategies are preferred?

B.3.2 The Structure of Conversational Repair

Schegloff was the first researcher to look more closely at how people recover from communication problems in natural language dialogue, and his work is central to the theory of repair in human-human communication. Extending Clark's work on grounding in natural language dialogue (see Section B.1 earlier in this appendix), Schlegloff [Schegloff, Jefferson et al. 1977] examines how grounding works in the situation of repair. The following paragraph describes Schlegloff's characterization of conversational repair according to when repair is initiated and who performs the repair.

Schegloff argues that repairs are organized according to the temporal sequence of opportunities to *initiate* the repair. Beginning with the turn which caused the communication problem (trouble source turn), repair can be initiated either within that same turn, during the transition to the next turn, in that next turn, or in a following turn. According to which conversation partner *initiates* the repair, and who actually *performs* it, Schegloff distinguishes four different repair categories: In s*elf-initated* repair, the speaker notices the communication problem and initiates repair, whereas in *other-initiated* repair, another conversation partner initiates the repair. In *self-repair,* the speaker who caused the communication problems resolves it, whereas in *other-repair* another conversation partner. Figure B-3 shows the four main positions when repairs are typically initiated [Schegloff, Jefferson et al. 1977; Hirst, McRoy et al.

1994]. Figure B-4 illustrates the different types of repair with examples.

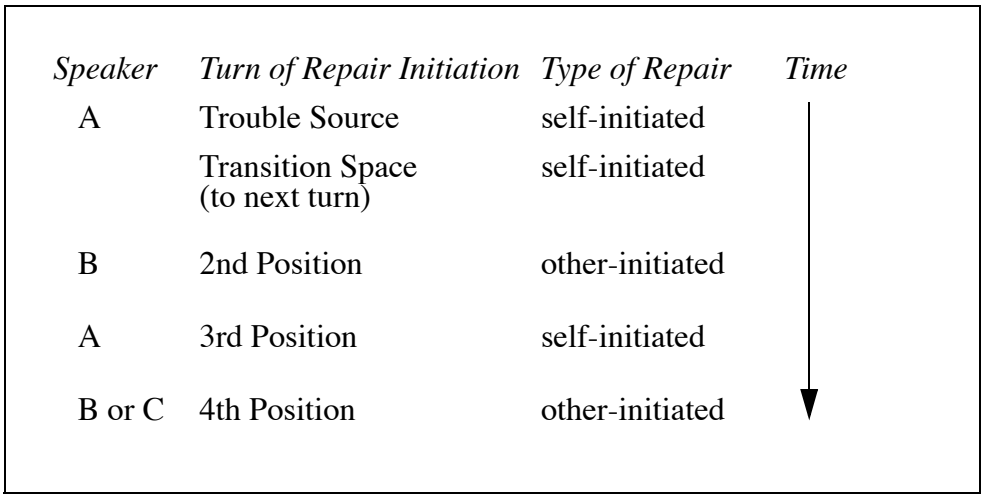


Figure B-3. *Main positions for initiating repair, according to Schegloff*

Self-initiated self-repair in trouble source turn:
A: I can't - hmm - I can meet on Tuesday.

Self-initiated self-repair in transition space to next turn:
A: I can meet on Tuesday (...) I mean Wednesday.

Other-initiation of repair in 2nd position:
A: I can meet on Tuesday.
B: I thought you were busy then.

Self-initiated self-repair in 3rd position:
A: Let's meet on Tuesday.
B: How about 2pm?
A: Oh, I just realize I'm busy Tuesday all day. Let's meet Wednesday instead.

Other-initiated other-repair in 4th position:
A: Do you know a group meeting will take place tomorrow?
B: Where?
A: I don't know.
B: Probably in the Red Conference room.

Figure B-4. *Examples for types of errors according to the position of repair initiation (from spontaneous human-human dialogues in a scheduling domain, cf. [Waibel 1996])*

However not all of these different types of repair are equally important. On the contrary, most

repairs are self-repair initiated within the trouble source turn or the transition space to the next turn. Schegloff describes this phenomenon as the *preference for self-correction* [Schegloff, Jefferson et al. 1977]. Why would self-repairs be preferred?

There are multiple explanations for the preference for self-correction. Since speakers constantly self-monitor their speech during the processes of verbalization and vocalization, the speaker is more likely to notice when an utterance is ill-formed [Jernudd and Thuan 1983]. But also the principle of least collaborative effort can explain the preference for self-correction: self-repair is preferred because it usually requires less overall effort.

Zahn [Zahn 1984] extends Schegloff's work by providing more data on the frequency of different conversational repairs, and by looking beyond sequential organization of turns as only explanation for the findings. He argues that in addition to sequential and structural determinants, repairs are sensitive to content, as well as social and communicative constraints. His analyses show that sequencing (repair initiation in same-turn versus in the transition space, or second, third and forth position) *and* context predict the structure of repair episodes much better than either independently.

Schegloff does not characterize what strategies people employ to resolve the communication problem, once it has been detected. The next section summarizes research in the field of medicine which investigated different conversational repair strategies.

B.3.3 Conversational Repair Strategies

Research in the medical field [Brinton, Fujiki et al. 1986; Brinton, Fujiki et al. 1988; Gagné, Stelmacovich et al. 1991] has investigated how the hard-of-hearing overcome communication problems with normal hearing partners. The goal of this research was to identify strategies which facilitate communication of the hearing-impaired with normally hearing people. It has been shown that training hearing-impaired adults in such strategies can decrease the number of unrepaired communication breakdowns [Gibson and Caissie 1994]. Before looking at specific strategies, the subsequent paragraph defines the concept of "communication repair strate-

gies".

Gagné defines a *communication repair strategy* as "any verbal or non-verbal action taken by an individual to overcome communication problems" [Gagné, Stelmacovich et al. 1991]. Hereby a communication problem is any message which a communication partner fails to understand in the way the speaker intended. *Requests for clarification* are the most common repair strategy. Gagné distinguishes between non-specific and specific requests for clarification. *Non-specific requests* typically ask to repeat the utterance, for example "What?", "Pardon me!", "Huh?". *Specific requests* ask for repetition of a specific constituent (e.g. "Where did he go?"). In terms of Schlegloff's taxonomy of conversational repair which was described in the previous section, requests for clarification elicit other-initiated self-repair.

Extending Gagné's work, Brinton [Brinton, Fujiki et al.] introduces five categories to code verbal responses to clarification requests which are shown below in Figure B-5. Communication repair strategies therefore include: repair by repetition, revision (paraphrase), addition and cue. Are there any preferences in the usage of these strategies?

Various studies examine the effectiveness of different communication repair strategies. Most studies report that specific requests for clarification are more effective than non-specific requests [Owens and Telleen 1981; Gagné and Wyllie 1989]. There is evidence that paraphrases are more effective than simple repetitions [Gagné and Wyllie 1989]. One study [Tye-Murray, Purdy et al. 1990] suggests that the effectiveness of these five strategies is not significantly different.

In summary, specific requests for clarification, and simple repetitions as well as paraphrases appear to be effective communication repair strategies. However, a communication problem may require more than one repair interaction to be resolved successfully. Are there any patterns in the use of strategies across multiple attempts at resolving the same communication problem?

Type	*Definition*	*Example*
Repetition	All or part of the original utterance is repeated verbatim.	A: How about next Tuesday? B: Huh? A: How about next Tuesday?
Revision	Modify surface form, but keep meaning of utterance	A: How about next Tuesday? B: Huh? A: How about coming Tuesday
Addition	Add some information to original utterance	A: How about next Tuesday? B: Huh? A: How about next Tuesday in the red conference room?
Cue	Define terms or give background context	A: How about next Tuesday? B: Huh? A: On Monday I am busy, but I could meet you on Tuesday.
Inappropriate	Speaker ignores request for clarification.	A: How about next Tuesday? B: Huh? A: Let's say at 4pm.

Figure B-5. *Brinton's taxonomy of conversational repairs in response to requests for clarification*

B.3.4 Strategy Preferences in Multiple Repairs

A single attempt at resolving a communication problem may not be sufficient, since the speaker knows what he wants to communicate, and it is hard to appreciate that the conversation partners do not share this knowledge. Furthermore, natural language is prone to ambiguity and draws heavily on contextual information. Therefore, multiple attempts to repair a communication problem may be necessary. Work by Brinton et. al [Brinton, Fujiki et al. 1988] extends the previously described studies on repair strategies of the hard-of-hearing in interactions with normally hearing people to the situation of multiple repairs.

To investigate repair strategy preferences in multiple repairs, Brinton examined verbal and non-verbal responses of children to stacked requests for clarification. A *stacked request for clarification* consists of several requests to clarify the same message. The results suggest that repetitions are frequently used initially. If multiple repair attempts are necessary, other strategies are employed. In particular, the frequency of cue responses increases dramatically after

the second request for clarification. Verbal repairs are often accompanied by changes in stress of intonation and by gestural cues.

These findings are quite intuitive and confirm that the principle of least collaborative effort guides strategic decisions in repairs: the easiest strategy to repair is to repeat utterance. If that fails, the conversation partners are willing to spend more effort by paraphrasing or providing background information.

B.4 Concluding Remarks

This appendix represents a cursory review of research on repair in human-human dialogue, with an eye on possible applications to the investigation of repair in human-machine dialogue in general, and speech user interfaces in particular. Please refer to Section 2.2, page 29, to find out how the theory of repiar in human-human dialogue relates to the approach chosen in this thesis.

Appendix C: Standard Benchmark Tasks for Continuous Speech Recognition

Intensive research on speech recognition algorithms in the past decade has resulted in significant performance improvements. In particular, programs like "Speech Language Technology" and "Human Language Technology", which are funded by the U.S. government's Advanced Research Project Agency (ARPA), have contributed to the progress in the field. Regular formal performance evaluations which are mandatory for all research sites funded by these programs have spurred competition among sites. Speech recognition systems are compared on standard benchmark recognition tasks, and the evaluations are administered by an independent organization, the National Institute for Standards in Technology (NIST). The following provides a cursory overview of the most important benchmark tasks in chronological order. The characteristics of each task, as well as the specific technological challenges introduced by each task, are highlighted.

Resource Management (RM): an artificial task centered around logistics in the military. The vocabulary is limited to 1000 words. The data set consists of 1000 sentences which were selected from the domain. Evaluations were held 1987-1991.

Air Travel Information Service (ATIS): the task revolves around scheduling flights. A costumer inquires about flight information, particularly flight schedules. He can ask for any appropriate constraint, for instance the cheapest flight from one location to another. For the first time, speech was recorded while the user tries to accomplish a "real" task, and the systems developed have obvious direct commercial applications. The domain is still very limited, and the vocabulary is size small.

Wall Street Journal (WSJ) / North American Business News (NAB): the task is to read sen-

tences from the Wall Street Journal or other major news sources. As evaluations progressed, additional conditions were included, such as using different microphones to record the speech, or collecting "spontaneously" dictated sentences. The main challenge in moving to this task was to tackle a much broader domain, with a large (potentially unlimited) vocabulary. Since 1994, the task was expanded to include business news text other than the Wall Street Journal, including the New York Times and Reuter.

Switchboard (SWB): phone conversations between people talking from home, about 70 different topics, ranging from sports to gun control. Added difficulties are the telephone channel (low bandwidth and noise) and the spontaneous *conversational* character of the speech.

Spontaneous Scheduling Tasks: two people try to schedule a meeting. Data is recorded in different languages, including German, English, Spanish, Japanese and Korean. The initial letter of the language serves as initial letter for the abbreviation of these databases: GSST for the German database; ESST, SSST, JSST and KSST accordingly. Similar to Switchboard, the challenge of this task is spontaneous, conversational speech. The vocabulary size is much smaller, to ease the diffiulty: these tasks were defined for speech translation projects, and speech translation on a task like Switchboard would not be feasible with current technology.

Call Home: a recent extension of the Switchboard task, people who are travelling are calling "home". No restrictions on topics are imposed. Speech is collected in several languages, including English, Spanish and German.

Table 33 shows important quantitative characterists of these benchmark tasks: the years during which data was collected and regular performance evaluations were held, the method of data collection, the size of the database (text; the amount of transcribed audio data is considerably smaller for all benchmark tasks since WSJ), the vocabulary size, the perplexity of standard language models (until ATIS bigram perplexity, since WSJ trigram perplexity), and the recog-

nition performance (word accuracy) of the best system in the most recent evaluation.

Task	Years	Data Collection	Database Size	Vocabulary Size	PP	Best WA
RM	87-91	read given text	1000 sentences	1000	60	97.5%
ATIS	89-94	record spontaneous dialogues	~20000 sentences	1800	20	97%
WSJ/ NAB	92-95	read printed text	38 Bio. / 300+ Bio. words	60,000	140	93%
ESST	93-96	spontaneous scheduling dialogues	200,000	2,000	40	80%
Broadcast News	95-today	radio and TV news broadcasts	50 h (1997)	unlimited		73%
SWB/Call Home	94-today	spontaneous telephone conversations	2 Mio. / Mio words	20,000 / unlimited	70	70%

Table 33: *Important benchmark tasks for (U.S. English) continuous speech recognition*

Appendix D: Experiment Data

Table 34: Demographic data of participants of final user study

Participant ID	Age Group	Sex	Completed Education	Prior Experience with Speech Recognition	(Self-reported) Typing Skill (scale: 1-6)	Measured Typing Speed [wpm]
lp	<20	f	n/a	no	3	25
lm	20-30	f	high-school	no	2	19
td	20-30	m	B.S.	no	4	30
tb	20-30	m	undergrad	no	3	30
mr	20-30	m	high-school	no	4	28
dj	<20	m	undergrad	no	4	36
br	20-30	m	undergrad	n/a	5	31
tg	20-30	f	undergrad	no	5	31
jm	20-30	m	undergrad	no	4	34
kg	<20	f	undergrad	some Dragon-Dictate	3.5	40
ss	30-40	f	undergrad	no	4	48
hd	<20	f	undergrad	no	5	40
jl	30-40	f	undergrad	no	1	25
ag	<20	f	undergrad	no	3	17
mm	20-30	m	M.S.	some demo sys-tems	3	27

Appendix E: Glossary

Confidence Measure: Statistical method that measures the recognition system's confidence in having recognized the input correctly.

Conversational (Interface, Repair, ...): Communication with a computer system that imitates natural spoken dialogue.

Correction Method: Set of correction modalities that allows the user to effectively correct recognition errors

Clarification Dialogue: Interactive correction of recognition errors in a spoken language dialogue, similar to conversational repair methods employed in human-human dialogue. Synonyms: repair dialogue.

Cross-modal correction: Interactive correction of recognition modality that switches modality, compared to the primary input, for example, from continuous speech to spelling or handwriting.

Deictic References: Referring to one object out of many objects which are all within the visible range, either by language, or using hand gestures.

Dialogue (or conversational) speech recognition application: Speech recognition application which involves the user in a dialog, analogous to human-human conversation. Synonyms: dialogue system, spoken dialogue system.

Error Correction: Method to resolve a communication problem (e.g., recognition errors) once it has been detected. Used in this thesis mainly in the context of lexical system interpretation errors. Synonyms: error recovery, repair.

(Pen-drawn) Gesture: In the context of this dissertation, mark that is drawn on a writing-sensitive screen, with either a pen or a finger. By contrast, *pointing* refers to selecting objects with a pointing device (typically a mouse); and *3d gesture* refers to a movement of hands, arms,

heads, etc. in three dimensional space.

Hidden Markov Model, HMM: Statistical model widely used (among others) in acoustic modeling for speech recognizers. Automaton with states and transitions that are weighted by probabilities. State transitions depend only on the previous state (Markov assumption). Each (hidden) state is associated with an output probability density function that indicates how probable measurable output events are.

Human-human dialogue: Spoken natural language dialogue between two or more people, as opposed to written communication between people, and human-machine dialogue. Synonyms: "conversation", "natural language dialogue", "discourse".

Multimodal: Offering more than one input modalitites, including speech. Includes both modalities humans traditionally use (handwriting, facial expressions, gestures) as well as artifical modalities (keyboard, pointing devices).

Multimodal Dictation System, **Multimodal Text Editor**: Automatic dictation system that offers effective editing and error correction without keyboard input, using multimodal interactive correction methods.

N-best List: Used as short term for the list of the N (N some given constant) best matching hypotheses that an automatic recognizer can identify for some input signal.

New Word, Out-of-vocabulary Word (OOV): Word that is outside the vocabulary of an automatic recognition system.

Partial-Word Correction: Interactive method to correct errors (in speech recognition applications) that allows the user to correct on the level of characters within a word, rather than entire words or sequences of words.

Repair: Any conversational technique which has the global to prevent or resolve communication problems in (human-human or human-machine) natural language dialogue. Used as the

broad category comprising several types of communication problems (lexical, syntactic or semantic) and several strategies to deal with them (prevention, monitoring, resolution).

Respeaking: Interactive method to correct errors (in speech recognition applications) by repeating the input using continuous speech.

Speech User Interface: User interface that incorporates speech recognition technology as an input modality. Synonym: speech-enabled interface.

Spelling: Interactive method to correct errors (in speech recognition applications) by spelling words verbally, such as S-P-E-L-L-I-N-G.

Spoken Language System: System which allows users to communicate with it using spoken language. Synonyms: Dialogue system, speech recognition application.

Time Delay Neural Network, TDNN: Artificial neural network architecture that achieves time-invariance by feeding a window of input features (over several consecutive times) into the input layer, and potentially carrying windows of processed features on across the hidden layers. Word-level information can be incorporated by feeding the results of a TDNN-based phonetic classifier into a standard dynamic time warping (DTW) search, resulting in a neural network architecture called "Multi-State TDNN".

Unimodal Correction: Interactive method to correct errors (in speech recognition applications) that offers only a single modality for correction.

Bibliography

(1998). "Are You Talking to Me?". Newsweek, April 1997, p. 85-86.

Acero, A. (1998). Personal Communication.

Ainsworth, W. A. (1992). “Feedback strategies for error correction in speech recognition systems.” International Journal of Man-Machine Studies **36**: 833-842.

Allen, J. F., Miller, B. W., Ringger, E. K. and Sikorski, T. (1996). "A Robust System for Natural Spoken Dialogue". 34th Annual Meeting of the ACL, June 1996.

Alleva, F., Huang, X., Hwang, M.-Y. and Jiang, L. (1997). "Can Continuous Speech Recognizers Handle Isolated Speech?". European Conference on Speech Communication and Technology EUROSPEECH, Rhodes (Greece), September 22-25, 1997, ESCA. **2:** 911-914.

Alto, P., Brandetti, M., Ferretti, M., Maltese, G. and Scarci, S. (1989). "Experimenting Natual-Language Dictation with a 20000-Word Speech Recognizer". VLSI and Computer Peripherals, IEEE Computer Society Press. **2:** 78-81.

Asadi, A., Schwartz, R. and Makhoul, J. (1990). "Automatic Detection of New Words in a Large-Vocabulary Continuous Speech Recognition System". International Conference on Acoustics, Speech, and Signal Processing, Albuquerque, IEEE, p. 125-128.

Baber, C. and Hone, K. S. (1993). “Modeling error recovery and repair in automatic speech recognition.” International Journal of Man-Machine Studies **39**: 495-515.

Baber, C., Stammers, R. B. and Usher, D. M. (1990). "Error correction requirements in automatic speech recognition". *In:* Contemporary Ergonomics. E. J. Levesey. London, Taylor and Francis.

Blattner, M. M. and Dannenberg, R. (1990). “CHI '90 Workshop on Multimedia and Multimodal Interface Design.” Bulletin of the ACM Special Interest Group on Computer-Human Interaction **22**(2): 54-57.

Bolt, R. A. (1980). “Put-That There: Voice and Gesture at the Graphics Interface.” Computer Graphics Journal of the Association of Computing and Machinery **14**(3): 262-270.

Borenstein, N. S. (1985). "The Evaluation of Text Editors: A Critical Review of the Roberts and Moran Methodology Based on New Experiments". International Conference on Computer-Human Interaction, April 1985, ACM. **1:** 99-105.

Bradford, J. H. (1990). “Semantic strings: a new technique for detecting and correcting user errors.” International Journal of Man-Machine studies **33**: 399-407.

Brennan, S. E. and Hulteen, E. A. (1995). “Interaction and feedback in a spoken language system: a theoretical framework.” Knowledge Based Systems **8**: 143-151.

Briggs, R. O., Beck, B. S., Dennis, A. R., Carmel, E., Nunamaker, J. F. and Pfarrer, R. (1992). "Is the Pen Mightier Than the Keyboard?". 25th Hawai International Conference on Systems Sciences, Kauai (HI), IEEE. **3:** 201-210.

Brinton, B., Fujiki, M. and Sonnenberg, E. A. (1988). “Responses to requests for clarification in linguistically normal and language-impaired children in conversation.” Journal of Speech and Hearing Disorders **53**: 383-391.

Brinton, B., Fujiki, M., Winkler, E. and Loeb, D. (1986). “Responses to requests for clarification by lingustically normal and language-impaired children.” Journal of Speech and Hearing Disorders **51**: 370-378.

Card, S. K., Moran, T. P. and Newell, A. (1980). “The Keystroke Level Model for User Performance Time with Interactive Systems.” Communications of the Association of Computing and Machinery **23**: 396-410.

Card, S. K., Newell, A. and Moran, T. P. (1983). "The Psychology of Human-Computer Interaction". Hillsdale (NJ), L. Erlbaum Associates.

Casey, M. A., Gardner, W. G. and Basu, S. (1995). "Vision Steered Beam-forming and Transaural Rendering for the Artificial Life Interactive Video Enfironment (ALIVE)". Technical Report 352, 1995, MIT Media Lab.

Chase, L. L. (1997). "Error-Response Feedback Mechanisms for Speech Recognizers". P.h.D. Thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh (PA). 261 pages.

Cheyer (1997). "MVIEWS: Multimodal Tools for the Video Analyst". International Conference on Intelligent User Interfaces, San Francisco (CA), January 6-9, 1998, ACM Press, p. 55-62.

Cheyer, A. and Julia, L. (1995). "Multimodal Maps: An agent-based approach". International Conference on Cooperative Multimodal Communication, Eindhoven (NL), May 1995.

Chomsky, N. (1965). "Aspects of Theory of Syntax". Ph.D. Thesis, Research Laboratory of Electronics, Massassuchetts Institute of Technology, Cambridge (MA). 251 pages.

Clark, H. H. (1987). “Collaborating on contributions to conversations.” Language and Cognitive Processes **2**: 1-13.

Clark, H. H. (1994). “Managing Problems in Speaking.” Speech Communication **15**: 243-250.

Clark, H. H. and Brennan, S. E., Eds. (1991). "Grounding in Communication". Perspectives on Socially Shared Cognition, APA.

Clark, H. H. and Schaefer, E. F. (1989). “Contributing to Discourse.” Cognitive Science **13**: 259-294.

Clark, H. H. and Wilkes-Gibbs, D. (1986). “Referring as collaborative process.” Cognition **22**: 1-39.

Cole, R. A., Mariani, J., Uszkoreit, H., Zaenen, A. and Zue, V., Eds. (1995). "Survey of the State of the Art in Human Language Technology".

Coleman, M. L. (1969). "Text editing on a graphic display device using hand-drawn proofreader's symbols". Second University of Illinois Conference on Computer Graphics, Urbana Champaign (IL), University of Illinois Press, p. 283-290.

Dahlbäck, N., Joensson, L. and Ahrenberg, L. (1992). "Wizard of Oz Studies - why and how". Technical Report LiTH-IDA-R-92-19, 1992.

Danieli, M. (1996). "On the use of expectations for detecting and repairing human-machine miscommunication". AAAI Workshop on Detecting, Repairing and Preventing Human-Machine Miscommunication, Portland (OR), August 1996.

Danis, C. M. (1989). "Developing Successful Speakers for an Automatic Speech Recognition System". Proceedings of the Human Factors Society 33rd Annual Meeting, Computer Systems: Speech and Writing. **1:** 301-304.

Dybjkaer, L., Bernsen, N. O. and Dybjkaer, H. (1996). "Reducing Miscommunication in Spoken Human-Machine Dialogue". AAAI Workshop on Detecting, Repairing and Preventing Human-Machine Miscommunication, Portland (OR), August 1996.

Farag, R. F. H. (1979). “Word Level Recognition of Cursive Script.” IEEE Transactions on Computers **28**: 172-175.

Fauré, C. and Julia, L. (1993). "Interaction Homme-Machine par la Parole et le Geste pour l'edition de documents: TAPAGE". International Conference on Interfaces to Real and Virtual Worlds, March 1993, p. 171-180.

Finke, M. and Rogina, I. (1997). "Wide Context Acoustic Modeling in Read versus Spontaneous Speech". International Conference on Acoustics, Speech and Signal Processing, Munich (Germany), IEEE. **3:** 1743-1746.

Fisher, W. M. (1996). "Factors Affecting Rcognition Error Rate". ARPA Workshop on Spoken Language Technology, New York, April 1996.

Flanagan, J. (1997). "Synergetic Modalities for Human/Machine Communication". IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara (CA), IEEE Signal Processing Society, p. 1-8.

Frankish, C., Hull, R. and Morgan, P. (1995). "Recognition Accuracy and User Acceptance of Pen Interfaces". International Conference on Computer-Human Interaction, Denver (CO), ACM, p. 503-510.

Fraser (1991). “Simulating Speech Systems.” Computer Speech and Language **5**: 81-99.

Gagné, J. P., Stelmacovich, P. and Yovetich, W. (1991). “Reactions to Requests for Clarification Used by Hearing-Impared Individuals.” Volta Review, 1991, p.129-143.

Gagné, J. P. and Wyllie, K. A. (1989). “Relative effectiveness of three repair strategies on the visual identification of misperceived words.” Ear and Hearing **10**: 368-374.

Gamma, E. (1995). "Design patterns: Elements of reusable Object-Oriented Software". Reading (MA), Addison-Wesley.

Gibbon, D., Moore, R. and Winski, R., Eds. (1997). "Handbook of Standards and Resources for Spoken Language Systems". Berlin, New York, Mouton de Gruyter.

Gibson, C. L. and Caissie, R. (1994). “The Effectiveness of Repair Strategy Intervention with a Hearing-Impaired Adult.” Journal of Speech and Language Pathologies **18**(1): 14-22.

Gomoll, K. (1990). "Some Techniques for Observing Users". *In:* The Art of Human-Computer Interface Design. B. Laurel, Addison-Wesley, p. 85-90.

Gould, J. D. (1978). “How Experts Dictate.” Journal of Experimental Psychology: Human Perception and Performance **4**(4): 648-661.

Gould, J. D., Conti, J. and Hovanyecz, T. (1983). “Composing Letters with a Simulated Listening Typewriter.” Communications of the ACM **26**(4): 295-308.

Govindaraju, V., Gyeonghwan, K. and Srihari, S. N. (1997). "Paradigms in Handwriting Recognition". IEEE International Conference on Systems, Man , and Cybernetics, Orlando (FL), October 12-15, 1997, IEEE. **2:** 1498-1503.

Guyon, I., Henderson, D., Albrecht, P., LeCun, Y. and Denker, J. (1992). "Writer Independent and Writer Adaptive Neural Network for On-line Character Recognition". *In:* From Pixels to Feature III: Frontiers in Handwriting Recognition. I. S. and J. C. Simon, Elsevier Science Publishers.

Haffner, P. and Waibel, A. (1992). "Multi-State Time Delay Neural Networks for Continuous Speech Recognition". *In:* Advances in Neural Network Information Processing Systems. San Mateo, Morgan Kaufmann.

Harding, T. H., Martin, J. S. and Beasley, H. H. (1996). "A Survey of Flat Panel Display Technologies". Technical Report . U.S. Army Aeromedical Research Laboratory, March 1996

Hauptmann, A. G. (1989). "Speech and Gestures for Graphic Image Manipulation". International Conference on Computer-Human Interaction, Austin (TX), April 1989, ACM. **1:** 241-245.

Hild, H. (1997). "Buchstabiererkennung mit neuronalen Netzen in Auskunftssystemen". Ph.D. Thesis, Computer Science Department, Fredericiana University, Karlsruhe (Germany), 1997, 216 pages.

Hild, H. and Waibel, A. (1995). "Integrating Spelling into Spoken Dialogue Recognition". International Conference on Acoustics, Speech and Signal Processing, Madrid (Spain), September 18-21, 1995, IEEE Computing Society.

Hildebrandt, T. H. and Liu, W. (1993). “Optical Recognition of handwritten Chinese Characters: Advances since 1980.” Pattern Recognition **26**(2): 205-225.

Hirst, G., McRoy, S., Heeman, P., Edmonds, P. and Horton, D. (1994). “Repairing Conversational Misunderstandings and Non-Understandings.” Speech Communication **15**: 213-229.

Huang, X. (1998). Personal Communication.

Hwang, M.-Y. (1993). "Subphonetic Acoustic Modeling for Speaker-Independent Continuous Speech Recognition". Ph. D. Thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh (PA). 161 pages.

Jacob, R. J. K. (1993). "Eye-Movement Based Human-Computer Interaction Techniques: Toward Non-Command Interfaces". *In:* Advances in Human-Computer Interaction. H. R. Hartson and D. Hix, Ablex Publishing. **4:** 151-189.

Jelinek, F. (1990). "Self-Organized Language Modeling for Speech Recognition". *In:* Readings in Speech Recognition. A. Waibel and K.-F. Lee. San Mateo, CA, Morgan Kaufmann**:** 450-506.

Jernudd, B. H. and Thuan, E. (1983). “Control of language through correction in speaking.” International Journal of the Society of Language **44**: 71-97.

Kamm, C. A. and Walker, M. A. (1997). "Design and Evaluation of Spoken Dialog Systems". IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara (CA), December 13-17, 1997, IEEE.

Kankaanpaa, A. (1988). “FIDS - A Flat-Panel Interactive Display System.” IEEE Computer Graphics and Applications, March 1998, p. 71-82.

Kemp, T. and Schaaf, T. (1997). "Estimating Confidence using Word Lattices". European Conference on Speech Communication and Technology EUROSPEECH, Rhodos (Greece)**. 2:** 827-830.

Koons, D. B., Sparrell, C. J. and Thorisson, K. R. (1993). "Integrating Simultaneous Input from Speech, Gaze, and Hand Gestures". *In:* Intelligent Multimedia Interfaces. M. Maybury, Morgan Kaufmann, p. 257-275.

Kuhn, R., Lazadrides, A., Normandia, Y. and Browsseau, J. (1995). "Improved Decision Trees for Phonetic Modeling". International Conference on Acoustics, Speech and Signal Processing, Detroit (Michigan), IEEE. **2:** 552-555.

Lai, J. and Vergo, J. (1997). "MedSpeak: Report Creation with Continuous Speech Recognition". International Conference on Computer-Human Interaction CHI, Atlanta (USA), March. **1:** 431-438.

Landay, J. (1996). "Interactive Sketching for the Early Stages of User Interface Design". Ph.D. Thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh (PA), 1996.

Lee, K.-F. (1990). "Large Vocabulary Speech Recognition - the SPHINX System". P.h.D. Thesis, Computer Science Department, Carnegie Mellon University, Pittsburgh (PA), 1990.

Legetter, C. J. and Woodland, P. C. (1996). “Maximum likelihood linear regression for speaker adaptation of continuous density HMMs.” Computer, Speech and Language **9**: 171-186.

Leopold, J. L. and Ambler, A. L. (1997). "Keyboardless Visual Programming Using Voice, Handwriting, and Gesture". IEEE Symposium on Visual Languages, Isle of Capri (Italy), September 23-26, 1997, IEEE Computer Society,p. 28-35.

LuperFoy, S. and Loehr, D. (1997). "Run-Time Discourse Processing To Supplement Incomplete ASR". IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara (CA), December 13-17, 1997, IEEE.

Madhvanath, S. (1996). "The Holistic Paradigm in Handwritten Word Recognition and its Application to Large and Dynamic Lexicon Scenarios". Ph.D. Thesis, Computer Science Department, State University of New York, Buffalo (NY).

Manke, S. (1998). "On-line Erkennung kursiver Handschrift bei großen Vokabularien (On-line Recognition of Cursive Handwriting with Large Vocabularies)". Ph.D. Thesis, Computer Science 'Department, Fredericiana University, Karlsruhe (Germany), 1998.

Manke, S., Finke, M. and Waibel, A. (1995). "NPen++: A Writer Independent, Large Vocabulary On-Line Cursive Handwriting Recognition System". International Conference on Document Analysis and Recognition, Montreal.

Martin, T. B. and Welch, J. R. (1980). "Practical speech recognisers and some performance effectiveness parameters". *In:* Trends in Speech Recognition. W. A. Lea. Englewood Cliffs (NJ), Prentice Hall.

McNair, A. E. and Waibel, A. (1994). "Improving Recognizer Acceptance through Robust, Natural Speech Repair". International Conference on Spoken Language Processing, Yokohama (Japan). **3:** 1299-1302.

Mellor, B. and Baber, C. (1997). "Modelling of Speech-based User Interfaces". European Conference on Speech Communication and Technology EUROSPEECH, Rhodes (Greece), September 22-25, 1997, ESCA. **4:** 2263-2266.

Mostow, J., Roth, S., Hauptmann, A. G. and Kane, M. (1994). "A Prototype Reading Coach that Listens". Twelfth National Conference on Artificial Intelligence AAAI, Seattle (WA), p. 785-792.

Murray, A., Frankish, C. F. and Jones, D. M. "Data entry by voice: facilitating correction of misrecognitions". *In:* Interactive Speech Technology. C. Baber (ed.).

Nerzig, P. (1996). "Two Methods for Recognizing Erroneous Plans in Human-Machine Dialogues". AAAI Workshop on Detecting, Repairing and Preventing Human-Machine Miscommunication, Portland (OR), August 4, 1996.

Newell, A. F., Arnott, J. L., Dye, R. and Cairns, A. Y. (1991). “A full-speed listening typewriter simulation.” International Journal of Man-Machine studies **35**: 119-131.

Ney, H., Essen, U. and Kneser, R. (1994). “On Structuring Probabilistic Dependencies in Stochastic Language Modeling.” Computer Speech and Language **8**(1): 1-38.

Nigay, L. and Coutaz, J. (1993). "A Design Space for Multimodal Systems: Concurrent Processing and Data Fusion". International Conference on Computer-Human Interaction, 24-29 April, 1993, ACM Press, p. 172-178.

Norman, D. (1988). "The Psychology of Everyday Things". New York, Basic Books.

Nouboud, F. and Plamondon, R. (1990). “On-line Recognition of Handprinted Characters: Survey and Beta Tests.” Pattern Recognition **23**(9): 1031-1044.

Ostendorf, M., Kannan, A., Austin, S., Kimball, O., Schwartz, R. and Rohlicek, J. R. (1991). "Integration of diverse recognition methodologies through reevaluation of N-best sentence hypotheses". Proceedings DARPA Speech and Natural Language Processing Workshop, Pacific Grove, CA, p. 83-87.

Oviatt, S., Cohen, P. R. and Wang, M. (1995). “Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity.” Speech, Language and Communication, 1995, **15**: 283-300.

Oviatt, S., DeAngeli, A. and Kuhn, K. (1997). "Integration and Synchronization of Input modes during multimodal Human-Computer Interaction". International Conference on Computer-Human Interaction, Atlanta (GA), ACM. **1:** 415-422.

Oviatt, S., Levow, G. A., MacEachern, M. and Kuhn, K. (1996). "Modeling Hyperarticulate Speech During Human-Computer Error Resolution". International Conference on Spoken Language Processing, Philadelphia (PA), October 1996. **2:** 797-800.

Oviatt, S. and VanGent, R. (1996). "Error Resolution During Multimodal Human-Computer Interaction". International Conference on Spoken Language Processing, Philadelphia (PA), October 1996. **2:** 204-207.

Owens, E. and Telleen, C. C. (1981). “Tracking as an aural rehabilitative process.” Journal of the Academy of Rehabilitative Audiology **14**: 259-273.

Pallett, D. S., Fiscus, J. G., Fisher, W. M., Garofolo, J. S., Lund, B. A., Martin, A. and Przybocki, M. A. (1994). "1994 Benchmark Tests for the ARPA Spoken Language Program". ARPA Workshop on Spoken Language Technology, Princeton (NJ), Margan Kaufmann Publishers, Inc., p. 5-36.

Paul, D. (1994). "The Lincoln Large Vocabulary Stack Decoder Based HMM Continuous Speech Recognizer". ARPA Workshop on Spoken Language Technology, Plainsboro (NJ), March 6-8, 1994, Morgan Kaufmann, p. 120-134.

Rabiner, L. (1991). "A tutorial on Hidden Markov Models and selected applications in speech recognition". *In:* Readings in Speech Recognition. A. Waibel and K.-F. Lee, Morgan Kaufman.

Rhyne, J. (1987). “Dialogue Management for Gestural Interfaces.” Computer Graphics **21**(2): 137-142.

Rhyne, J. R. and Wolf, C. G. (1993). "Recognition-Based User Interfaces". 'In:'. Advances in Human-Computer Interaction. H. R. Hartson and D. Hix. Norwood (NJ), Ablex Publishing. **4:** 191-212.

Ries, K. (1996). "Class Phrase Models for Language Modeling". International Conference on Spoken Language Processing, Philadelphia (PA), October 1996, p. 398-401.

Ringger, E. K. and Allen, J. F. (1996). "A Fertility Channel Model for Post-Correction of Continuous Speech Recognition". International Conference on Spoken Language Processing, Philadelphia (PA), p. 893-897.

Robbe, S., Carbonell, N. and Valot, C. (1996). "Towards usable multimodal command languages: Definition and ergonomic assessment of constraints on users' spontaneous speech and gestures". International Conference on Spoken Language Processing, Philadelphia (PA), October 16-20, 1996, IEEE. **3:** 1655-1658.

Roberts, T. L. and Moran, T. P. (1983). “The Evaluation of Text Editors: Methodology and Empirical Results.” Communications of the Association of Computing and Machinery **26**(4): 265-283.

Rogina, I. and Waibel, A. (1995). "The JANUS Speech Recognizer". ARPA Workshop on Spoken Language Technology, Austin (TX), January, Morgan Kaufmann, p.166-169.

Rosenfeld, R. (1994). "A Maximum-Entropy Approach to Language Modeling". Ph.D. Thesis Computer Science Department, Carnegie Mellon University, Pittsburgh PA. 104 pages.

Rubine, D. (1991). “Specifying Gestures by Example.” ACM Journal on Computer Graphics **25**(4): 329-337.

Rudnicky, A. I., Hauptmann, A. G. and Lee, K.-F. (1994). “Survey of Current Speech Technology.” Communications of the Association of Computing and Machinery **37**(3): 52-57.

Rudnicky, A. I., Reed, S. D. and Thayer, E. H. (1996). "SpeechWear: A Mobile Speech System". International Conference on Spoken Language Processing, Philadelphia (PA), October 1996. **1:** 538-541.

Sacks, H., Schegloff, E. A. and Jefferson, G. (1974). “A simplest systematics for the organzation of turn-taking in conversation.” Language **50**: 696-735.

Sarukkai, R. R. and Hunter, C. (1997). "Integration of Eye Fixation Information with Speech Recognition Systems". European Conference on Speech Communication and Technology (EUROSPEECH), Rhodes (Greece), September 18-22, 1997**, 1:**97-100.

Scattone, F., Baker, J., Gilliek, L., Orloff, J. and Roth, R. (1993). "Dragon's Large Vocabulary Speech Recognition System". ARPA Workshop on Spoken Language Technology, 1993.

Schaaf, T. (1996). "Vertrauensmasse für die maschinelle Spracherkennung". Master Thesis, Computer Science Department, Fredericiana University, Karlsruhe (Germany), 1996.

Schegloff, E. A., Jefferson, G. and Sacks, H. (1977). “The preference for self-correction in the organization of repair in conversation.” Language **53**: 361-382.

Schwartz, R. and Chow, Y.-L. (1990). "The N-best Algorithm: An Efficient and Exact Procedure for Finding the N Most Likely Sentence Hypotheses". International Conference on Acoustics, Speech and Signal Processing, Albuquerque (NM), April, IEEE. **1:** 81-84.

Setlur, A. R., Sukkar, R. A. and Jacob, J. (1996). "Correcting Recognition Errors via Discriminative Utterance Verification". International Conference on Spoken Language Processing, Philadelphia (PA), October 1996. **2:** 598-601.

Shneiderman, B. (1997). "Designing the User Interface: Strategies for Effective Human-Computer Interaction". Menlo Park, Addison Wesley.

Soltau, H. (1998). Personal Communication.

Srihari, R. and Baltus, C. M. (1993). "Incorporating syntactic constraints in recognizing handwritten sentences". International Joint Conference on Artificial Intelligence, Chambery (France), AAAI, p. 1262-.

Starner, T., Makhoul, J., Schwartz, R. and Chou, G. (1994). "On-line Cursive Handwriting Recognition Using Speech Recognition Methods". International Conference on Acoustics, Speech and Signal Processing, Adelaide, April 1994, IEEE.

Suhm, B. (1993). "Das Problem Neuer Wörter in Spracherkennungssystemen". Master Thesis Computer Science Department, Fridericiana University, Karlsruhe (Germany), 1993.

Suhm, B. and Waibel, A. (1994). "Towards Better Language Models for Spontaneous Speech". International Conference on Spoken Language Processing, Yokohoma (Japan), September 1994, **2**:831-834.

Sweeney, M., McGuire, M. and Shackel, B. (1993) "Evaluating user-computer interaction: a

framework", International Journal of Man-Machine Studies, 1990, **38(3):**689-711.

Tannas, L. E., Ed. (1985). "Flat-Panel Displays and CRTs". New York, Van Nostrand Reinhold.

Thomas, C. (1987). "Designing Electronic Paper to Fit User Requirements". *In:* People and Computers III. D. Diaper and R. Winder (eds.), p. 247-257.

Traum, D. R. and Dillenbourg, P. (1996). "Miscommunication in Multi-modal Collaboration". AAAI Workshop on Detecting, Repairing and Preventing Human-Machine Miscommunication, Portland (OR), August 4, 1996.

Trubitt, D. (1990). “Alternate Input Devices.” Electronic Musician, December 1990, p.74-77.

Tucker, A. B., Ed. (1997). “The Computer Science and Engineering Handbook”. Boca Raton (FL), CRC Press.

Tye-Murray, N., Purdy, S. C., Woodworth, G. G. and Tyler, R. S. (1990). “Effects of repair strategies on visual identification of sentences.” Journal of Speech and Hearing Disorders **55**: 621-627.

Véronis, J. (1991). “Error in natural language dialogue between man and machine.” International Journal of Man-Machine studies: 187-217.

Vo, M. T. (1998). ”A Framework and Toolkit for the Construction of Multimodal Learning Interfaces”. Ph.D. Thesis, Computer Science Department, Carnegie Mellon, Pittsburgh. 195 pages.

Vo, M. T. and Wood, C. (1996). ”Building an application framework for speech and pen input integration in multimodal learning interfaces”. International Conference on Acoustics, Speech and Signal Processing, Atlanta (GA), May 1996, IEEE. **6:** 3545-3548.

Waibel, A. (1996). “Translation of Conversational Speech.” Speech Communication **29**(7): 41-48.

Waibel, A., Suhm, B., Vo, M. T. and Yang, J. (1997). "Multimodal Interfaces for Multimedia Information Agents". International Conference on Acoustics, Speech and Signal Processing, Munich (Germany), IEEE Signal Processing Society.

Wang, J. (1995). "Integration of Eye-gaze, Voice and Manual Response in multimodal User Interface". IEEE Conference on Systems, Man and Cybernetics, Vancouver (BC), IEEE Computer Society. **4:** 3938-3942.

Weintraub, M. (1995). "Why Are LVCSR Error Rates So High?". Menlo Park (CA), SRI International.

Wolf, C. G. and Morrel-Samuels, P. (1987). "The use of hand-drawn gestures for text editing." International Journal of Man-Machine studies **27**: 91-102.

Woszczyna, M. (1998). "Fast Speaker Independent Large Vocabulary Continuous Speech Recognition", Ph.D. Thesis, Computer Science Department, Fredericiana University, Karlsruhe

Young, S. (1995). "Large Vocabulary Continuous Speech Recognition: A Review." .

Zahn, C. J. (1984). "A Reexamination of Conversational Repair." Communication Monographs **51**: 56-66.

Zeppenfeld, T., Finke, M., Ries, K., Westphal, M. and Waibel, A. (1997). "Recognition of Conversational Telephone Speech using the JANUS Speech Engine". International Conference on Acoustics, Speech and Signal Processing, Munich (Germany), IEEE.

Zoltan-Ford, E. (1991). "How to get people to say and type what computers can understand." International Journal of Man-Machine studies **34**: 527-547.

Index

Symbols

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V