
INTERNATIONAL JOURNAL OF SPEECH TECHNOLOGY 6, 183–194, 2003

Speech Error Correction: The Story of the Alternates List

KEVIN LARSON AND DAVID MOWATT
Microsoft, 1 Microsoft Way, Redmond, WA 98052, USA
kevlar@microsoft.com
dmowatt@microsoft.com

Abstract. Error correction with speech recognition products is extraordinarily difficult for users. Users spend much more time correcting errors than they spend dictating new text. In order to find ways to improve users' error correction experience, we examined the use of four different error correction mechanisms. The two error correction methods that users were most successful with were redictation and selection of a list of alternatives ("the alternates list"). Users rated the latter as the more satisfying method. User satisfaction with the alternates list was surprising as it was not a terribly accurate error correction method. On the Tablet PC we made several interface enhancements to facilitate the use of the alternates which included the use of (1) strong modes, (2) a push-to-talk model for microphone control, (3) a lighter weight alternates list which was easier to open and dismiss. Users performed transcription tasks with this new interface and we examined which error correction methods people preferred. Users of the new interface no longer compounded error upon error and were far more likely to use the alternates list than was the case for users of pre-existing interfaces. Users were very likely to switch modes from the alternates list to redictation when the alternates list did not contain the target word.

Keywords: error correction, alternates list

Introduction

Automatic speech recognition (ASR) is an exciting technology that promises a more natural way of interacting with computers in the future. An obstacle with ASR is that recognition accuracy is not 100%, and no one projects that a 100% solution is forthcoming. For ASR to be useable, there must be a quick and easy way to correct recognition errors. Currently, users of ASR systems spend much more time correcting errors than they spend dictating new text. The goal of this paper is to search for trends that we can leverage to make error correction an easier process. We will start by examining the literature on error correction strategies.

Karat et al. (1999) described two problems with ASR that might hinder its adoption by users. The first problem is that speech recognition errors are different from keyboard errors as the system, rather than the user, is frequently the cause of the errors. When a user presses a key on the keyboard, the corresponding letter will appear on the screen. If a different letter appears than expected, it is most likely caused by the user pressing the wrong key accidentally. In contrast, when a user speaks a word to a speech engine, a percentage of the words is recognized incorrectly. This causes users great frustration. A second problem that hinders adoption is that some users claim that speech is not a natural method for composing text because they have become accustomed to using the keyboard to do this. Change is unwelcome to most users except when the benefits are immediate and clear.

Karat et al. (1999) examined user performance with a keyboard and with the three major commercial ASR systems. A subset of their participants completed extensive work with an ASR system over a period of weeks. They found that for transcription tasks participants could input at a rate of 32.5 corrected words per minute (CWPM) with a keyboard, 13.6 CWPM for initial ASR use, and 25.1 CWPM for extended use. Karat et al. (1999) described four kinds of errors that occurred

during this study:

- Simple misrecognitions or one-word errors,
- Multi-word misrecognitions,
- Commands inserted as text
- Dictation inserted as command misrecognitions.

The participants used the following correction strategies to correct the misrecognition:

- 38% select text then reenter
- 23% delete than reenter
- 8% use an advanced correction dialog

32% of corrections are necessitated by the need to correct new problems created during an earlier correction attempt.

Oviatt and VanGent (1996) demonstrated that users usually first attempt to correct an error with the same modality (e.g. speech) that caused the error, but will not continue with the same modality forever. After the first attempt, users will switch to a less error prone modality such as typing letters. Oviatt (1999) further generalized this finding. It is now generally accepted that users will attempt to switch from one correction method to another looking for one that will be successful.

Halverson et al. (1999) investigated the errors that occurred in the Karat et al. (1999) study in greater detail. They used the term *correction episode* to describe all the steps needed to correct an error. Minimally, every correction episode needs at least two steps or *correction primitives*: (1) selecting the misrecognized word then (2) changing it to the correct word. There were five techniques for correcting errors:

- Undo command followed by redictation (2 steps);
- Select command followed by redictation (2 steps);
- Select command, open the correction dialog box, and select an alternate (3 steps);
- Select command, open the correction dialog box, spell, and close correction dialog (4 steps);
- Select command, open the correction dialog box, type, and close correction dialog (4 steps).

The most common correction technique that the participants chose was redictation. Redictation was used 40% of the time despite working successfully only 47% of the time. Selecting an alternate was used only 8% of the time even though it was usually successful.

In Halverson et al. (1999), participants initially used 7.3 steps on average to correct an error and experienced participants used 3.5 steps. They identified two kinds of errors that prevented participants from being efficient: (1) redictating multiple times and (2) creating new errors when the speech engine incorrectly distinguishes between dictation and a command. *Spiral depth* is the term for the number of times a participant will redictate. 50% of the time participants continued to a spiral depth of 3 and 25% of the time to a spiral depth of 4. Efficient users learn to stop redictating at a spiral depth of 2 and to switch to another method for correcting the error. A cascade of errors occurs when a new error is introduced before the original error is successfully corrected. Cascades are caused by commands being recognized as dictation or vice-versa. 22% of the correction primitives were in response to errors created by a misrecognized command. The correction dialogs in these systems were unhelpful because they assumed that participants would invoke them first and not last. When participants did invoke them last, the speech engine had already discarded the data, leaving the alternates list empty.

Suhm et al. (1999) demonstrated that participants who were proficient with a keyboard were most successful correcting errors using the keyboard. More generally, given that users currently have a higher CWPM (Corrected Words Per Minute) rate with keyboards than with speech even after extended use, there is little incentive for users to switch to existing ASR systems. However, a variety of popular, new devices do not have a standard, full-size keyboard, and on these devices ASR may be a compelling input mechanism. In the following two studies, we will examine methods of speech error correction that are available on devices like a Pocket PC, Palm, or Tablet PC.

While there is good research analyzing the current usage of error correction methods in research and commercial ASR systems, further work would provide direction for guiding the development of better error correction methods.

Comparison of Four Error Correction Methods

The first question we will address is the success of different error correction methods when we restrict a participant to use one particular method. By testing one method at a time, we will avoid the problem Halverson et al. (1999) experienced where the engine discarded relevant speech data after the first error correction attempt. We will not include a regular keyboard since it has already been shown to be the most successful error

correction mechanism for most users. We will choose to examine four error correction methods that are most prevalent in ASR systems today: voice commanding, redictation, selecting from an alternates list, and using a soft-keyboard (a keyboard displayed on the bottom of the screen where users 'type' by clicking virtual keys). Spelling was not examined because users only spontaneously attempted to speak letters in 1 of 1,680 error correction attempts in Oviatt and VanGent (1996).

Some in the industry have claimed that people can expect to enter text at up to 140 WPM with ASR, more than three times faster than the average computer user can type. While 140 WPM might be the rate some people speak, users input text at a significantly slower rate when time to correct errors is factored in. It is desirable to reduce the amount of time it takes users to correct errors. The goal of this research is to discover which error correction methods are quick as well as satisfactory to the user.

In this study, we will examine four error correction methods:

- Voice Commanding—the participant selects the misrecognized word with a speech command and then redictates the word to correct it. If redictation fails twice then the participant can switch to the soft-keyboard to correct the error.
- Redictation—the participant selects the misrecognized word with a mouse and then redictates the word to correct it. If redictation fails twice then the participant can switch to the soft-keyboard to correct the error.
- Alternates List—the participant uses the mouse to select the misrecognized word and to open the alternates list dialog. The participant then will select the correct word with the mouse if it is available or will close the dialog and will switch to the soft-keyboard to correct the error.
- Soft-keyboard—the participant selects the misrecognized word with the mouse and immediately uses the soft-keyboard to correct the word.

Methods

Participants. Twelve participants performed tasks in each of the four error correction conditions. Each participant used only one of three speech recognition systems. This was done both for time considerations and so that a participant's familiarity with one system would not conflict with use of a later system. Four participants used each system. No participant had prior experience with any of the speech recognition systems. All participants were native U.S. English speakers. Throughout, we will refer to those who participated in the usability study as "participants", while generic users of ASR systems will be referred to as "users".

Materials. Three commercial voice recognition programs were used in this study: IBM Via Voice Pro 8.0, L&H Voice Xpress Pro 5.0, and Dragon NaturallySpeaking Preferred 5.0. Differences in these systems were not examined as an independent variable in this study.

The participants used an Andrea desktop microphone instead of the recommended close-talk microphone. This had the deliberate effect of increasing the total number of errors, though the increase was constant across conditions because the same microphone was used in all four error conditions.

Following Karat's lead (1999), the dictation tasks consisted of 87–90 word paragraphs taken from an old western novel. There were no out-of-vocabulary (OOV) words in these paragraphs, but the writing was unusual enough that the engine's language model was certainly not optimized for this style of writing.

Procedure. At the start of the study, participants were briefed about the basics of speech recognition and were told that the goal was to examine different kinds of error correction mechanisms. Participants completed microphone training, one session of enrollment (speech engine training), and all relevant tutorials using the built-in tutorials for the speech recognition system. Participants were guided whenever they asked for help or ran into any trouble to make sure that their initial speech recognition experience was a good one. Participants were given a practice task in each of the four error correction conditions. Participants practiced dictating and correcting a sentence until they were comfortable with the error correction condition, and were provided with any support they needed to help them understand each condition.

After training, tutorials, and practice items, the participants completed a single 87–90 word dictation task four times—once using each condition. The four conditions were voice commanding, redictation, alternates list, and soft-keyboard. In all four of these conditions, the participants used speech to enter new text (dictation), but used different methods to correct any errors that occurred. A Latin square design was used to order the conditions differently for each participant.

For each condition, we measured the time to correct the paragraph. At the end of the study, we asked participants about their satisfaction level using each correction method.

Results

Transcription Time. On average, the participants dictated the paragraphs in this study in 53 seconds for an uncorrected rate of 102 words per minute. On average across all conditions, participants took 544 seconds to correct those same paragraphs for a corrected input rate of 10 words per minute. The uncorrected rate of 102 is reasonably close to some industry claims of 140 words per minute, but there is an immense difference between rates of corrected and uncorrected words per minute.

On average, participants were fastest correcting with the soft-keyboard at 314 seconds (SD = 65), next fastest with the alternates list at 403 seconds (SD = 144), third fastest with redictating at 411 seconds (SD = 128), and slowest with voice commanding at 1,029 seconds (SD = 374) (Fig. 1).

There is a highly reliable one-way ANOVA $F(3, 33) = 37.57$, $p < .001$. There are reliable post-hoc differences between voice commanding and each of the other three conditions $p < .01$.

Satisfaction Scores. Participants were given 7-point Likert scale statements, which they would rate a 7 if they completely agree with the statement and a 1 if they completely disagree with the statement. For the satisfaction statement "I liked it", on average participants rated the alternates list highest at 5.3 (SD = 0.5), redictation second highest at 5.1 (SD = 0.6), the soft-keyboard third highest at 4.4 (SD = 0.8), and voice commanding lowest at 3.5 (SD = 1.1). For the satisfaction statement "I had control", on average participants rated the soft-keyboard highest at 6.2 (SD = 0.4), redictation second highest at 5.1 (SD = 0.7), the alternates list third highest at 5.0 (SD = 0.6), and voice commanding lowest at 3.0 (SD = 0.9) (Fig. 2).

For the statement "I liked it", there is a reliable one-way ANOVA $F(3, 33) = 3.59$, $p < .05$. The only reliable post-hoc difference is between voice commanding and alternates list, $p < .05$. For the statement "I had control", there is also a reliable one-way ANOVA $F(3, 33) = 14.83$, $p < .001$. There are reliable post-hoc differences between voice commanding and each of the other three conditions at the $p < .01$ level.

Discussion

Voice commanding faired poorly in this study, taking 10 minutes longer than any other correction strategy, and receiving poorer satisfaction scores than any other condition. The only difference between voice commanding and redictation is the presence of selecting words with voice commands instead of using a mouse. Four problems caused participants to take more time in this condition.

- First, it was difficult to learn the available commands. Though it is not reflected in the task times, participants spent much of the time during tutorials and

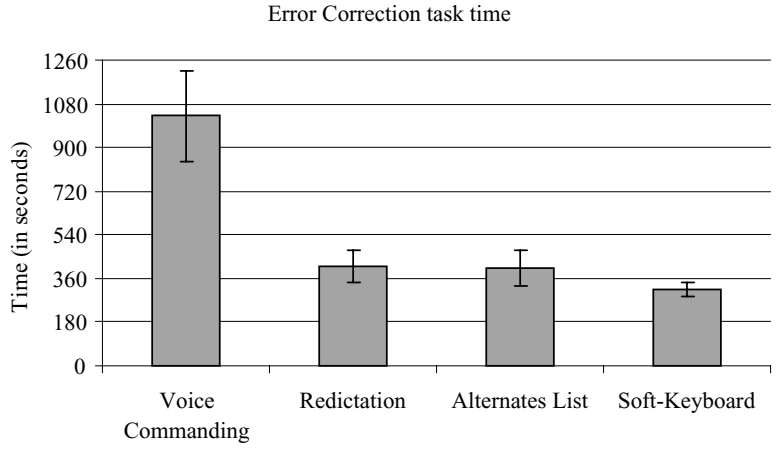

Figure 1. Time to correct a 90-word paragraph in each condition.

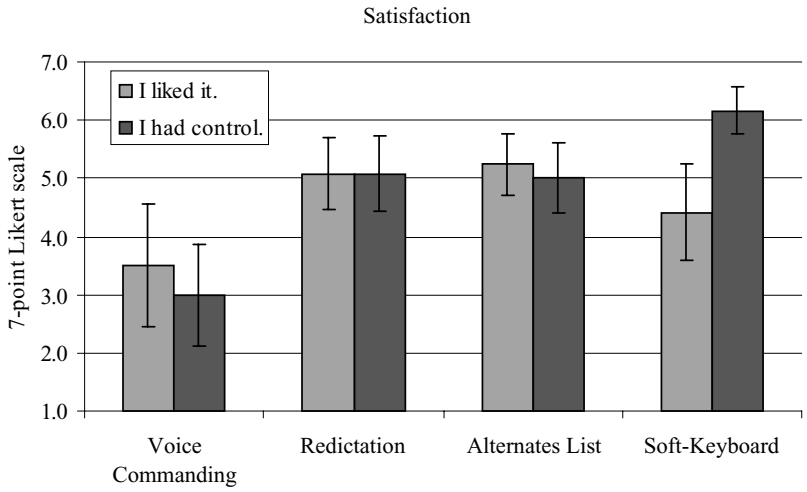

Figure 2. Participant satisfaction ratings to the two statements "I liked it" and "I had control."

the practice items learning the available commands. Participants also would examine the list of available commands during the tasks.

- Second, it was difficult to select words, particularly common words like "be" or "to". If there are multiple instances of a word in a document the speech recognition program may select a different instance than the one intended. It could take several repetitions of the command to achieve the desired result.
- A third problem is that commands are frequently interpreted as dictation. For example, if a participant pauses between the command "select" and the target word, both the word "select" and the target word will be transcribed. If a command is misrecognized, it will frequently be inserted as dictation.
- The fourth problem is that errors tend to compound and cause frustration. This frequently is in combination with the commands being interpreted as dictation. If a participant is trying to correct a mistake and a new mistake is generated, it becomes especially hard to fix the resultant compound error.

The soft-keyboard received mixed results in this study. It was slightly faster than any other error correction method, and participants rated it highest for the statement "I had control." Participants were 98% accurate with the soft-keyboard, and those errors were not caused by the keyboard misinterpreting the pressed key but by the participant missing the intended key with the mouse. The soft-keyboard was rated lower than both redictation and the alternates list for the statement "I liked it." Participants complained that typing on a soft-keyboard was much slower than typing on a regular keyboard. This is the case because the motor skills available for two-handed, eyes-free typing are very different from one-handed eyes-on tapping (MacKenzie et al., 1999).

Redictation and the alternates list correction methods had similar task times and satisfaction ratings. Both had task times that were slightly slower than the soft-keyboard, and both had above-average responses to the satisfaction statements "I liked it" and "I had control." Both methods performed well because they did not suffer from the selection errors or spiraling errors that voice commanding suffered from, but neither method was quite as accurate as the soft-keyboard. Participants were successful in correcting a word with redictation 68% of the time and successful with the alternates list 38% of the time. The rest of the time participants switched to the soft-keyboard.

Interface Improvements to Encourage Use of the Alternates List

Of the four error correction methods studied here, the alternates list showed the most promise. It received the highest ratings on the "I liked it" satisfaction statement, just slightly higher than redictation, and participants only completed tasks faster with the soft-keyboard. This was quite surprising, given that it helped correct a misrecognition less than half the time. As the alternates list becomes more accurate with improvements in ASR technology, we believe it has the potential to increase

dramatically in popularity as a correction method. In Karat et al. (1999) the alternates list was only used 8% of the time because of problems with items in the alternates list disappearing and because it took three steps while dictation only takes two steps.

We took this research and incorporated it into the design of the design of the Microsoft Windows XP Tablet PC edition ASR system. We made several design changes to facilitate the use of the alternates list correction mechanism. Four significant design choices were:

- The use of *strong modes*, allowing the user to tell the system if an utterance is a dictation or a command
- Requiring the user to *push-to-talk*, to hold down a button when the user intends to be recognized
- Evangelizing the "correct <word>" command as a low cost mechanism for examining the alternates list
- Improving the engine's alternate list accuracy so the correct word is more likely to be in the list.

Strong Modes

In the current iteration of speech recognition on Microsoft Office XP, there are two mode controls. One is the microphone on/off control that is standard on desktop speech programs like Dragon NaturallySpeaking, IBM ViaVoice, and L&H Voice Express. The second control is for dictation/command & control. This control exists to improve the accuracy of the speech recognition. When the command mode is selected, the SR engine will not confuse a command to access menu functionality with text insertion. To reduce the frequency of mode switching, users can both insert text and use text-editing commands (i.e. a limited subsection of all available commands) when dictation mode is selected. However, on Tablet PCs, all the text- editing commands have been removed from dictation mode by default (advanced users can change this back). Users can use the dictation button only for text entry and the command button only for text editing commands and menu commands (Fig. 3). We named the separation between the functionality of the two buttons "strong modes".

The incorporation of strong modes led to an improved error correction experience for novice users because it eliminates the problem of cascades of errors. When the user issues a command while in command mode, the command will never be inserted as text. Similarly, while in dictation mode it is impossible for a text insertion to be interpreted as a command. In combination, these two problems cause cascades of errors.

Figure 3. Virtual buttons for turning on dictation and voice command modes (dictation mode currently on).

Push-to-Talk

A second improvement implemented as the result of usability feedback was the addition of push-to-talk buttons. Instead of pressing once to turn dictation (and thus the microphone) on and leaving it on until it is pressed a second time to turn it off, users instead hold the dictation button down while dictating and the command button down while speaking a command. On the Tablet PC, this is most effectively done with configurable hardware buttons on the side of the Tablet (Fig. 4), but can also be done by holding the pen down on the software dictation and command buttons, or even assigning keyboard or mouse buttons on a desktop system to do the same.

Push-to-talk has been a very successful interface mechanism on a research prototype called MiPad

Figure 4. Prototype Tablet PC hardware with buttons on the right side that can be configured for push-to-talk.

(Huang, 2000). The advantage of push-to-talk on the Tablet PC is that it reduces error caused by extraneous noise between the time the user stops speaking and remembers to turn off the microphone. Sometimes that noise can be caused by someone walking in and speaking before the user had time to turn off the microphone, and, more often, caused by the user vocalizing thoughts without intending to speak to the system. Push-to-talk is more effective at shorter utterances and less effective when leaving an open microphone for a long period of time. Most ASR utterances are short such as commanding to the system or composition where phrases or sentences are spoken and then corrected. With push-to-talk, stopping the ASR engine is a simple matter of the user lifting a finger or pen off of the button.

"Correct <word>" Command

A third improvement was to make it easier to access and dismiss our alternates list. There are two ways the alternates list can be accessed. The first is with the voice command "correct <word>" where <word> is the name of the misrecognized text. This command has the effect of both selecting the word and opening a list of alternates directly beneath that word.

A second way of accessing the alternates list (Fig. 5) is with the pen (which acts as a mouse does with a normal PC). After selecting the misrecognized word with the pen, a new control (in the shape of a green corner) appears to the left of the word. Selecting this control will also open the alternates list beneath the word. This control is also used for accessing a list of alternates when users are correcting words misrecognized with the handwriting input modality on the Tablet PC.

Once the alternates list is open, users can scan the list for the correct word. If the correct word is in the list, they can select it with a "select ⟨number⟩" voice command or by selecting it with the pen. To close the alternate list, users can issue a cancel voice command or go to the dictation button and say the correct word. This has the effect of closing the alternates list and replacing the word with a potentially correct new word.

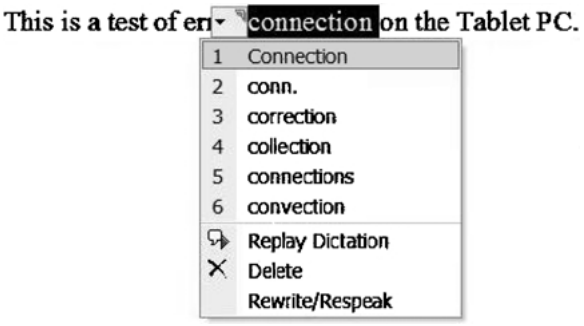

Figure 5. The Alternates list on the Tablet PC.

This is an improvement over systems that require users to open and close separate alternates list dialogs. In systems that have explicit open and close dialog actions, many additional steps are needed to complete corrections. Halverson et al. (1999) described successful correction with the alternates list as three steps because of the additional commands for both selecting a word and opening the correction dialog. This requires only two steps using the "correct <word>" command. Switching modality from the alternates list to redictation in Halverson is five steps because the participants needed to close the correction dialog before redictating. With our system, users only need three steps because they can immediately switch from the alternates list to redictation.

Better Alternates

It was an explicit goal for the Microsoft ASR group to populate the alternates list with more suggestions, while not filling it with so many words (i.e. noise) that the correct word could not be found quickly by the user.

In dictation mode, version 5 of the Microsoft SR engine pruned and discarded the recognition hypotheses (generated as part of the recognition pass) to such an extent that there was too little information to generate many alternate recognitions. After Office XP and for version 6 of the SR Engine, we explicitly saved more hypotheses during the recognition pass so as to be able to generate more alternates.

Additionally, the v6.0 SR engine now explicitly populates the search space with exact homophones ("there" vs. "their") and quasi homophones ("spied" vs. "spade"). This was done using a phone decoder to map any sound (e.g. the /a/ in "spade") to similar sounds (e.g. /ie/ in "spied" or /ee/ in "speed"). We discovered this worked best when the least probable words (i.e. the lowest language model scores) were explicitly not considered (e.g. not considering "inks" as an alternate of "things"). This resulted in production of more alternates without adding a large number of incorrect alternates in the list.

Finally, we added Inverse Text Normalization (ITN) alternates to the list of alternates as a post-processing

stage, so "2" had an alternate of "two", "Dr." had an alternate of "doctor" and vice versa. These, together with generally improved recognition accuracy, made our alternates significantly more accurate and thus more efficient as a form of error correction.

Error Correction Usage on the Tablet PC

The first interface that will include strong modes, push-to-talk, emphasis on the "correct <word>" command, and the improved alternates is the Microsoft Windows XP Tablet PC edition. Given these changes, we want to know how this will influence the error correction methods that users choose. We expect to see the alternates list used for a greater percentage of corrections than in previous studies that examined error correction usage.

The purpose of this second study is to examine the pattern of usage in error correction methods on the Tablet PC, similar to what Halverson et al. (1999) did for earlier ASR systems. Users will not be instructed to use any particular method. We will collect data on the error correction methods that users choose and the combinations of methods that users choose when the first attempt fails.

Methods

Participants. Six participants performed identical transcription tasks. The participants were equally distributed by gender, and they ranged in age from 20–50. No participant had prior experience with any of the speech recognition systems. All participants were native U.S. English speakers.

Materials. All participants used a desktop computer with a pre-release build of Tablet PC with the Tablet PC Microsoft v6.0 ASR engine. Participants had access to an onscreen keyboard and two virtual push-to-talk buttons for controlling the microphone. Users pressed the dictation button to insert text and punctuation, and used the command button to issue all other commands. The participants used a closetalk microphone.

The dictation tasks consisted of property descriptions taken from the newspaper. The descriptions were normalized to remove all abbreviations, but irregular grammar and sentence formation were not changed. There were 20 test tasks and 4 practice tasks, each averaging 39.1 words (SD = 7.6). There were no out-of-vocabulary words in these paragraphs, but the writing was unusual enough that the engine's language model was not optimized for this style of writing. OOV words were not included in this study because they would bias the results by forcing users to correct with the soft-keyboard, the only method that can correct that error.

Procedure. At the start of the study, we briefed participants about the basics of speech recognition and told that the goal was to examine different kinds of error correction mechanisms. Participants completed microphone training, one session of enrollment, and were given an in-person demonstration of the software and its error correction mechanisms. The information that was contained in the in-person demonstration was later used to build the product tutorial. Participants were also guided whenever they asked for help or ran into any trouble to make sure that their initial speech recognition experience was a good one. Participants completed four practice tasks before moving on to the test tasks, and were provided with any support they needed to help them understand the range of available error correction mechanisms.

Participants were told that we were interested in seeing what error correction mechanisms they thought would help them be most efficient. Much care was taken not to bias their usage towards any particular error correction mechanism.

For each of the 20 test tasks a variety of measurements were taken. For each task, we measured the time to dictate and correct the entire transcription. We counted the total number of insertion, omission, and substitution errors. For each substitution error, we recorded the method of initial error correction used and what method was used on each subsequent attempt. While participants were expected to correct capitalization errors in these tasks when they happened, capitalization errors were not counted as misrecognitions nor were the correction methods recorded.

Results

Transcription Time. On average the participants entered and corrected the transcriptions at a rate of 17.3 CWPM (SD = 3.6). This rate is slightly faster than the fastest individual correction rate in the previous study.

Error Analysis. There were a total of 589 recognition errors in this study.

- The engine omitted 33 words; participants redictated each of these errors.
- The engine added 56 words; participants deleted each of these errors.
- 9 words went uncorrected by the participants.
- The rest of the 490 errors were substitution errors; we will analyze these errors in greater depth.

For simplicity, we will ignore the method that was used to select the error and concentrate on the methods that were used to correct the error. Documenting the number of selections using the mouse versus those by voice commands before redictating a word would add a needless level of complexity. The three correction methods used were redictation, alternates list, and soft-keyboard.

291 of the 490 substitution errors were successfully corrected on the first try (59%); 172 were corrected with redictation, 90 with the alternates list, and 29 with the keyboard (Fig. 6). Each of these was a two-step procedure. For redictation, the participant would either select the word with the mouse, a select command, or an undo command before redictation. For the alternates list, the participant would use the "correct <word>" command or open the alternates list with the mouse using the control next to the word before selecting the correct word. The correct alternate could be selected either with the mouse or with a voice command. For the soft- keyboard, the participant could select the word with a voice command or with the mouse before typing the correct word. When the soft-keyboard was used as an initial correction method, it was usually in cases where only small changes were needed, such as adding a plural "-s" to the end of a word.

Figure 6. Success and failures with different methods on first correction attempt.

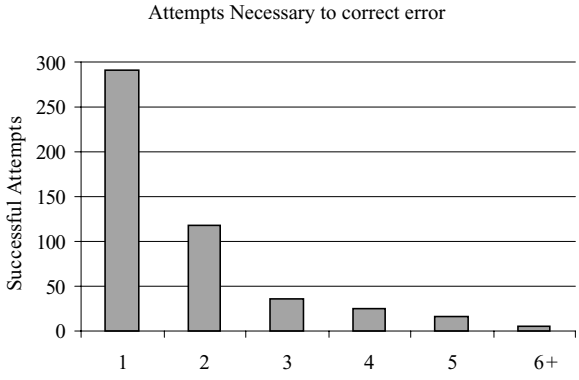

Figure 7. Number of attempts to correct an error.

199 of the substitution errors were not successfully corrected on the first attempt. 84 of these were unsuccessful redictations and 115 were attempts at using the alternates list when the correct word was not available in the list, while the soft- keyboard was always successful. It took two attempts to correct 118 of the errors, three attempts to correct 35 errors, four attempts to correct 25 errors, five attempts to correct 16 words, and six or more attempts to correct 5 errors (Fig. 7).

When the alternates list failed on the first attempt, participants switched to redictation 111 times (successful 67 times) and soft-keyboard 4 times (successful 4 times). When redictation failed on the first attempt, participants redictated a second time 50 times (successful 25 times), used the alternates list 18 times (successful 6 times), and used the soft-keyboard 16 times (successful 16 times) (Fig. 8).

Satisfaction Scores. For the satisfaction statement "I liked it", on average participants rated redictation highest at 6.7 (SD = 0.8), the soft-keyboard second highest at 4.8 (SD = 1.7), and the alternates list lowest at 4.5 (SD = 1.8). For the satisfaction statement "I had control", on average participants rated redictation highest at 6.3 (SD = 0.8), the soft-keyboard second highest at 6.0 (SD = 1.1), alternates list lowest at 5.5 (SD = 0.8) (Fig. 9).

For the statement "I liked it", there is a reliable one-way ANOVA, $F(2, 10) = 5.45$, $p < 0.05$. In post-hoc tests, redictation is rated reliably higher than the alternates list. For the statement "I had control", there is not a reliable one-way ANOVA, $F(2, 10) = 1.61$, $p > .05$. There are no reliable differences between any conditions.

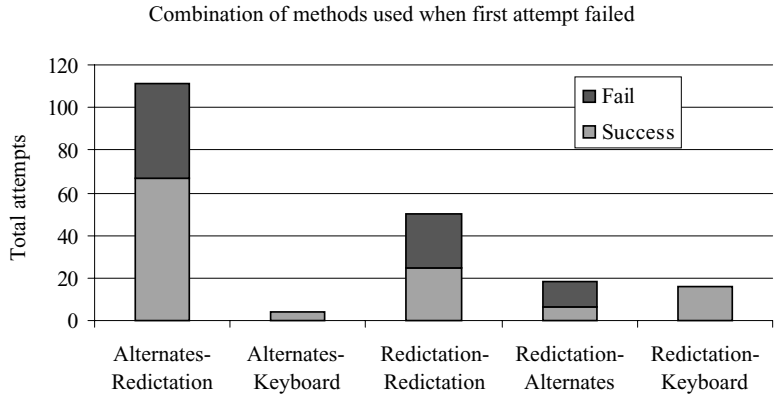

Figure 8. First two correction methods used when at least two were necessary.

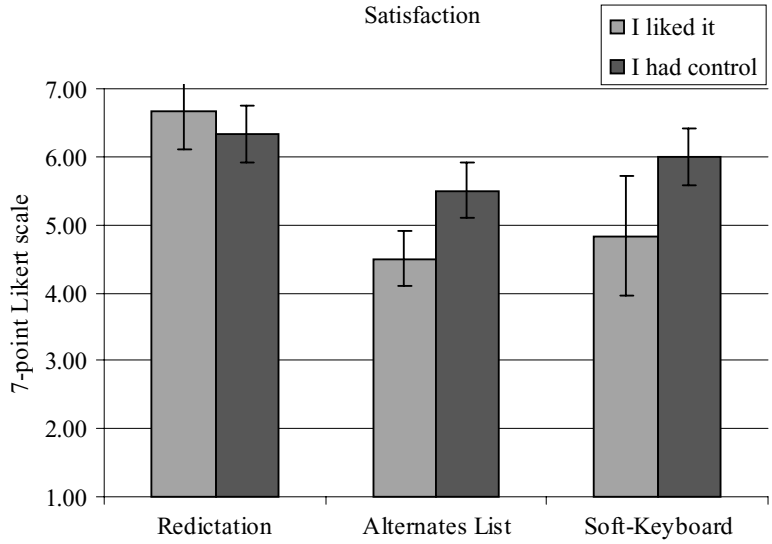

Figure 9. User satisfaction ratings to the two statements I liked it and I had control.

Discussion

Redictation was the most frequently used correction method in this study, and it was the most popular. Participants used redictation for initial correction 52% of the time and it accounted for 59% of the successes on the first try. Redictation also scored slightly higher than the other correction methods on the "I liked it" and "I had control" satisfaction statements.

The alternates list experienced far more use than had been seen in previous studies. In Halverson et al. (1999), the alternates list was only used in 8% of all corrections. In this study, participants used the alternates list as the initial correction method 42% of the time and attempted to use it at some point during 47% of all correction attempts. This increase in usage is attributable to the work that we did in the interface to facilitate its usage. Strong modes, push-to-talk, the correct command, and better alternates all worked toward making the alternates list a low cost correction mechanism.

A very popular combination of correction methods in this study was to try alternates first and then redictate. Of the 199 errors that took two attempts to correct, participants used that pair of correction methods 56% of the time. By themselves, both the alternates list and redictation are two-step procedures, but when going from the alternates list to redictation there is no need

to select the misrecognized word again. The participant can open the alternates list with a correct word command and examine the list. If the target word is not in the list, the participant can immediately move to redictating the target word. Redictating will cause the alternates list to close and the misrecognized word to be replaced with the result of the new dictation.

Surprisingly, while use of the alternates list was very high, user satisfaction with the alternates list remains about the same as in the first study. Participants liked redictation reliable better than the alternates list. This is a reflection of the accuracy of the alternates list. While improvements made it easier to use the alternates list, the correct word is in the list less than 40% of the time. Users are more likely to correct a word successfully with another correction mechanism.

Participants in this study were much faster at transcribing text than the participants in the first study. This is not surprising because participants in this study were allowed to switch from one error correction method to another as they saw fit. Oviatt (1999) showed that users are more successful at correcting when they switched modalities.

One potential problem with both of the studies presented in this paper is that participants were asked to perform transcription tasks. It is far more common for users to compose documents in real time than to transcribe documents that are already written out. It is possible that users find interface operation (e.g. mode switching) easier when transcription is the task because transcription takes less cognitive load than composition. Transcription tasks were used in these studies because it is an easier task to use in a lab environment and guarantee the corrections are made due to an error by the speech engine. It would be worthwhile to examine composition in future studies.

The interface enhancements discussed in this paper are neither appropriate for all users nor for all tasks. They were optimized for the average knowledge worker in a workplace environment. These enhancements are not intended for users engaged in other activities where their hands or eyes are not available. Similarly, this is also not an accessibility solution.

Conclusion

In the first study, we compared users' preference and performance for different error correction methods by comparing their exclusive use with four different methods. We found that participants performed the best with redictation and the alternates list, and preferred the alternates list. Participants struggled when using voice commands to navigate between errors, and this frequently led to a cascade of errors. We developed four interface improvements with the intent of increasing the usage of the alternates list and eliminating the problems that participants experienced with voice commands.

A second study examined users' experience with an improved speech interface on the Tablet PC. Usage of the alternates list increased dramatically from 8% in Halverson et al. (1999) to 47%. The combination of strong modes, push-to-talk, and the "correct <word>" command made the alternates list a light-weight mechanism that users were more willing to use to attempt corrections. It also facilitated mode switching, which is known to be a useful correction strategy. Mode switching was facilitated by allowing participants to redictate immediately while the alternates list was still open.

Participants still spend approximately 10 times longer correcting ASR errors than they are entering new text. This is the real barrier that keeps users from adopting dictation as a more viable text input method than the keyboard. Improving initial recognition accuracy is not the challenge anymore—we can only make incremental improvements here. We need radical improvements to help users correct the few words that ASR systems will continue to misrecognize for the near future. It should be a goal for systems to make correction easy enough that users spend less time correcting text than they initially spend composing it.

References

Ainsworth, W.A. and Pratt, S.R. (1992). Feedback strategies for error correction in speech recognition systems. *International Journal of Man-Machine Studies*, *36*(6):833–842.

Baber, C. and Hone, K.S. (1993). Modeling error recovery and repair in automatic speech recognition. *International Journal of Man-Machine Studies*, *39*(3):495–515.

Halverson, C., Horn, D., Karat, C., and Karat, J. (1999). The beauty of errors: Patterns of error correction in desktop speech systems. *Proceedings of INTERACT'99*. Amsterdam: IOS Press, pp. 133–140.

Huang, X., Acero, A., Chelba, C., Deng, L., Duchene, D., Goodman, J., Hon, H., Jacoby, D., Jiang, L., Loynd, R., Mahajan, M., Mau, P., Meredith, S., Mughal, S., Neto, S., Plumpe, M., Steury, K., Venolia, G., Wang, K., and Wang, Y. (2001). MIPAD: A Multimodal Interactive Prototype. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, pp. 9–12.

Karat, C., Halverson, C., Horn, and Karat, J. (1999). Patterns of entry and correction in large vocabulary continuous speech recognition

systems. *CHI'99: Proceedings of the CHI 99 Conference on Human Factors in Computing Systems: The CHI is the Limit*. New York: ACM Press, pp. 568–575.

Lai, J. and Vergo, J. (1997). MedSpeak. *CHI'97: Proceedings of the CHI 97 Conference on Human Factors in Computing Systems*. New York: ACM Press, pp. 431–438.

MacKenzie, I.S., Zhang, S.X., and Soukoreff, R.W. (1999). Text entry using soft keyboards. *Behaviour & Information Technology*, *18*:235–244.

Mankoff, J. and Abowd, G.D. (1999). Error correction techniques for handwriting, speech, and other ambiguous or error prone systems. Georgia Tech GVU Center Technical Report, GIT-GVU-99-18.

Oviatt, S. (1999). Mutual disambiguation of recognition errors in a multimodel architecture. *CHI'99: Proceedings of the CHI 99 Conference on Human Factors in Computing Systems: The CHI is the Limit*. New York: ACM Press, pp. 576–583.

Oviatt, S. and Van Gent, R. (1996). Error resolution during multimodal human-computer interaction. *Proceedings of the Fourth International Conference on Spoken Language Processing (ICSLP 96)*, pp. 204–207.

Suhm, B., Myers, B., and Waibel, A. (1999). Model-based and empirical evaluation of multimodal interactive error correction. *CHI'99: Proceedings of the CHI 99 Conference on Human Factors in Computing Systems: The CHI is the Limit*. New York: ACM Press, pp. 584–591.

Suhm, B., Myers, B., and Waibel, A. (2001). Multimodal error correction for speech user interfaces. *ACM Transactions on Computer-Human Interaction (TOCHI)*, *8*(1):60–98.