ResearchGate

See discussions, stats, and author profiles for this publication at:
http://www.researchgate.net/publication/220286396

Multimodal error correction for speech user interfaces.

ARTICLE *in* ACM TRANSACTIONS ON COMPUTER-HUMAN INTERACTION · JANUARY 2001

Impact Factor: 0.57 · DOI: 10.1145/371127.371166 · Source: DBLP

CITATIONS
121

DOWNLOADS
198

VIEWS
84

3 AUTHORS, INCLUDING:

Bernhard Suhm
Raytheon BBN Technologies
39 PUBLICATIONS **759** CITATIONS

SEE PROFILE

Brad A. Myers
Carnegie Mellon University
431 PUBLICATIONS **13,500** CITATIONS

SEE PROFILE

Available from: Brad A. Myers
Retrieved on: 09 September 2015

Multimodal Error Correction for Speech User Interfaces

BERNHARD SUHM
BBN Technologies
BRAD MYERS
Carnegie Mellon University
and
ALEX WAIBEL
Carnegie Mellon University and Karlsruhe University (Germany)

Although commercial dictation systems and speech-enabled telephone voice user interfaces have become readily available, speech recognition errors remain a serious problem in the design and implementation of speech user interfaces. Previous work hypothesized that switching modality could speed up interactive correction of recognition errors. This article presents multimodal error correction methods that allow the user to correct recognition errors efficiently without keyboard input. Correction accuracy is maximized by novel recognition algorithms that use context information for recognizing correction input. Multimodal error correction is evaluated in the context of a prototype multimodal dictation system. The study shows that unimodal repair is less accurate than multimodal error correction. On a dictation task, multimodal correction is faster than unimodal correction by respeaking. The study also provides empirical evidence that system-initiated error correction (based on confidence measures) may not expedite error correction. Furthermore, the study suggests that recognition accuracy determines user choice between modalities: while users initially prefer speech, they learn to avoid ineffective correction modalities with experience. To extrapolate results from this user study, the article introduces a performance model of (recognition-based) multimodal interaction that predicts input speed including time needed for error correction. Applied to interactive error correction, the model predicts the impact of improvements in recognition technology on correction speeds, and the influence of recognition accuracy and correction method on the productivity of dictation systems. This model is a first step toward formalizing multimodal interaction.

This research was sponsored by the DARPA under the Department of Navy, Office of Naval Research under grant number N00014-93-1-0806. The views and conclusions in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the U.S. Government.
Authors' addresses: B. Suhm, Speech and Language Processing, BBN Technologies, 70 Fawcett Street, Cambridge, MA 02138; email: bsuhm@bbn.com; B. Myers, Human Computer Interaction Institute, School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213-3891; email: bam@cs.cmu.edu; A. Waibel, Interactive Systems Laboratories, School of Computer Science, Carnegie Mellon University and Karlsruhe University (Germany), Pittsburgh, PA 15221; email: ahw@cs.cmu.edu.
Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of the ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.
© 2001 ACM 1073-0516/01/0300–0060 $5.00

Categories and Subject Descriptors: H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Evaluation/methodology*; *Input devices and strategies*; *Interaction styles*; H.1.2 [**Models and Principles**]: User/Machine Systems—*Human factors*; H.5.3 [**Information Interfaces and Presentation**]: Group and Organization Interfaces—*Evaluation/methodology*; *Theory and models*; J.1 [**Computer Applications**]: Administrative Data Processing; J.3 [**Computer Applications**]: Life and Medical Sciences

General Terms: Design, Experimentation, Human Factors, Measurement

Additional Key Words and Phrases: Multimodal interfaces, interactive error correction, speech input, pen input, dictation systems, performance model, speech user interfaces

1. INTRODUCTION

Although speech user interfaces have begun to replace traditional interfaces (for example, in dictation systems and in speech-enabled automated call centers), speech recognition technology comes with inherent limitations, including poor performance in noisy environments and on unrestricted domains, restrictions on vocabulary which are difficult to convey to users, lack of toolkits to support application development, and recognition errors. Our research addresses the *repair problem* in speech user interfaces: how to correct the recognition errors which occur due to imperfect recognition. Although continuous speech dictation systems have been available commercially for two years, recent studies (e.g., Karat et al. [1999]) show that repair is still a significant problem. Assuming that continued progress in recognition algorithms will not completely eliminate recognition errors, our research investigates interactive error correction methods and presents *multimodal* error correction as a solution to the repair problem.

Usage of the term "multimodal" has been inconsistent in the field of multimodal user interfaces. By definition, "multimodal" should refer to using more than one modality, regardless of the nature of the modalities. However, many researchers use the term "multimodal" referring specifically to modalities that are commonly used in communication between people, such as speech, gestures, handwriting, and gaze. In this article, "multimodal" refers to more than one modality. The research presented in this article focuses on the modalities keyboard and mouse input, speech, gesture, and handwriting. Gesture and handwriting input by means of a pen on touch-sensitive displays is referred to as *pen input*.

Previous research has investigated multimodal error correction in a simulation study [Oviatt and VanGent 1996], and other work [Oviatt 1999] has shown that redundant speech and pen input can significantly increase interpretation accuracy, thus reducing the need for error correction. But no previous research has investigated the benefits of multimodal error correction in the context of a prototypical multimodal interface. This article empirically shows benefits of multimodal error correction in the context of a dictation task. The article presents multimodal correction methods and a prototype *multimodal dictation system,* which integrates multimodal error

correction with an automatic dictation recognizer. The article then describes a user study that compares unimodal correction by respeaking with several multimodal correction methods, including conventional multimodal correction by keyboard and mouse input. To extrapolate results from the user evaluation to future recognition performance, a preliminary model of multimodal recognition-based interaction is developed and applied to several important issues. Such performance models could evolve into useful tools for the design of future multimodal interfaces, which may reduce the need for costly empirical studies in exploring the trade-off between unimodal and multimodal interaction.

1.1 Previous Research on the Repair Problem

Martin and Welch [1980] introduced the concept of *interactive* correction for speech recognition errors. They proposed to store preliminary recognition results in a buffer and have the user interactively edit the buffer, by deleting single words, deleting the whole buffer, or repeating using speech (also called *respeaking*).

Since respeaking is the preferred repair strategy in human-human dialogue [Brinton et al. 1988], many speech user interface designers believe respeaking is the most intuitive interactive correction method (e.g., Robbe et al. [1996]). However, unlike in human-human dialogue, respeaking does not increase the likelihood that a speech recognizer correctly interprets the input. Ainsworth [1992] and Murray et al. [1992] suggest that the accuracy of respeaking could be increased by eliminating alternatives from the recognition vocabulary that are known to be incorrect ("repeating with elimination"). In addition, they introduced a second interactive correction method, *choosing from a list of alternative words*.

Baber and Hone [1993] discussed the problem of error correction in speech recognition applications in general terms. They pointed out that interactive correction consists of two phases: first an error must be detected; then it can be corrected. As a generalization of the concept "speech user interface," Rhyne and Wolf [1993] defined the term *recognition-based interface*: an interface that relies on imperfect recognition of user input. They were also the first researchers to discuss potential benefits of multiple modalities for error correction; switching to a different modality may help to avoid repeated errors. Oviatt and VanGent [1996] investigated multimodal error correction in a Wizard-of-Oz simulation study. Results suggested that users "naturally" switch modalities in error correction if given the possibility, alleviating user frustration in repeated failures. Another study [Oviatt et al. 1996] compared a GUI with a multimodal interface that supports simultaneous speech input and pen input. This study reported that total task completion time and error correction time is shorter for multimodal interaction.

McNair and Waibel [1994] implemented novel multimodal error correction methods: a method to select an error by voice and a method to interactively correct errors by either respeaking or spelling the misrecognized

words. Meanwhile, voice-selection of errors has become a standard feature in today's dictation systems. McNair's multimodal correction methods assumed that the correct word would be included in the list of alternative words returned by the recognizer for the original utterance. This is a severe limitation for most continuous speech applications, because the correct hypothesis may be far down or missing from the list of alternatives.

Karat et al. [1999] showed, for text creation tasks, that current commercial dictation systems are still significantly slower than traditional keyboard and mouse editing. Detailed analyses of users' error correction patterns revealed that the potential productivity gain of using speech dictation is lost during error correction. However, the study does not provide conclusive results about speech versus keyboard as correction modalities, because the two modalities were not separated and because correction speed was not measured. More recently, a longitudinal study by the same researchers [Karat et al. 2000] revealed that users can create text more efficiently with dictation systems than by typing, but only after extended exposure and learning time. Other recent work [Oviatt 1999] showed redundant multimodal, speech and pen input can significantly increase interpretation accuracy on a map interaction task. This work showed that multimodal interaction can help to avoid recognition errors, especially if foreign-accented speech deteriorates speech recognition accuracy, yet the repair problem was not addressed specifically in that study.

1.2 Evaluation of Speech User Interfaces

Baber and 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" [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 methodologies have predominated: user studies and modeling. Both have limitations, especially when applied to recognition-based interfaces. While providing rich data, results from usability tests with human participants and real speech recognition systems depend on the specific speech recognizer used, the task (vocabulary), and the participants (experience and training). Simulation studies may abstract from specific recognizers, but the error behavior of real recognition systems is very difficult to simulate. Model-based evaluation has the advantages of low cost, abstraction from implementation details, and the possibility to iterate design cycles quickly. But the validity of model predictions can be questionable because model assumptions may not apply to other situations.

This article argues that applying both model-based evaluation and empirical studies in complementary ways is a powerful methodology for

evaluating recognition-based multimodal interfaces. Lack of external validity of user studies can be overcome using predictions from model-based evaluation, and model-based predictions are more credible if the model is validated with data from user studies.

1.3 Outline

The article is divided into two parts. Sections 2–4 describe our implementation of multimodal correction. Sections 5 and 6 evaluate multimodal error correction by applying a user study and performance modeling as two complementary evaluation methodologies.

Section 2 presents a general multimodal repair algorithm, which is an abstraction of our previous description of multimodal interactive error correction [Suhm et al. 1996]. In a generalization of previously published analyses [Suhm et al. 1999], the current article provides evidence that unimodal repair in general, not only respeaking, is less accurate than multimodal repair. But recognizing (multimodal) corrections is challenging; recognition performance on correction input is substantially lower than the accuracy on standard benchmarks. To increase recognition accuracy on correction input, Section 3 presents algorithms that exploit information from the context of an interactive correction. While some of these algorithms have been described earlier [Suhm 1997], this article describes new algorithms, reevaluates the old algorithms on a more realistic database, and analyzes the statistical significance of the effects. Section 4 describes our prototype multimodal dictation system, including how we implemented system-initiated detection of recognition errors. The description of the prototype's system architecture and its usability problems, also included in Section 4, could be useful for designers of other multimodal applications.

The second part of this article—evaluation of multimodal error correction—consists of Sections 5 and 6. Section 5 describes the empirical evaluation of interactive error correction using the prototype multimodal dictation system. Section 6 describes our performance model of multimodal recognition-based interaction. These two sections significantly extend a previous publication [Suhm et al. 1999] by presenting new statistical analyses of the data and by describing the performance model in more detail. Furthermore, the current article presents empirical data which shows that system-initiated error detection does not expedite error correction. Finally, Section 7 summarizes the contributions of this article, and Section 8 concludes with implications of this research for future speech and multimodal user interfaces.

2. MULTIMODAL INTERACTIVE ERROR CORRECTION

This section and the following two sections describe the technology and implementation of multimodal interactive error correction. After presenting a general algorithm of multimodal repair in Section 2.1, subsequent sections describe multimodal correction methods: cross-modal correction by repeating and editing using pen gestures.

2.1 Multimodal Repair Algorithm

A multimodal interface that supports multimodal repair must include the following main components: recognition components (in particular, recognizers for continuous speech, spelled letters, handwriting, and pen gestures), components that capture user input and present the output to the user, and several modules to support integration, such as the dialogue manager, the correction algorithm module, and the application kernel. Figure 1 shows the flowchart of our multimodal repair algorithm, which is described in more detail below.

In interacting with a recognition-based multimodal interface, a user first provides *primary input* in some modality. In speech user interfaces, this modality is typically continuous speech. The primary user input is automatically interpreted using an adequate recognizer, in the flowchart denoted as *continuous recognition*. For example, in dictation applications, a large-vocabulary continuous speech recognizer interprets the dictation input. After primary user input has been recognized and processed, the application provides feedback to the user. This feedback may range from visual presentation of the recognition output (e.g., in dictation applications: displaying the recognition result on the screen) to execution of the action intended by the user (e.g., in an automatic flight booking system: retrieval of information on flights and visual or verbal presentation of the results). After the feedback phase is completed, it must be decided whether a recognition error has occurred ("Accept?" in the flowchart). This decision is made by either the system or the user. If the recognition is accepted, no repair is necessary, and user interaction with the application can proceed ("Repair Done" in the flowchart). If an error is detected, one or more repair interactions follow to recover from the error, until correction is successful.

For interactive correction of a recognition error, the exact location of the error within a larger sequence of input may have to be determined ("Locate Error" in the flowchart). After an error has been detected and located, the user chooses an appropriate multimodal correction method and provides the required correction input (e.g., spelling a misrecognized word). Before recognizing the correction input, the repair context is updated with the most recent primary user input, the recognition result, and information on the located error ("Update Repair Context" in the flowchart). This information may be used in recognizing the correction input and later in the correlation step. The correlation step selects the recognition output from appropriate recognizers, and it optionally applies algorithms to increase the likelihood of successful correction (such algorithms are described in Section 3). After selecting the final hypothesis, which optionally includes a correlation step, the system provides feedback on the completed correction attempt (the loop back to "Recognition Feedback" in the flowchart).

The present study explores only *sequential* multimodal interaction. It is unclear whether *simultaneous* use of several modalities may improve error correction. A simulation study [Oviatt et al. 1997] suggested that simultaneous use of modalities is frequent for spatial location commands, but

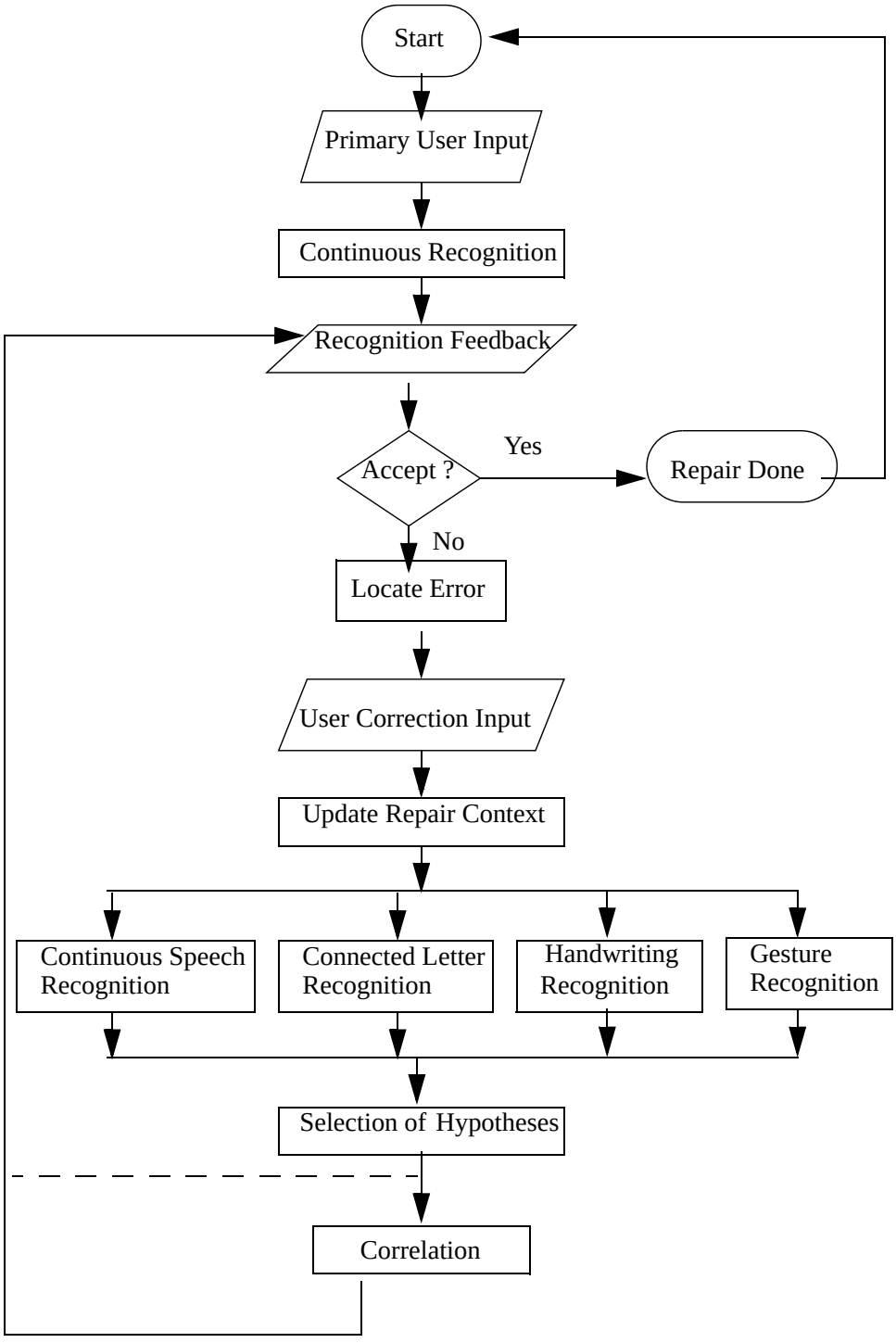

Fig. 1. Flowchart of multimodal repair algorithm.

Table I. Database of Multimodal Corrections

Type of User Input	Items in Database
Initial dictation	503 sentences (9750 words)
Respeaking (multiple words possible)	515 repairs (1778 words)
Spelling	816 words
Handwriting	1301 words
Choose from list of alternatives	478 repairs
Pen-based gestures	747 repairs
Editing with mouse/keyboard	431 repairs

rather infrequent in general action commands. Future work needs to investigate whether error correction can benefit from simultaneous multimodal interaction.

The flowchart in Figure 1 provides an overview of multimodal error correction. But how can errors be located and interactively corrected? Current methods to interactively locate recognition errors include user-initiated and system-initiated methods. The user can detect and locate errors by pointing, using voice commands, or applying conversational techniques. Pointing is natural and effective if the application permits visual feedback. Voice commands to select errors are already available in commercial dictation systems. Conversational techniques to detect errors build on research on repair in human-human dialog. People frequently paraphrase in dialogs and use certain trigger phrases when they notice communication problems. But interpreting such conversational cues automatically is more challenging than recognizing the initial speech input, and beyond the capabilities of current technology. (For more details on conversational repair, see Appendix B in Suhm [1998]). This article focuses on correction methods that use pointing to detect errors, because such methods can be successfully realized with today's technology.

This article evaluates multimodal error correction methods using a database of multimodal corrections, shown in Table I. Our database was collected during the user studies of the prototype multimodal dictation system, which are described in detail in Section 5. For the analyses below, the data were pooled across all 15 participants, and only the data on initial dictation and correction by respeaking, spelling, or handwriting are used. Note, that among these correction modalities, only respeaking allows the user to correct more than one word at a time. On this dataset, users spoke an average of 3.5 words per correction.

2.2 Correction by Cross-Modal Repeating

Repeating input is a very simple and intuitive correction method. In fact, there is evidence that repetition is the preferred correction method in human-human dialogue [Brinton et al. 1988]. Although very effective in human-human dialogue, repeating input in the same modality decreases the chances of success of repair in recognition-based interfaces, because repeating does not eliminate the cause of recognition errors—deficiencies

Fig. 2. Cross-modal insertion using handwriting.

in recognition models. Moreover, when the primary user input is spoken, the tendency to hyperarticulate in spoken repairs deteriorates recognition accuracy rather than increasing it [Oviatt et al. 1996]. Hyperarticulation increases the mismatch between spoken correction input and the acoustic models of the speech recognizer, which are trained only on normally pronounced speech. For that reason, correction by repeating in the same modality frequently leads to repeated errors.

This article examines two approaches to make correction by repetition effective: switching modality for repetitions (in *cross-modal* repetitions), and correlating the correction input with repair context. In cross-modal repair, the user corrects with a different modality than used for the primary input. For example, assuming the primary input was continuous speech, the user may switch to spelling verbally or to handwriting. Figure 2 illustrates cross-modal inserting using handwriting.

To show that unimodal repetition is ineffective (not only for speech, but also other modalities) and that cross-modal repetition is an effective correction strategy, Figure 3 plots correction accuracies for consecutive correction attempts *in the same modality*. The original input was dictated using continuous speech and automatically recognized at 75% word accuracy. Note, that in this context, respeaking is a "unimodal" correction method (because the initial input was dictated), but spelling and handwriting are multimodal methods. A two-way ANOVA shows a significant effect for the factor correction attempt ($F = 26.2$; $df = 2, 4$; $p < 0.01$). Post hoc Scheffé comparisons confirm that the second correction attempt (in the same modality) is significantly less accurate than the first ($t = 3.63$, $p < 0.05$, one-tailed), and that cross-modal repetition by spelling or handwriting is more accurate than (unimodal) repetition by respeaking ($t = 2.92$, $p < 0.05$, one-tailed).

Recognition accuracy on cross-modal corrections is still much lower than the reported accuracy on standard benchmarks. For the recognizers used in this research, the developers report more than 90% recognition accuracy for similar vocabulary sizes [Hild 1997; Manke 1998]. We can explain this discrepancy, since correction input is more difficult to recognize than benchmark data. This is because short words are more frequent in correction input, and short words are more difficult to recognize than long words if recognition is limited to a vocabulary. For more details on this phenomenon see Suhm [1998].

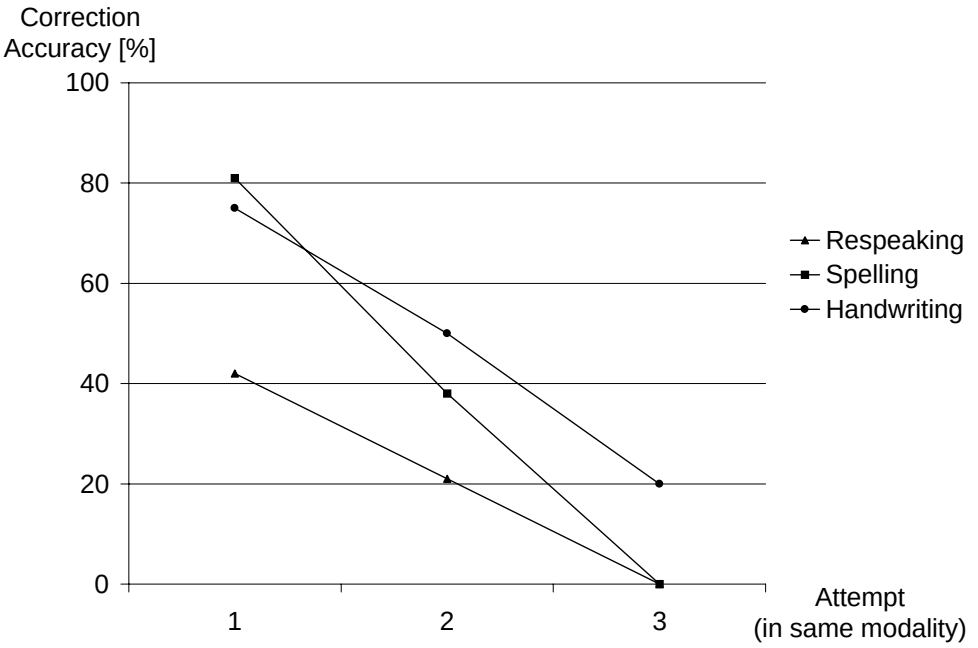

Fig. 3. Deterioration of correction accuracy for repetitions in the same modality.

2.3 Editing Using Pen-Based Gestures

Correction by repeating addresses two correction tasks, substitution and insertion repairs, but a complete set of correction methods must include methods to delete and edit as well. Previous research systems and commercial dictation systems offer voice editing or editing by mouse and keyboard input. Our research investigates using *pen gestures* for editing tasks. Recently, other researchers have also begun to explore pen gestures for editing in dictation systems [Vergo 1999].

Editing tasks 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. Previous research suggests that pen gestures are intuitive and efficient for such command control tasks [Rhyne and Wolf 1993; Wolf and Morrel-Samuels 1987]. Figure 4 illustrates the pen gestures used in our multimodal dictation system, including gestures for deleting, positioning the cursor, and selecting input items at different input levels (phrases, words, and characters within a word).

Ultimately, the choice between voice, keyboard/mouse, and pen gesture editing is one of the design decisions of a multimodal user interface. Pen gestures are attractive for applications that naturally include a graphic user interface and where a pen (rather than a mouse) is available, such as dictation and data-entry applications.

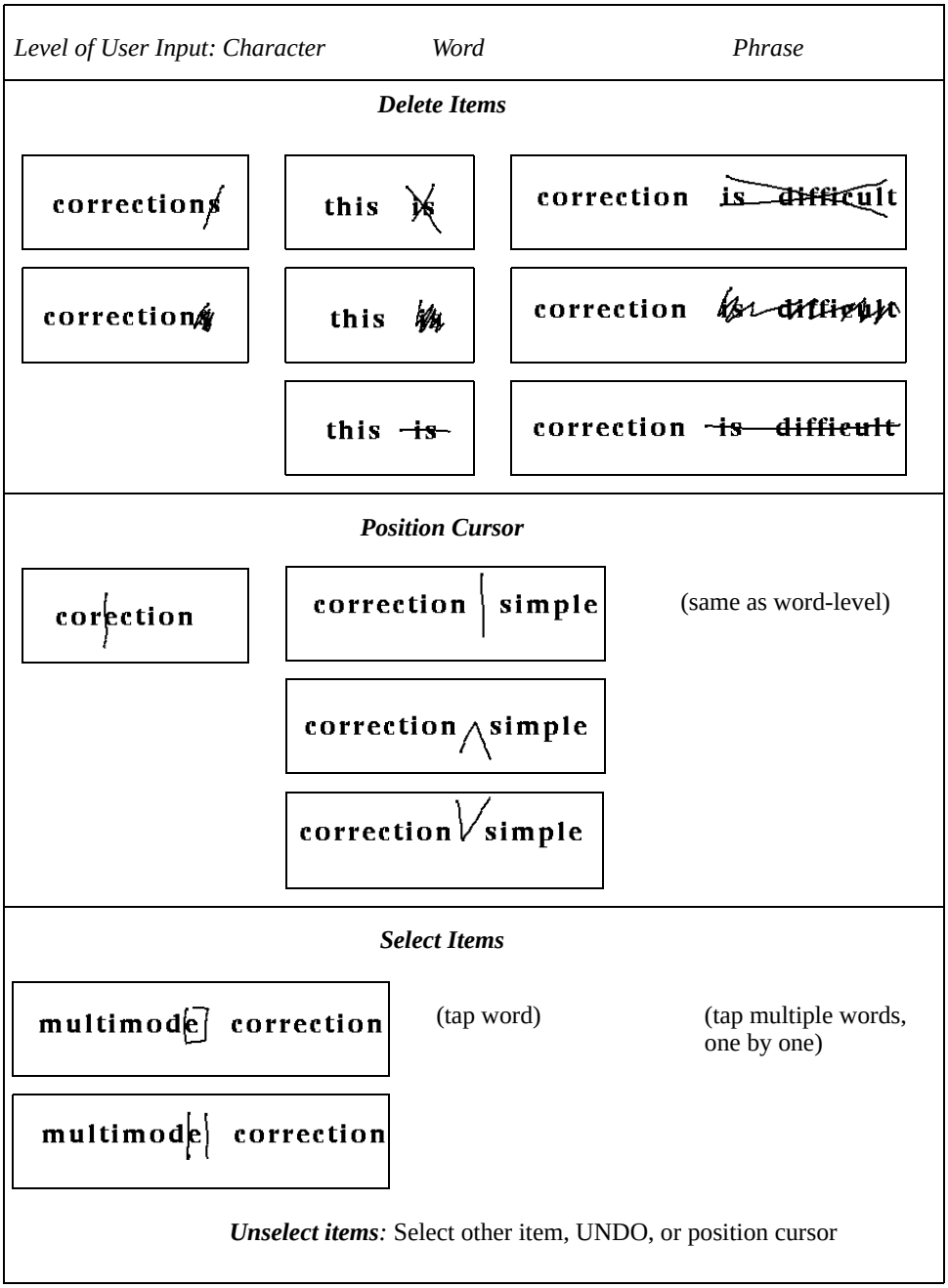

Fig. 4. Editing using pen-based gestures.

3. INCREASING CORRECTION ACCURACY USING CONTEXT INFORMATION

Interactive error correction is effective if the modality is switched for correction, although recognizing correction input is less accurate, compared with standard recognition benchmarks. To further increase the effectiveness of correction by repeating, correction input should not be processed as an independent event. Context information can further constrain recognition of correction input. The simplest way to exploit context information, mentioned earlier in Section 1.1, is eliminating alternatives from the recognition vocabulary that are known to be incorrect [Ainsworth 1992]. This section proposes two more powerful algorithms that improve correction accuracy: *N*-gram context modeling and bias toward frequently misrecognized words. Both methods have in common that the correction input is correlated with the repair context. The effectiveness of these algorithms is demonstrated on the database of multimodal corrections mentioned earlier.

3.1 *N*-Gram Context Modeling

Context modeling exploits the observation that once an error has been located, the surrounding input is probably correct. Recognizing correction input should then enforce the same context constraints as used in recognizing primary (continuous) input. This section describes context modeling for word-level correction input and dependencies modeled as statistical *N*-gram language models. But the idea can be applied to other types of input (e.g., digit sequences) and other formalizations of dependencies between input items (e.g., finite-state grammars).

Statistical language models determine the probabilities of word sequences. Widely applied in the speech recognition field, an *N*-gram language model factors the joint probability of a word sequence into a product of conditional probabilities: $P(w_1 \ldots w_L) = \prod_i^L P(w_i | w_{i-N+1} \ldots w_{i-1})$ (cf. Jelinek [1990]).

Corrections that replace an error region by 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 $w_{i \ldots} w_{i+M}$, the word context to the left of it as $w_{i-2} w_{i-1}$, and the word context to the right as $w_{i+M+1} w_{i+M+2}$. User input intended to correct the reparandum is denoted as $v_1 \ldots v_k$. In the notation of this article, alternatives for the same input are indexed with a superscript, and the items of sequences of input with a subscript.

With *N*-gram context modeling, the recognition system processes correction input as if it occurred in the context of the *precontext* $w_{i-2} w_{i-1}$ and the *postcontext* $w_{i+M+1} w_{i+M+2}$, applying the appropriate language model constraints as illustrated in Figure 5.

The implementation of the context-modeling function $F[P(Repair | w_{i-2}, w_{i-1}) P(w_{i+M+1} | Repair)]$ depends on the recognizer.

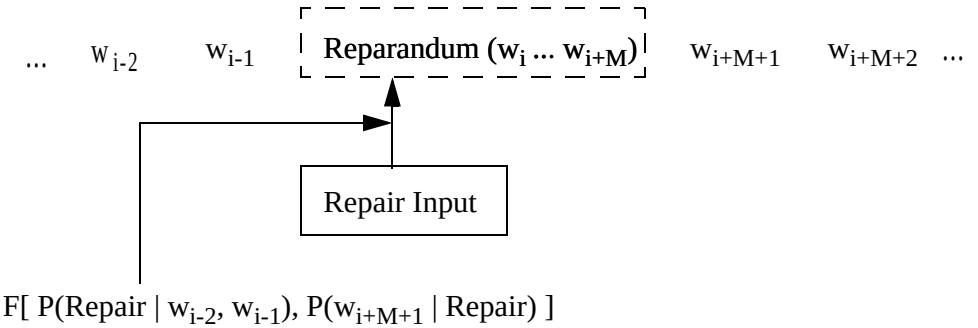

Fig. 5. *N*-gram context modeling.

For continuous input recognizers that use a statistical language model, context modeling is implemented by replacing the neutral language model context at the beginning and end of an utterance by the appropriate pre- and postcontext, e.g., $P(v_1|w_{i-2}, w_{i-1})$, instead of the neutral $P(v_1|\langle s\rangle)$, where $\langle s\rangle$ denotes the beginning of a sentence. For isolated-word recognizers, a rescoring algorithm can be applied: after interpreting isolated-word correction input in the usual way (as an independent event), the K-best list of hypotheses for the repair $\{v^1, \ldots, v^K\}$ is reordered based on a combination of recognition and context-modeling scores. Equation (1) defines the context-modeling score $CS(k)$ for the kth alternative hypothesis and a trigram language model. This equation formalizes conforcing the appropriate context constraints (from words w_j) on the language-modeling score for the kth hypothesis v^k of the correction input.

$$CS(k) = P(v^k|w_{i-2}, w_{i-1}) \cdot P(w_{i+M+1}|w_{i-1}, v^k) \cdot P(w_{i+M+2}|v^k, w_{i+M+1}) \quad (1)$$

Context modeling scores for isolated work repairs and a trigram language model.

N-gram context modeling was evaluated on data from our multimodal correction database. Table II shows the performance of *N*-gram context modeling for corrections by repeating in continuous speech, spelling, and handwriting. In addition, Table II compares the performance of context modeling, either using only the precontext or using both pre- and postcontext. A two-way ANOVA shows a significant effect for context modeling ($F = 881.7$; $df = 2, 4$; $p < 0.01$). A planned comparison confirms that precontext modeling significantly increases correction accuracy, compared to the baseline ($t = 11.63$; $df = 6$; $p < 0.01$, one-tailed). However, there is no significant increase of pre- and postcontext modeling over precontext modeling alone. It may be surprising that using more context does not consistently improve accuracy, but users do not consistently select maximally contiguous regions of errors; and thus the postcontext is frequently incorrect.

Table II. Increase of Correction Accuracy by *N*-Gram Word Context Modeling

Experiment Condition	Respeaking	Spelling	Handwriting
Baseline (no context modeling)	43%	73%	67%
Pre context modeling	53%	80%	75%
Pre and post context modeling	52%	81%	74%

In summary, context modeling is a very effective way to increase the accuracy of interactive correction, both for cross-modal corrections and for corrections in the same modality.

3.2 Bias Toward Frequently Misrecognized Words

As a second method to correlate correction input with repair context, this section proposes to bias recognition of repair input toward frequently misrecognized words. Biasing correction input toward frequent recognition errors exploits the fact that errors are not randomly distributed but that, within one input modality, certain words are more frequently misrecognized than others. In first-order approximation, the error behavior of a recognizer for modality m 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? If the recognizer employs a language model, the language model score computation is modified to compute the probability that a word is correct by adding a weighted unigram bias that w is correct $P(correct|w)^{\mu}$, as illustrated in Figure 6. A weight parameter μ determines how the regular language model and bias should be balanced. The value for μ can be determined empirically, by maximizing correction accuracy on a cross-validation set of correction data.

If the recognizer does not utilize a language model, a rescoring technique can be employed. For each alternative hypothesis v^k (obtained by recognizing the correction input as an independent event), a bias score $B(k) = \log P(\text{correct}|\text{v}^{\text{k}})$ is computed. 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.

Table III evaluates the performance increase for applying the bias across different modalities. In all cases, the bias was applied in addition to precontext modeling. For the continuous speech and spelling modalities, the bias was integrated with a language model; for the handwriting modality, the bias was implemented as an additional rescoring pass. While Table III shows a relative increase in correction accuracy with the bias, a planned comparison fails to confirm statistical significance ($t = 1.94$; $df = 4$; $0.05 < p < 0.1$). However, the bias does significantly increase accuracy of multimodal correction by spelling and handwriting.

$$Score(w, A) = \log P(w|A)P(w)P(correct|w)^{\alpha}$$

Acoustic Model | Language Model | Bias

Fig. 6. Extending word scores by a bias toward frequent errors. A denotes the current correction input and w the hypothesized word.

Table III. Increase of Correction Accuracy by Biasing Toward Frequently Misrecognized Words

Experiment Condition	Respeaking	Spelling	Handwriting
Without bias	53%	80%	75%
With bias	53%	84%	78%

4. A PROTOTYPE MULTIMODAL DICTATION SYSTEM

A *multimodal* dictation system is a standard dictation system that supports multimodal input for both original input and error correction. To build a prototype multimodal dictation system, we integrated multimodal error correction with the state-of-the-art JANUS large vocabulary dictation recognizer [Rogina and Waibel 1995]. The next subsection presents methods for locating and correcting recognition errors. The second subsection discusses various user interface problems in the design of a multimodal dictation system and how they were addressed in our prototype. The final subsection describes the prototype's system architecture, which could be applied to other multimodal recognition-based applications.

4.1 Locating Recognition Errors

Two methods for locating recognition errors were implemented in the prototype, both a user-initiated and system-initiated method. For user-initiated error detection, the user looks at the recognition result, which is displayed on a touch-sensitive screen, and selects recognition errors by tapping on words. For system-initiated error detection likely recognition errors are highlighted.

Voice editing is an attractive user-initiated method for locating recognition errors, and surveys of commercial dictation systems suggest that users like voice-editing capabilities. However, a recent study [Karat et al. 1999] revealed that voice editing as implemented in current dictation systems introduces severe usability problems, and significantly slows down error correction. Choosing the method for locating recognition errors is thus another design decision that depends on the application. We decided to evaluate editing using pen gestures (as illustrated in Figure 4 earlier in this article).

Confidence scores can be used to identify likely recognition errors by applying a threshold criterion [Chase 1997]. Words with low confidence scores are tagged as possible recognition errors. Since confidence scores

themselves are not reliable, these tags may be incorrect. More specifically, misrecognized words may be tagged as "correct" (i.e., missed detections of recognition errors), and correct words may be tagged as recognition errors (i.e., false alarms). Hence, an automatic method for highlighting errors based on imperfect confidence scores must balance missed detections and false alarms. In the prototype, confidence scores were computed using the Gamma feature [Kemp and Schaaf 1997]. The threshold on the confidence score was tuned to minimize the total classification errors, i.e., the sum of missed detections and false alarms. A classification accuracy of 89% was achieved using a threshold of 0.6.

4.2 Usability Issues in a Multimodal Dictation System

This section describes usability problems that we encountered while developing our multimodal dictation system. Usability problems occurred in two general areas: triggering input and distinguishing correction modalities. Table IV lists the most important problems, the different designs that were tried, and the usability problems of each design. Although there are problems with most designs, our informal usability tests showed the trade-off that seems to work best. Designers of future multimodal user interfaces will probably face similar problems and may benefit from this discussion.

To correct by choosing from alternatives, we tried a design that leveraged user experience with pull-down menus in traditional GUIs. To display a menu with the list of alternatives for a recognized word, the user could tap on that word (either with a pen or a finger) and move downward on the screen, as if to pull down a menu. However, the pull-down gesture confused many users in our informal tests, and the gesture recognizer frequently misinterpreted the gesture to delete the word. A better design is to display the list of alternatives (as a pop-up menu) after touching a word for approximately one second.

To distinguish between the two speech modalities (continuous speech and spelling), the prototype's user interface contains two buttons, one for continuous speech and one for spelling. Automatic classification of the speech input in continuous speech or spelling is the desired solution, but with current technology, it was not accurate enough on large vocabulary tasks.

Pen input leads to a similar design problem: the system must distinguish between handwriting and pen gestures. An early design introduced a separate button to switch to handwriting mode. This design led to mode errors, and the button cluttered the interface. However, pen input can be automatically classified into gesture versus handwriting using a combination of the Mahalanobis distance [Rubine 1991] and application-specific heuristics (for more details see Suhm [1998]).

Finally, determining the end of pen and speech input leads to usability problems. In an early design, the user pressed a button to initiate automatic recognition of the most recent input. However, users forget to press

Table IV. 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 for a long time	Either slows down interaction 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 error
	One button for continuous speech, tap selection to trigger spelling	Mode errors
Distinguish handwriting from pen gestures	Interpret pen input as gesture by default, button to switch to handwriting mode	Mode errors; button clutters interface
	Classify automatically	Sufficiently accurate
End criterion for user input	Time-out	Time-out has to be adapted to user, and slows down interaction
	Button to launch recognition	Users forget to press button

the button. A better design is launching recognition automatically after a sufficiently long time-out. Approximately one second worked well in our informal tests, and did not significantly slow down interaction.

4.3 System Architecture for Multimodal Recognition-Based Interfaces

The multimodal dictation system employs the client-server architecture shown in Figure 7. Two important issues are addressed: first, input capture and system feedback are separated from all processing; second, processing of multimodal input is implemented as a server that delegates recognition to the appropriate subsystems. Both ideas are well known within the software engineering and user interface communities. The first idea ensures that the application's user interface can run in heterogeneous computing environments, for example, an X windows environment and a Web browser. By applying the second idea, the heavy computational burden on the application's back-end (needed for the automatic recognition of multiple

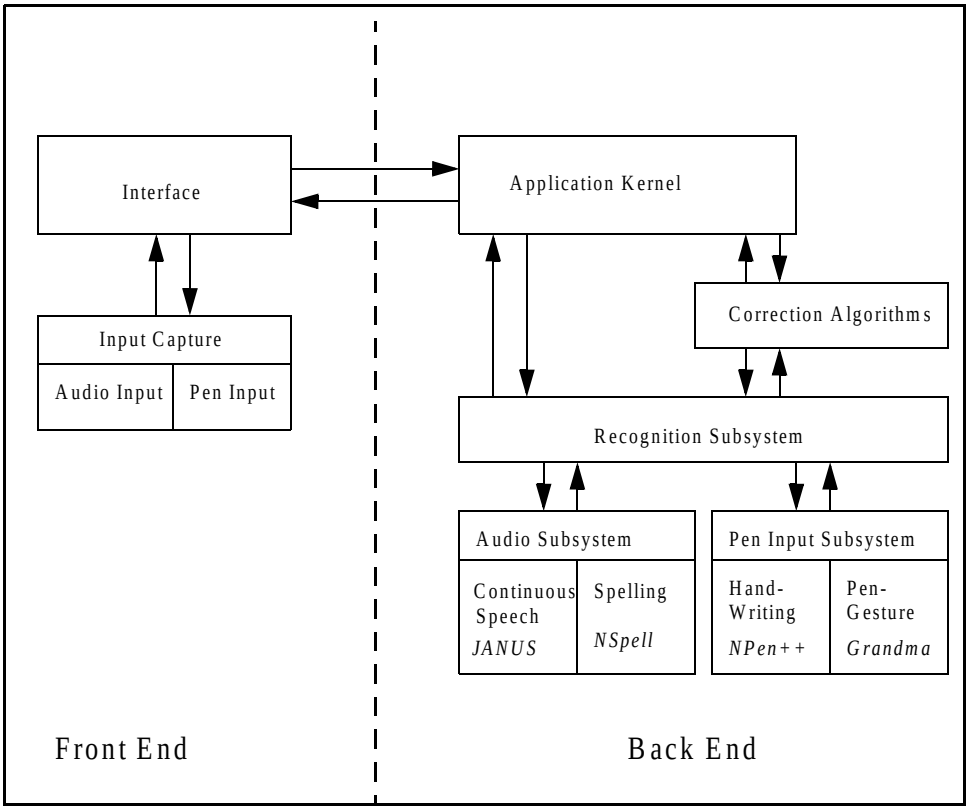

Fig. 7. System architecture to integrate multimodal correction in speech user interfaces.

modalities) can be distributed among several powerful server hosts. The module "Correction Algorithms" encapsulates the error correction functionality, keeping it separate from all application-specific functionality (summarized as "Application Kernel" module). Both these modules delegate recognition to the recognition subsystems ("Audio Subsystem" and "Pen Input Subsystem"). "Recognition Subsystems" denotes a communication layer for uniform access to all recognizers. This architecture is similar to the one Vo proposed for a multimodal application toolkit [Vo 1998]. Future work might extend such toolkits to support multimodal error correction (e.g., Mankoff et al. [2000]).

Figure 7 also shows which recognition systems the prototype employs: for continuous speech, the JANUS speech recognizer, trained on the Wall Street Journal task [Rogina and Waibel 1995]; for spelled speech, the NSpell connected letter recognizer [Hild 1997]; for handwriting, the NPen on-line cursive handwriting recognizer [Manke et al. 1995]; for pen gestures, the gesture recognizer of the Grandma system [Rubine 1991]. All recognizers (except for the gesture recognizer) use the same standard 20,000–word Wall Street Journal vocabulary.

5. USER EVALUATION OF MULTIMODAL ERROR CORRECTION

Moving beyond technology to human factors issues, the remaining sections of this article evaluate error correction by applying user studies and modeling techniques in complementary ways. This section presents our user evaluation of interactive multimodal error correction in the context of the prototype multimodal dictation system. Unimodal correction by respeaking is compared with novel and conventional multimodal correction methods.

5.1 Experimental Design

5.1.1 *Hypotheses and Experimental Conditions*. The study was designed to address the following three research hypotheses:

(1) Multimodal correction is faster than unimodal correction.

(2) System-initiated error detection (based on confidence measures) expedites error correction.

(3) Users prefer the most accurate modality and quickly learn which modality works best for them.

Experimental conditions were chosen to allow us to quantitatively analyze each of these hypotheses. Current commercial dictation systems offer correction by respeaking, choosing from alternatives, or typing, which we call "convential correction" henceforth. Experimental conditions compared these conventional correction methods with novel multimodal methods. Note that in the context of a dictation application, correction by respeaking is "unimodal," and correction using modalities other than (continuous) speech is "multimodal." Since all conditions allowed the user to correct by choosing from alternatives, this modality will not be mentioned explicitly in every case.

Table V shows which correction modalities (shown as rows) are available in each of the four experimental conditions ("Keyboard & Mouse," "Respeak," "Speech & Pen Input," and "Speech & Pen Input, system highlights likely errors"). To evaluate the impact of typing skill, (conventional multimodal) correction using keyboard and mouse input is considered separately from (conventional unimodal) respeaking ("Keyboard & Mouse" versus "Respeak" in Table V). To evaluate the effectiveness of automatic highlighting of likely errors, "Speech & Pen Input" is contrasted with "Speech & Pen Input, system highlights likely errors." Throughout this section, we refer to modalities either in isolation (such as respeaking, handwriting, pen gestures), or grouped as correction methods (such as "Keyboard & Mouse," "Speech & Pen Input").

5.1.2 *Tasks*. Participants read aloud one or more sentences, which were chosen from newspaper text. On average, participants dictated 19.4 words to the system in one utterance (cf. Table I). After reading a sentence, the recognized words were displayed on the touch-sensitive screen. Then,

Table V. Experimental Conditions (shown as columns) and Available Correction Methods (shown as rows)

Experiment Condition	Keyboard & Mouse	Respeak	Speech & Pen Input	Speech & Pen Input, system highlights likely errors
Choose from list of alternatives	X	X	X	X
Respeaking		X	X	X
Spelling			X	X
Handwriting			X	X
Typing/mousing	X			
Pen-based gestures		X	X	X
Imperfect highlighting of errors				X

participants visually located recognition errors, selected them by tapping on the screen, and corrected them using one of the available correction methods. Experimental conditions differed in which methods were available for correcting the recognition errors, as shown in Table V. Participants were instructed to correct all errors as quickly as possible.

5.1.3 *Study Design and Participants*. To minimize the impact of the known high variation of recognition accuracy across users, a within-subject (repeated measures) design was chosen. The order of the four experimental conditions was randomized using a Latin Square, to eliminate order effects.

Fifteen participants were recruited from the local campus community, balanced across gender and the three categories of typing skill, but very fast and very slow typists were not represented. Participants included students and administrative staff, and most participants did not have any prior experience with speech-recognition software.

5.1.4 *Procedure*. Participants first completed a typing test in order to assign them to one of the three categories of typing skill: slow, average, and fast typists. Participants then learned to use the different correction modalities in a 45-to-60-minute tutorial and practice session. After this session, all participants showed sufficient familiarity with the different correction methods on trial tasks. The participants then proceeded to the experimental sessions. Different sets of sentences and the conditions were randomly assigned to the experimental sessions, to avoid order effects. After completing the experimental sessions, participants filled out a postexperimental questionnaire.

During experimental sessions, data were collected in two ways: timestamped records of all user interactions with the multimodal dictation system and video-tapings of all sessions. The timestamped records were manually annotated with the correct system response for each interaction, to assess interpretation accuracy. The record also contains the sequences of modalities used until successful correction. These sequences were analyzed for modality choice patterns. Additionally, we built a database of multimodal

corrections using the collected data. This database has been introduced earlier (cf. Table I).

5.1.5 *Experimental Variables*. Performance at the level of a single input modality was measured using the following three measures: *input rate* (i.e., how many words can a user enter per minute), *system response time* (i.e., how much time does automatic recognition require), and *recognition accuracy* (i.e., the probability of recognizing a word correctly). To distinguish between initial input and correction input, this article uses the term *correction accuracy* to refer to recognition accuracy on correction input.

Correction modality was the main independent variable. Additionally, the input rate, system response time, and overhead time are independent variables for some analyses.

To assess performance at the task-level, the following two measures were defined as the main dependent variables. *Correction speed* is the average number of words that can be successfully corrected per minute, including multiple correction attempts when necessary. For example, a correction speed of 6 cwpm (corrected words per minute) means that a user spends on the average 10 seconds to correct a recognition error. The second task-level measure, *text creation speed* is defined as the average number of words that can be successfully entered per minute, *including the time necessary for the correction of recognition errors*. Like correction speed, the text creation speed is measured in cwpm (correct words per minute, cf. Karat et al. [1999]). In addition to correction speed and text creation speed, some analyses use correction accuracy and usage frequency as dependent variables.

5.1.6 *Experimental Setup*. The prototype multimodal dictation system captures speech and pen input as follows. Similar to commercial dictation systems, the user speaks into a close-talking microphone (e.g., a headset). By using two different push-to-talk buttons, the user indicates the type of speech input: the "Dictate/Respeak" button for dictating whole sentence or correcting by respeaking, and the "Spell" button for (spoken) spelling input. For pen input, the user writes with a pen on a touch-sensitive screen. Figure 8 below shows the prototype's GUI.

Users select recognition errors either by tapping on incorrect words or by pressing on the "Select Next Error" button, shown in Figure 8. This button is available only in the experimental condition "Speech & Pen Input, system highlights likely errors"; it is not visible in any other experimental condition. When the "Select Next Error" button is pressed, the system selects the next region of likely recognition errors after the current selection or the position of the insertion cursor.

5.2 Results

5.2.1 *Comparing Unimodal with Multimodal Correction*. Table VI shows average correction speed for conventional unimodal correction ("Respeak"), conventional multimodal correction by keyboard and mouse ("Keyboard & Mouse"), and novel multimodal correction ("Speech & Pen Input").

Fig. 8. GUI of the prototype multimodal dictation system.

Note that higher correction speeds are better. A repeated measures ANOVA indicates a significant effect for correction speed ($F = 46.3$; $df = 3, 42$; $p < 0.01$). Repeated measures post hoc Scheffé comparisons confirm that multimodal correction is faster than conventional, unimodal correction by respeaking ($F = 51.8$; $df = 3, 14$; $p < 0.01$, one-tailed). The comparison among multimodal correction methods, including correction by keyboard and mouse input, depends on the user's typing skill. On average, across all participants in the study, correction by speech and pen input is significantly slower than correction by keyboard and mouse input ($F = 70.0$; $df = 3, 14$; $p < 0.01$, two-tailed).

To compare the various correction modalities by correction accuracy as dependent variable, Table VII shows the accuracy for the various correction modalities: choosing from the list of alternatives ("Choose from Alternatives"), respeaking, spelling, handwriting, and typing.[1] The correction

[1]The correction accuracies reported in this table are slightly different from the ones reported earlier in Table III, because they represent an average of estimates derived for each participant individually, while the former were estimated by pooling data across all participants (as customary in the speech recognition community).

Table VI. Average Correction Speeds

Experiment Condition	Correction Speech [cwpm]
Keyboard & Mouse	6.6
Speech & Pen Input	4.8
Speech & Pen Input, system highlights likely errors	4.1
Respeak	2.7

accuracy of 24% for "Choose List" means that the correct word was among the top six choices about every forth time. A repeated measures ANOVA reveals a significant effect for correction accuracy ($F = 61.5$; $df = 4, 60$; $p < 0.05$). Almost all pairwise post hoc comparisons are significant (Tukey $HSD = 15\%$; $df = 5, 60$; $p < 0.05$, two-tailed), except for the pairs Choose List–Respeaking, Handwriting–Spelling, Typing–Handwriting, and Typing–Spelling.

5.2.2 *Effectiveness of System-Initiated Error Detection*. The experiment conditions "Speech & Pen Input" and "Speech & Pen Input, system highlights likely errors" allow us to evaluate the effectiveness of system-initiated detection of recognition errors. System-initiated error detection was implemented by highlighting words with low confidence scores as likely recognition errors, as described earlier in Section 4.1. Since confidence scores themselves are unreliable, it was not clear whether such system-initiated detection of recognition errors would expedite error correction.

Table VI suggests that system-initiated detection of recognition errors, at least in this implementation, slows down correction. Post hoc comparisons reveal that the difference in correction speed between "Speech & Pen Input" and "Speech & Pen Input, system highlights likely errors" is not significant (Tukey $HSD = 1.2\,cwpm$, $df = 4, 14; p > 0.05$). This result obviously depends on the recognition technology, in particular, the reliability of confidence scores. If system-initiated detection of recognition errors was perfect, correction speed would increase. With 89% tagging accuracy, as achieved by this implementation of confidence scores, automatically highlighting likely recognition errors hurts overall performance. Future research will have to show whether confidence score algorithms can be designed that are reliable enough to realize a gain with system-initiated error detection.

5.2.3 *User Preferences between Modalities*. To perform a longitudinal analysis of modality user prefererence, the data for the two experiment conditions with speech and pen input were pooled, and relative modality usage frequencies were estimated every 40 correction interactions. Since (multimodal) correction by keyboard and mouse input, was a separate experiment condition, this study did not allow us to formally evaluate preferences relative to typing.

Table VII. (Average) Correction Accuracies

Modality	Correction Accuracy
Choose from alternatives	24%
Respeaking	35%
Handwriting	75%
Spelling	82%
Typing	87%

Table VIII. (Average) Usage Frequencies of Modalities

Modality	Usage Frequency
Spelling	0.14
Choose from alternatives	0.21
Respeaking	0.28
Handwriting	0.35

One time unit (in Figures 9–11) represents each a set of 40 interactions, which corresponds to about 20 minutes. This measure for the x-axis was chosen because the reliability of relative usage frequency estimates depends on the number of interactions considered, which can vary largely within equal time intervals.

A two-way ANOVA, based on usage frequency estimates for each participant and "time" interval, indicates signficant differences in usage frequency across modalities ($F = 22.5$; $df = 3{,}168$; $p < 0.05$), but no significant effect for time ($F = 0.03$; $df = 2{,}168$). As can be seen from the average frequencies in Table VIII, handwriting is used most for correcting errors, followed by respeaking, then choosing from alternatives, and spelling is used least often. Post hoc comparisons reveal that all pairwise comparisons between modality usage frequencies are significant (Tukey $HSD = 0.07$; $df = 4{,}168$; $p < 0.05$, two-tailed). While usage frequencies differ between modalities, the lack of a time effect might suggest that no learning occurs.

However, a closer look at individual users reveals that modality choice does change over time. Figure 9 shows the usage frequencies for two users. The most accurate correction modality differed for these users; it was handwriting for the user in the upper part of the figure, and spelling for the user in the lower part. With experience, both users avoid less accurate modalities in favor of more accurate modalities.

To examine whether correction accuracy determines modality choice, the correlation between usage frequency and *relative* correction accuracy was calculated and tabulated over time, as shown in Figure 10. A positive correlation indicates that users prefer more accurate modalities. A standard test reveals that the correlation for spelling ($t = 2.45$; $df = 13$; $p < 0.05$, one-tailed) and for handwriting ($t = 1.46$; $df = 13$; $p < 0.1$, one-tailed), at time "3", are significantly positive. However, planned

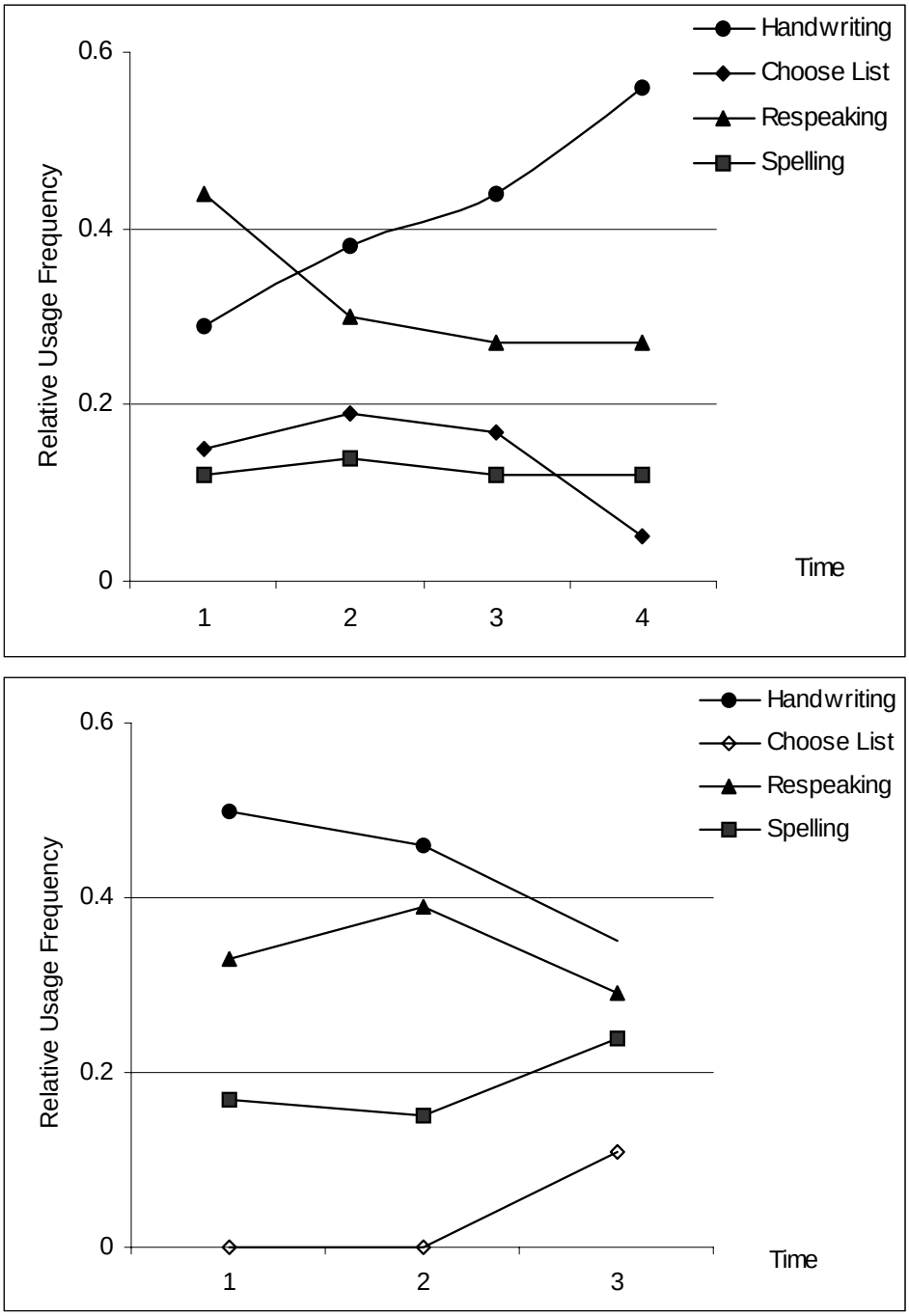

Fig. 9. Usage frequencies of different modalities for two typical users. The time axis corresponds to progressive points in time in the course of the experiment. The upper user learns to avoid speech and spelling and favors handwriting; the lower user learns to use respeaking and handwriting less in favor of spelling. These trends correspond to favoring more accurate modalities.

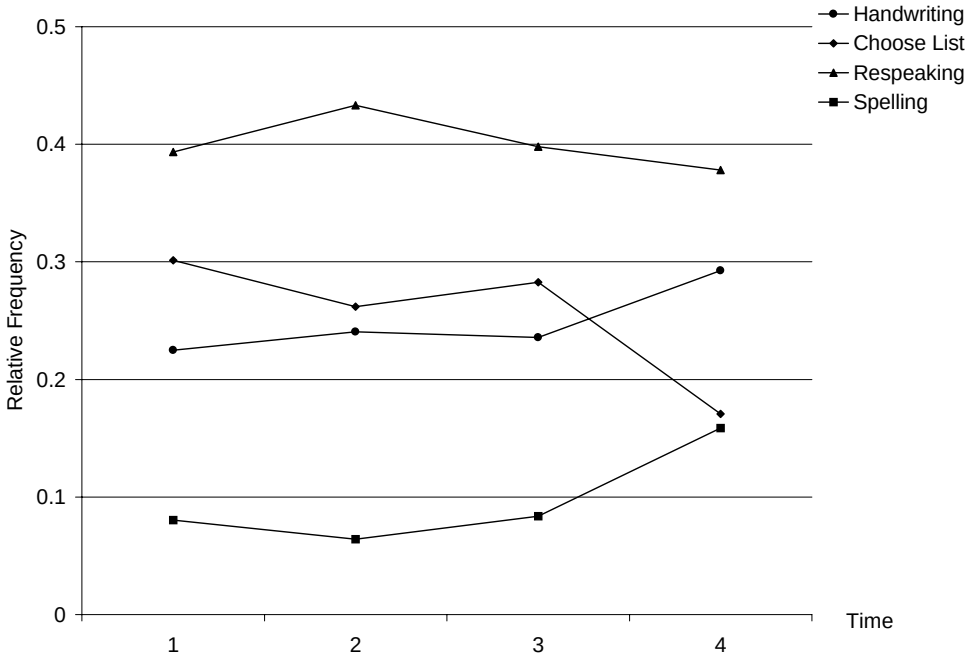

Fig. 11. Modality choice in the first correction attempt. Users initially prefer respeaking, and choose other modalities only after repeated evidence for the ineffectiveness of respeaking.

comparisons do not confirm a significant increase in correlation over time ($t = 1.75$; $df = 9$; $p > 0.1$, one-tailed).

Still, the correlation between usage frequency and correction accuracy becomes more positive with experience. The lack of a significant increase, in the course of this experiment, suggests that learning modality preferences requires a lot of time. Further evidence that learning occurs is the increase of correction speed with experience: an experienced user achieved 6.8 cwpm, which is significantly faster than the average of 4.8 cwpm reported in Table VI.

Further analyses reveal noteworthy details about the user choice with respect to respeaking. Figure 11 shows the average usage frequencies in the first correction attempt. As can be seen, respeaking is preferred in the first correction attempt. With increasing experience and repeated evidence that respeaking—and choosing from alternative words—are ineffective correction modalities, these modalities are used less frequently, in favor of spelling or handwriting. A two-way ANOVA indicates significant changes in usage frequency *for the first correction attempt* over time ($F = 20.1$; $df = 3{,}168$; $p < 0.01$). Post hoc comparisons confirm that the increase is significant for spelling and handwriting (Tukey $HSD = 0.05$; $df = 4, 14$; $p < 0.05$, one-tailed).

The bias toward respeaking is consistent with data from the postexperimental questionnaire: participants indicated they would prefer respeaking if it had the same accuracy as other modalities.

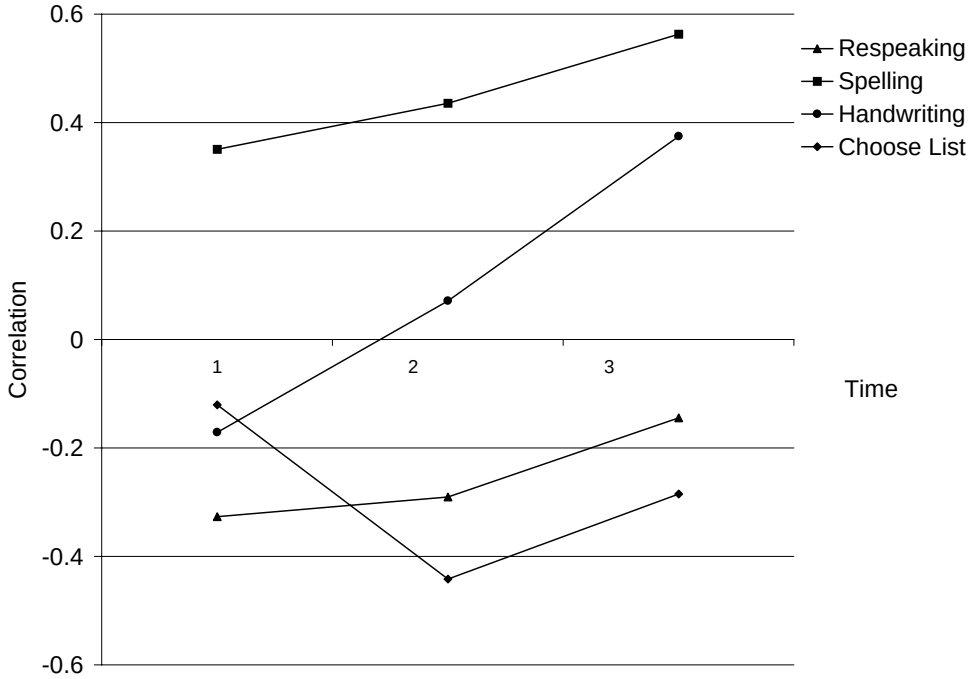

Fig. 10. Correlation between usage frequency and relative correction accuracy, in the course of the experiment.

6. PERFORMANCE MODEL FOR RECOGNITION-BASED MULTIMODAL INTERACTION

To extrapolate results from the user evaluation of multimodal repair, this section presents a performance model of recognition-based multimodal interaction. The model predicts interaction throughput. Interaction throughput was chosen as a performance measure because we expected rational users to prefer correction methods that minimize the effort, and effort is generally measured in time. Our analyses of modality preferences revealed that users prefer the most accurate correction method, but correction accuracy is obviously correlated with correction speed.

In related work, Mellor and Baber [1997] proposed a model of speech user interfaces that predicts task completion time, applying critical-path analysis. Although their model addresses imperfect recognition performance, and their model could be applied to multimodal interaction, it does not explicitly model the dependency of task completion time on modality, recognizer, or implementation-specific factors. The model presented here models these dependencies explicitly, based on an intuitive decomposition of the correction task. This work is still of preliminary nature and needs to be generalized to be useful beyond its application to multimodal correction in dictation systems.

6.1 Basic Performance Model

The performance model uses four basic parameters: recognition accuracy, input rate, recognition speed, and overhead time. To estimate correction and text creation speed based on these parameters, recognition-based multimodal interaction is decomposed in the following way: the user plans the interaction, chooses a modality, provides the necessary input, waits for the system to interpret the input, and finally decides whether further correction is necessary. How much time does such a multimodal interaction require? The steps of planning, choosing the modality, and the preparation of the actual input are modeled as overhead time $T_{Overhead}(m)$. Then, user provides correction input in modality m, which takes input time $T_{Input}(m)$ seconds.[2] The automatic interpretation of this input requires $R(m)$ times $T_{Input}(m)$ seconds, i.e., the recognition speed is captured by the real-time factor $R(m)$. $R(m) = 1$ means recognition finishes at the same time as user input, without any delay. This decomposition can be summarized by the simple linear additive relationship shown in Eq. (2).

$$T_{Attempt}(m) = T_{Overead}(m) + R(m) \cdot T_{Input}(m) \tag{2}$$

Basic decomposition interaction time
into overhead, input, and system response time.

Based on the estimate for the time to complete one interaction attempt derived in Eq. (2), the correction speed (measured in corrected words per minute) is the quotient of 60 seconds and the total time to correct a word. Assuming sequential interaction, the average total time is the product of the number of attempts until success $N(m)$ and the time per attempt. Thus, the correction speed can be estimated as shown in Eq. (3):

$$V_{Input}(m) = \frac{60}{N(m) \cdot T_{Attempt}(m)} \tag{3}$$

Factorization of correction speed into
time per interaction and number of interaction attempts.

Assuming a constant correction accuracy $CA(m)$ across repeated attempts (a simplifying assumption, as Figure 3 showed), the average number of correction attempts can be developed into a geometric series and estimated as $E[N(m)] = 1/(CA(m))$.

To apply the model, for example, to predict the correction speed as a function of correction accuracy, input rates are replaced by standard estimates, the overhead times and real-time factors are set to certain values, and correction accuracy is the independent variable.

[2]The inverse of the input time is the commonly known input rate, which is denoted as $V_{Input}(m)$. Examples for input rates include speaking and handwriting rate, and typing speed.

Table IX. Model Parameters for Interactive Error Correction, Estimated from Training Data (width of 95% confidence intervals in parentheses). The input speed for typing corrections is low because it includes the time required to position the hands over the keyboard, after selecting and error either by touch or by mouse.

Correction Modality	V_{input} [wpm]	Correction Accuracy [%]	Real-Time Factor	$T_{overhead}$ [s/correction]
Choosing from alternatives	58 (25)	21 (8)	1.0	4.6 (0.5)
Respeaking	47 (5)	36 (23)	2.6	5.4 (2.1)
Spelling	26 (6)	80 (17)	1.5	4.3 (0.7)
Handwriting	18 (4)	71 (8)	1.3	3.5 (1.1)
Pen-based gestures	36 (6)	86 (6)	1.0	5.0 (0.8)
Typing	17 (7)	84 (5)	1.0	2.6 (0.7)
Keyboard editing	N/a	82 (8)	N/a	4.3 (1.0)

How can the model parameters be estimated? Recognition accuracy and speed are standard performance parameters for any recognition system and are easily measured. Modality input rates have to be measured once; for standard input modalities (such as handwriting or typing), they can be found in the literature. Finally, overhead times depend on interface implementation and modality.

The data from the 15 participants was divided into a training set consisting of nine participants and a test set consisting of six participants. Table IX shows estimates, derived on the training set, for input rate (in words per minute), correction accuracy, real-time factor, and overhead time (in seconds per correction). These estimates will be used for predictions in following subsections. Although typing is not a recognition-based input modality, typing errors can be modeled similarly to recognition errors and quantified by the probability that a typed word is correct, denoted as CA(Typing) in Table IX.

6.2 Predicting Correction Speed for Complex Correction Methods

The previous sections introduced the model in a form suitable for sequential corrections in a single modality. But multimodal error correction, as evaluated in the user study, includes both recognition-based modalities (such as speech, spelling, handwriting, and pen gestures) and other modalities that are not interpreted with imperfect recognition (such as correction by choosing from alternative words). To predict correction speeds of correction methods that offer sets of correction modalities, such as multimodal error correction implemented in the prototype multimodal dictation system, the basic model must be extended in the following three ways:

(1) Correction by choosing from a list is modeled as one correction attempt that is successful with probability *CA(list)%*.

(2) Editing using pen gestures addresses other types of correction tasks than correction by respeaking, spelling, or handwriting. Interactive error correction typically requires both: words are deleted or the cursor is positioned typically *before* the user corrects by respeaking, spelling,

or handwriting. Therefore, time spent on editing gestures is modeled separately as $N(gest)T_{Attempt}(gest)$ and added to the overall correction time.

(3) Finally, multimodal error correction offers users a choice between different correction modalities, e.g., respeaking, spelling, and handwriting. User choice between different modalities m is modeled by empirical usage frequencies $freq(m)$.

Putting the pieces together—the average speed of correction by choosing from a list of alternatives, editing using pen gestures, and an additional set M of correction modalities to insert or replace words—the average correction speed is estimated as shown in Eq. (4).

$$V_{Correct}(M) = \frac{60}{CA(\text{“}list\text{”})T_{Attempt}(\text{“}list\text{”}) + (1 - CA(\text{“}list\text{”}))(N(\text{“}gest\text{”})T_{Attempt})(\text{“}gest\text{”}) + \sum_{m \in M} freq(m) \cdot N(m) \cdot T_{Attempt}} \quad (4)$$

Decomposition of correction speed for choosing from alternatives (“*list*”), pen gestures (“*gest*”), and a set M of correction modalities.

The user evaluation of interactive error correction, presented in Section 5, compared the following sets of correction modalities: conventional correction by choosing from alternatives, editing using keyboard and mouse, and typing ($M = \{typing\}$), and multimodal speech & pen correction by choosing from alternatives, editing using pen gestures, and repeating using speech, spelling, and handwriting ($M = \{speech, spelling, handwriting\}$).

6.3 Performance Model Validation

The performance model was validated by comparing model predictions with results from the user evaluation, as shown in Table X. The average absolute error of model predictions is used as measure of the goodness of fit for the performance model, as suggested in Kieras et al. [1997]. The average absolute error is 18% for multimodal correction ($N = 12$) and 12% for correction using keyboard and mouse input—within reasonable range for such empirical models. Predictions of text creation speed with a multimodal dictation system match empirical data equally well (cf. Suhm [1998]).

6.4 Modeling Interactive Error Correction in Dictation Systems

The model validation in the previous section showed that predictions from the performance model are reasonably accurate. This section applies the performance model to answer the following important questions about multimodal error correction in dictation systems:

(1) Under what conditions is multimodal correction faster than unimodal correction?

Table X. Validation of the Performance Model, Comparing Measured Correction Speeds (averaged across participants of test set) with Model Predictions

Correction Method	Participants in Test Set	Measured Correction Speed [cwpm]	Predicted Correction Speed [cwpm]	Signed Model Error
Speech & Pen Input	6	4.5	3.7	-18%
Keyboard & Mouse ("slow" typing)	2	5.9	6.2	5%
Keyboard & Mouse ("average" typing)	2	6.2	7.0	13%
Keyboard & Mouse ("fast" typing)	2	7.3	7.2	-1%

(2) What recognition accuracy is necessary to beat typing in correction speed?

(3) How does the text creation speed of a multimodal dictation system depend on dictation accuracy and error correction?

6.4.1 *Under which Conditions is Multimodal Correction faster than Unimodal Correction?* The correction speed depends on modality and recognition performance. To predict correction speed as a function of recognition performance and modality m, $T_{Attempt}(m)$ in Eq. (3) is replaced by Eq. (2), recognition in real time is assumed for all modalities ($R = 1$) in anticipation of faster computers. The remaining independent parameters are replaced by estimates for input rates as shown in Table IX. Finally, to normalize for implementation-specific differences across modalities, the overhead time is set to $T_{Overhead} = 3.0$ seconds for all modalities, which is more optimistic than the measured values.

Figure 12 shows, that at best with 100% recognition accuracy, correction by respeaking achieves 24 corrections per minute (cwpm), and correction by handwriting 15 cwpm. This compares favorably to correction by typing for users with good typing skills (15 cwpm).

Beyond predicting correction speeds, Figure 12 can be used to infer bounds on recognition accuracy such that unimodal correction by respeaking is as efficient as multimodal correction. While respeaking was slower than multimodal correction in this study, speech should ultimately be the fastest correction modality in a dictation system, provided that recognition can be made accurate enough. For example, multimodal corrections by spelling are 80% accurate with current recognizers (cf. Table IX). Figure 12 predicts that respeaking would be faster if respeaking was recognized with more than 60% correction accuracy, *across repeated correction attempts*. While some state-of-the-art recognition systems may achieve 60% accuracy on the first recognition attempt, maintaining a 60% average across multiple attempts is difficult. The speech recognizer used in this research achieved only 36% accuracy (cf. Table IX). Related research shows that accuracy of respeaking can be increased by adapting the speech recognizer on correction input [Soltau and Waibel 1998].

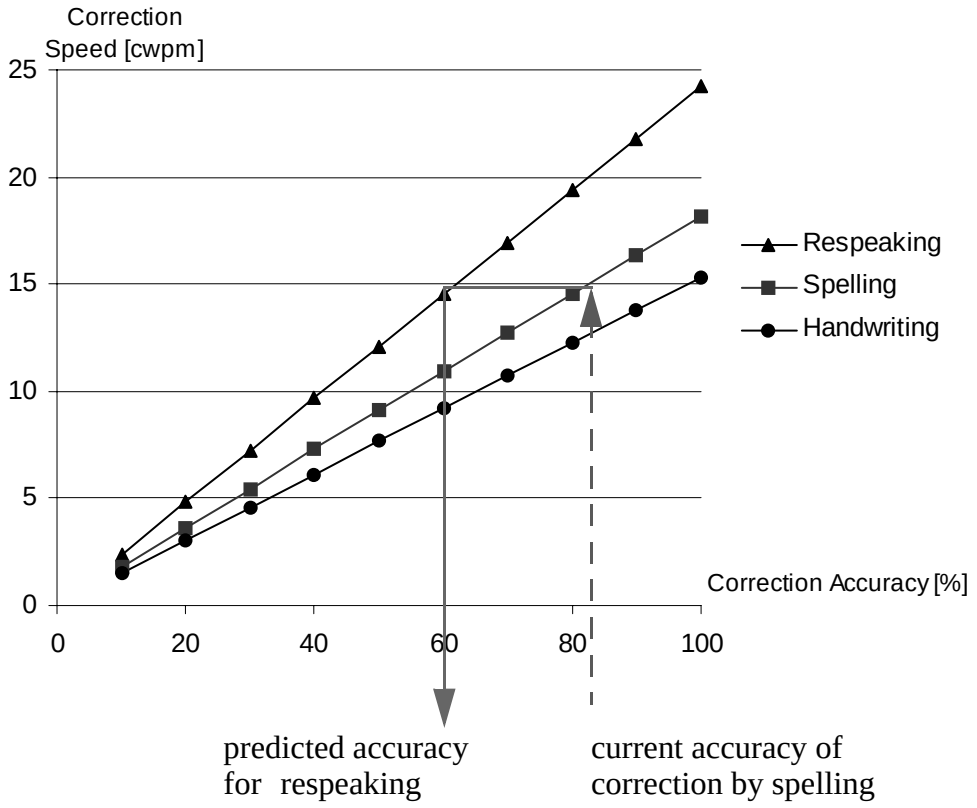

Fig. 12. Predicted correction speed for repeating in continuous speech, spelling, and handwriting.

6.4.2 *Comparing Multimodal with Typing Correction*. What correction accuracy is necessary to beat typing in correction speed? The answer establishes a lower bound on the recognition accuracy necessary such that dictation systems are really more productive than typing on text creation tasks. To answer this question, Figure 13 compares the speed of multimodal correction, as a function of correction accuracy, with the speed of correction by typing.

For example, a good typist can correct 15 errors per minute using mouse and keyboard. To beat this correction speed, correction by respeaking would have to be almost 65% accurate (see vertical dashed line in Figure 13), corrections by spelling 85% accurate, and correction by handwriting almost 100% accurate. Hence, multimodal correction would beat correction by typing even for users with good typing skills if correction accuracy could be further improved.

6.4.3 *Predicting the Productivty of Dictation Systems*. Moving beyond the issue of error correction, this section discusses implications for the overall text creation process. To assess the potential productivity gain of multimodal input methods, the model is applied to predict the text creation

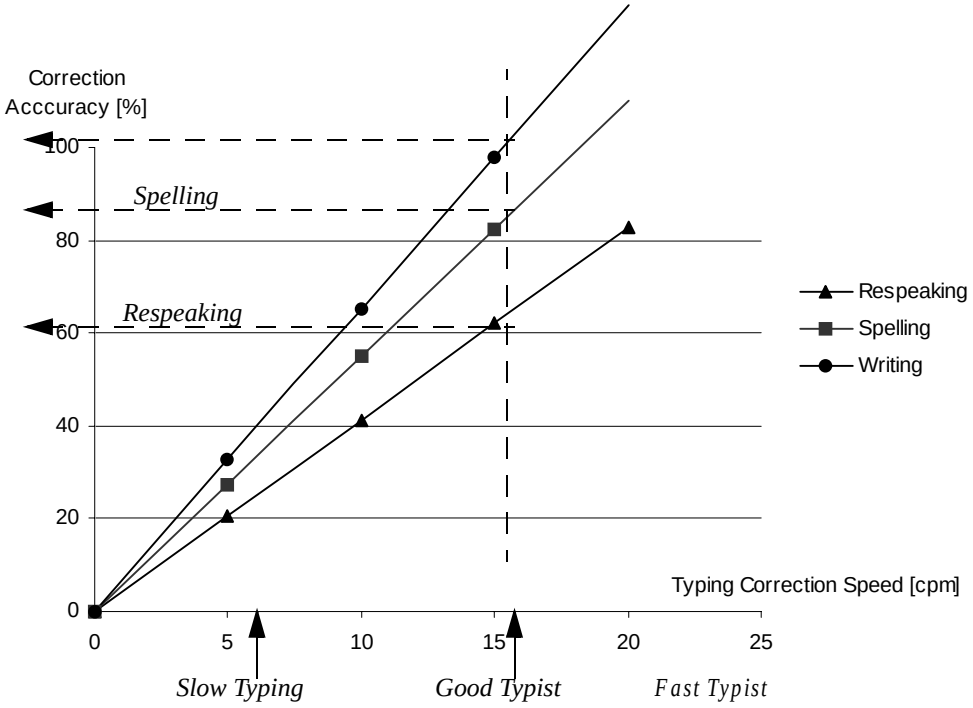

Fig. 13. Repair accuracy to beat typing in correction speed.

speed of an automatic dictation system. The formula derived in this subsection explicitly models the dependency of text creation speed on recognition accuracy and error correction method.

Text creation (transcription or composition) 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/dictating rate V_{input}(dictate) dictates $word_N = V_{Input}(dictate)$ words in one minute. Then, the speech recognizer needs $T_1 = R(m)$ minutes to interpret the dictation input. During automatic interpretation of the dictation input at accuracy *WA(dictate)*, on the average $error_N = word_N(1 - \text{WA(dictate)})$ recognition errors occur. The correction of these recognition errors using correction method m requires $T_2 = error_N T_{Correct}(m)$ seconds, where $T_{Correct}(m)$ is the inverse of the correction speed $V_{Correct}(m)$. The total time to input $word_N$ words including correction time is $T = T_1 + T_2$, leading to a simple formula for the text creation speed as a function of correction method and dictation accuracy.

Figure 14 compares the text creation speed of three text creation methods: a standard text editor (i.e., type the whole text), a conventional dictation system (i.e., first dictate, then correct using keyboard and choosing from alternatives), and a multimodal dictation system (i.e., first dictate text, then correct multimodally without any keyboard input). Note that the usage of the term "text creation speed" in this article is different from some

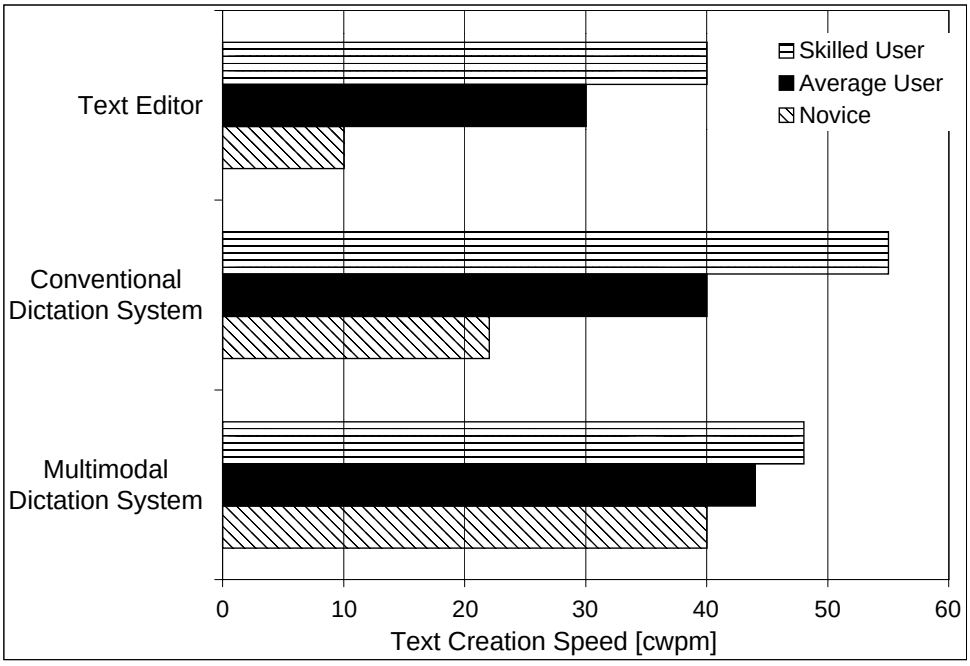

Fig. 14. Predicted speeds for different text creation methods, assuming 90% dictation accuracy.

commercial vendors of dictation systems who exclude the time necessary for correction, and thus can claim much higher speeds of 100 wpm (words per minute) and more. Figure 14 extrapolates the text creation speeds, as measured during our user study (with a dictation accuracy of 75%), to the performance of current commercial dictation recognizers, which achieve 90% accuracy in real time. Commercial recognizers achieve higher accuracies by adapting the speech recognizer to the user's voice. In our user study the dictation recognizer was not adapted to each participant so as to keep the length of experimental sessions within acceptable limits.

As can be seen in Figure 14, the text creation speed of a multimodal dictation system compares favorably to fast typing with 40 cwpm (correct words per minute). A multimodal dictation system allows all users to create texts fast without any keyboard input. Users with average typing skills are significantly more productive using a multimodal dictation system. The productivity gain is greatest for users with poor typing skills. However, for skilled typists, the fastest way to input text is to dictate using speech first, and then to correct all errors using keyboard and mouse input (shown as expert user of a conventional dictation system in Figure 14). For these users, the productivity gain for using a dictation system in only modest; dictation systems do not double or triple text creation productivity, as suggested by some commercial vendors of dictation systems.

7. DISCUSSION

The key result of the user study is that accuracy decreases in repeated correction attempts unless modality is switched. Previous studies have shown that accuracy significantly decreases in repeated spoken correction attempts [Levow 1998], and that users are more likely to switch modality following an error [Oviatt and VanGent 1996]. The present research extends these findings by demonstrating degraded recognition rates not only with respeaking but also with repeated spelled and handwritten corrections.

This study shows that multimodal error correction is faster than conventional unimodal correction by respeaking on a dictation task. We do not claim that respeaking is generally inefficient, nor that multimodal correction will always be faster. But our results suggest, that across modalities and across different state-of-the-art recognition systems, correction accuracy deteriorates when input is repeated in the same modality, but accuracy is higher when modality is switched for correction. The magnitude of this effect obviously depends on the recognition system and the task. Therefore, the decision about whether multimodal correction is faster than unimodal correction cannot be made in general. For the recognition systems used in this study there was a gain in using multimodal correction. In the future, if accuracy of unimodal correction by respeaking was significantly improved, it may outperform multimodal correction. The performance model presented in this article offers a general method to make such decisions for future speech and multimodal user interfaces.

Previous work suggested that automatic speech recognition technology could significantly increase productivity on dictation tasks [Gould 1978; Gould et al. 1983]. However, formal evaluations of dictation systems reported either only small productivity increases [Alto et al. 1989], lack of user acceptance despite significant productivity increases [Lai and Vergo 1987], or no gain at all [Karat et al. 1999], unless users learn efficient error correction strategies through extended exposure to the dictation system [Karat et al. 2000]. By applying the performance model, this article inferred lower bounds on recognition accuracy and error correction speed for realizing productivity gains with dictation systems. Generally speaking, the productivity gain of dictation systems may be smaller than widely assumed. First, most potential users of dictation systems have good typing skills, and this research showed, that for skilled typists, the productivity gain of dictation systems is rather modest. Second, regarding the creation of documents, studies suggest that not input speed, but the skill required to compose the text is the main limiting factor [Gould 1978].

In addition, this study investigated the important question of whether system-initiated error detection can improve dictation performance. Our results question the common belief (among many researchers in the speech recognition community) that confidence scores can facilitate error detection. While this result is limited to the present implementation of confidence scores, anecdotal comments suggest that other implementations by

developers of commercial dictation systems have failed to realize a gain as well.

Our user study also examines the issue of user choice between modalities and learning. On a dictation task, users initially preferred speech for error correction, but with repeated evidence that certain modalities are inefficient for correction (respeaking and choosing from alternatives), users eventually learn to switch to the most efficient modality. Other research [Karat et al. 2000] confirms that expert users learn to avoid inefficient usage of recognition-based correction methods. Our study remains inconclusive whether the initial bias toward using speech is an artifact of the dictation task, or whether users intuitively prefer to correct in the same modality as used for the initial input. But the results show that user intuition may counteract optimal recognition performance. Therefore, leading users to choose the "right" modality is a hard problem that designers of future speech and multimodal user interfaces must address. This is especially true for walk-up-and-use applications, where the designer cannot rely on the rational user's ability to learn which modalities are most efficient for a given task.

8. CONCLUSIONS

This article investigated the problem of correction errors in speech user interfaces and presented multimodal interactive error correction as a solution.

We presented novel multimodal methods that allow users to correct recognition errors efficiently without any keyboard input. Thus, multimodal error correction effectively solves the repair problem for speech recognition applications with a graphic user interface. Multimodal correction can be made more efficient by correlating the correction input with the repair context. The article also described how to integrate multimodal correction with a standard dictation system, i.e., how to engineer a multimodal dictation system. The user evaluation showed that multimodal correction is faster than unimodal correction by respeaking. Among multimodal correction methods, (conventional) multimodal correction by keyboard and mouse input, for skilled typists, is still faster than (novel) multimodal correction by speech and pen input. However, predictions from the performance model suggest that multimodal correction by speech and pen input could outperform correction by keyboard and mouse input for all users with modest improvements in recognition accuracy.

This research has important implications for speech recognition applications and multimodal applications in general. The user evaluation showed that not only high recognition accuracy, but also adequate error correction is crucial to realize productivity gains with dictation systems. Looking beyond dictation systems, this research showed that error correction is one of the areas that benefit from multimodal interaction. Multimodal input methods are particularly attractive for applications that do not allow fast keyboard input (e.g., small mobile devices), and for users with poor typing skills.

Addressing the challenging issue of choosing the set of modalities for future multimodal applications, the user evaluation showed that recognition accuracy has a significant influence on user choice between modalities: with practice, users learn to avoid ineffective modalities in favor of more effective modalities. Furthermore, while this research explored the trade-off between speed and accuracy of different modalities only for text input, it is clear that the most efficient input modality depends not only on input speed and accuracy, but also on the task. For example, for entry of numerical data, writing digits is about as fast as speech. We believe that the flexibility to change modality depending on the task holds great potential for future multimodal interfaces. While applications other than dictation may limit which alternative modalities are available, error correction benefits from just one alternative modality. If speech is the only modality available (e.g., in telephone applications), the speech user interface designer should consider switching between different speech modalities, such as continuous discrete, and spelled speech, or between speech and touch-tone input.

The performance model of (recognition-based) multimodal human-computer interaction presented in this article, while preliminary in nature, is a first step toward formalizing multimodal interaction. This article demonstrated how predictions from such a model help answer important design decisions in speech user interfaces, effectively complementing results from empirical evaluations. More generally, the investigation of multimodal correction showed the power of complementing component-level benchmark evaluations with both user studies and modeling techniques. As a contribution to evaluation methodology, we demonstrated that a user study and performance modeling complement each other in powerful ways, especially for evaluating multimodal interfaces.

ACKNOWLEDGMENTS

Many thanks to my former colleagues of the Interactive Systems Laboratories at Karlsruhe University and Carnegie Mellon University, who developed the continuous-speech, spelling, and cursive handwriting recognizers used in the prototype multimodal dictation system. Sincere thanks also to all participants of the user studies, to the reviewers for useful suggestions, and to Brinda Ganguly and Chloe Meadows for carefully proofreading the article.

REFERENCES

AINSWORTH, W. A. AND PRATT, S. R. 1992. Feedback strategies for error correction in speech recognition systems. *Int. J. Man-Mach. Stud. 36*, 6 (June), 833–842.

ALTO, P., BRANDETTI, M., FERRETTI, M., MALTESE, G., AND SCARCI, S. 1989. Experimenting natural-language dictation with a 20.000-word speech recognizer. *VLSI and Computer Peripherals: VLSI Comp. Periph. 2*, 78–81.

BABER, C. AND HONE, K. S. 1993. Modeling error recovery and repair in automatic speech recognition. *Int. J. Man-Mach. Stud. 39*, 3 (Sept.), 495–515.

BRINTON, B., FUJIKI, M., AND SONNENBERG, E. A. 1988. Responses to requests for clarification in linguistically normal and language-impaired children in conversation. *J. Sp. Hrg. Dis. 53*, 383–391.

CHASE, L. L. 1997. Error-response feedback mechanisms for speech recognizers. Ph.D. Dissertation. Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

COHEN, P. R., JOHNSTON, M., MCGEE, D., OVIATT, S. L., CLOW, J., AND SMITH, I. 1998. The efficiency of multimodal interaction: A case study. In *Proceedings of the International Conference on Spoken Language Systems* (Sydney, Australia).

GOULD, J. D. 1978. How experts dictate. *J. Exp. Psychol. 4*, 648–661.

GOULD, J. D., CONTI, J., AND HOVANYECZ, T. 1983. Composing letters with a simulated listening typewriter. *Commun. ACM 26*, 4 (Apr.), 295–308.

HILD, H. 1997. Buchstabiererkennung mit neuronalen Netzen in Auskunftssystemen (Spelling recognition for directory assistance systems with neural networks). Ph.D. Dissertation.

JELINEK, F. 1990. Self-organized language modeling for speech recognition. In *Readings in Speech Recognition*, A. Waibel and K.-F. Lee, Eds. Morgan Kaufmann Publishers Inc., San Francisco, CA, 450–506.

KARAT, C.-M., HALVERSON, C., HORN, D., AND KARAT, J. 1999. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *Proceedings of the International Conference on Computer-Human Interaction* (Pittsburgh, PA). 568–575.

KARAT, J., HORN, D. B., HALVERSON, C. A., AND KARAT, C. -M. 2000. Overcoming unusability: Developing efficient strategies in speech recognition systems. In *Proceedings of the International Conference on Computer-Human Interaction* (Amsterdam, The Netherlands).

KEMP, T. AND SCHAAF, T. 1997. Estimating confidence using word lattices. In *Proceedings of the European Conference on Speech Communication and Technology* (Rhodos, Greece). 827–830.

KIERAS, D. E., WOOD, S. D., AND MEYER, D. E. 1997. Predictive engineering models based on the EPIC architecture for a multimodal high-performance human-computer interaction task. *ACM Trans. Comput. Hum. Interact. 4*, 3, 230–275.

LAI, J. AND VERGO, J. 1997. MedSpeak: Report creation with continuous speech recognition. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '97, Atlanta, GA, Mar. 22–27), S. Pemberton, Ed. ACM Press, New York, NY, 431–438.

LEVOW, G.-A. 1998. Characterizing and recognizing spoken corrections in human-computer dialogue. In *Proceedings of the Annual Meeting of the Association on Computer Linguistic* (Montreal, Canada). 736–742.

MANKE, S. 1978. On-line erkennung kursiver handschrift bei großen vokabularien (On-line recognition of cursive handwriting with large vocabularies). Ph.D. Dissertation.

MANKE, S., FINKE, M., AND WAIBEL, A. 1995. NPen++: A writer independent, large vocabulary on-line cursive handwriting recognition system. In *Proceedings of the International Conference on Document Analysis and Recognition* (Montreal, Quebec).

MANKOFF, J., HUDSON, S., AND ABOWD, G. 2000. Providing integrated toolkit-level support for ambiguity in recognition-based interfaces. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI' 00, The Hauge, The Netherlands, Apr.). ACM Press, New York, NY, 368–375.

MARTIN, T. B. AND WELCH, J. R. 1980. Practical speech recognizers and some performance effectiveness parameters. In *Trends in Speech Recognition*, W. A. Lea, Ed. Prentice Hall Press, Upper Saddle River, NJ.

MCNAIR, A. E. AND WAIBEL, A. 1994. Improving recognizer acceptance through robust, natural speech repair. In *Proceedings of the International Conference on Spoken Language Processing* (Yokohama, Japan). 1299–1302.

MELLOR, B. AND BABER, C. 1997. Modelling of speech-based user interfaces. In *Proceedings of the European Conference on Speech Communication and Technology* (Rhodos, Greece). 2263–2266.

MURRAY, A. C., FRANKISH, C. R., AND JONES, D. M. 1993. Data-entry by voice: Facilitating correction of misrecognitions. In *Interactive Speech Technology: Human Factors Issues in the Application of Speech Input/Output to Computers*, C. Baber and J. M. Noyes, Eds. Taylor and Francis, Inc., Bristol, PA, 137–144.

OVIATT, S. 1999. Mutual disambiguation of recognition errors in a multimodal architecture. In *Proceedings of the International Conference on Computer-Human Interaction* (Pittsburgh, PA). 576–583.

OVIATT, S. AND VANGENT, R. 1994. Error resolution during multimodal human-computer interaction. In *Proceedings of the International Conference on Spoken Language Processing*. Accoustical Society of Japan, Japan, 204–207.

OVIATT, S., DEANGELI, A., AND KUHN, K. 1997. Integration and synchronization of input modes during multimodal human-computer interaction. In *Proceedings of the ACM Conference on Human Factors in Computing Systems* (CHI '97, Atlanta, GA, Mar. 22–27), S. Pemberton, Ed. ACM Press, New York, NY, 415–422.

OVIATT, S., LEVOW, G. A., MACEACHERN, M., AND KUHN, K. 1994. Modeling hyperarticulate speech during human-computer error resolution. In *Proceedings of the International Conference on Spoken Language Processing*. Accoustical Society of Japan, Japan, 797–800.

RHYNE, J. R. AND WOLF, C. G. 1993. Recognition-based user interfaces. In *Advances in Human-Computer Interaction*, H. R. Hartson and D. Hix, Eds. Ablex Publishing Corp., Norwood, NJ, 191–212. http://www.eecs.tufts.edu/~jacob/papers/hartson.txt; http://www.eecs.tufts.edu/~jacob/papers/hartson.ps.

ROBBE, S., CARBONELL, N., AND VALOT, C. 1994. Towards usable multimodal command languages: Definition and ergonomic assessment of constraints on users' spontaneous speech and gestures. In *Proceedings of the International Conference on Spoken Language Processing* (Yokohama, Japan). 1655–1658.

ROGINA, I. AND WAIBEL, A. 1995. The JANUS speech recognizer. In *Proceedings of the ARPA Workshop on Spoken Language Technology* (Austin, TX). 166–169.

RUBINE, D. 1991. Specifying gestures by example. *SIGGRAPH Comput. Graph. 25*, 4 (July), 329–337.

SHNEIDERMAN, B. 1997. *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Addison-Wesley, Reading, MA.

SOLTAU, H. AND WAIBEL, A. 1998. On the influence of hyperarticulated speech on recognition performance. In *Proceedings of the International Conference on Spoken Language Systems* (Sydney, Australia).

SUHM, B. 1997. Exploiting repair context in interactive error recovery. In *Proceedings of the European Conference on Speech Communication and Technology* (Rhodos, Greece). 1659–1661.

SUHM, B. 1998. Multimodal interactive error recovery for non-conversational speech user interfaces. Ph.D. Dissertation.

SUHM, B., MYERS, B., AND WAIBEL, A. 1996. Interactive recovery from speech recognition errors in speech user interfaces. In *Proceedings of the International Conference on Spoken Language Processing* (Philadelphia, PA). 861–864.

SUHM, B., MYERS, B., AND WAIBEL, A. 1999. Model-based and Empirical evaluation of multimodal interactive error correction. In *Proceedings of the International Conference on Computer-Human Interaction* (Pittsburgh, PA). 584–591.

VERGO, J. 1999. Multimodal text creation and manipulation. Presented at the CHI Workshop on Designing the User Interface for Pen and Speech Multimedia Applications. Proceedings of the ACM Conference on Human Factors in Computing Systems. ACM, New York, NY.

VO, M. T. 1998. A framework and toolkit for the construction of multimodal learning interfaces. Ph.D. Dissertation. Computer Science Department, Carnegie Mellon University, Pittsburgh, PA.

WOLF, C. G. AND MORREL-SAMUELS, P. 1987. The use of hand-drawn gestures for text editing. *Int. J. Man-Mach. Stud. 27*, 1 (July 1), 91–102.