

# Using Confidence Scores to Improve Hands-Free Speech Based Navigation in Continuous Dictation Systems

JINJUAN FENG and ANDREW SEARS  
UMBC

---

Speech recognition systems have improved dramatically, but recent studies confirm that error correction activities still account for 66–75% of the users' time, and 50% of that time is spent just getting to the errors that need to be corrected. While researchers have suggested that confidence scores could prove useful during the error correction process, the focus is typically on error detection. More importantly, empirical studies have failed to confirm any measurable benefits when confidence scores are used in this way within dictation-oriented applications. In this article, we provide data that explains why confidence scores are unlikely to be useful for error detection. We propose a new navigation technique for use when speech-only interactions are strongly preferred and common, desktop-sized displays are available. The results of an empirical study that highlights the potential of this new technique are reported. An informal comparison between the current study and previous research suggests the new technique reduces time spent on navigation by 18%. Future research should include additional studies that compare the proposed technique to previous non-speech and speech-based navigation solutions.

Categories and Subject Descriptors: H.5.1 [**Information Interfaces and Presentation**]: Multimedia Information Systems—*Audio input/output*; H.5.2 [**Information Interfaces and Presentation**]: User Interfaces—*Voice I/O, screen design*

General Terms: Design, Experimentation, Human Factors

Additional Key Words and Phrases: Confidence score, navigation, simulation, speech recognition

---

## 1. INTRODUCTION

Speech recognition has improved dramatically in the past two decades [Karat et al. 2003]. However, recent studies involving dictation-oriented tasks suggest that recognition accuracy continues to be problematic when using

---

This material is based upon work supported by the National Science Foundation under Grant Nos. 9910607 and 0328391. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

Authors' address: Interactive Systems Research Center, Information Systems Department, UMBC (University of Maryland, Baltimore County), 1000 Hilltop Circle, Baltimore, MD 21250; email: {jfeng2, asears}@umbc.edu.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or direct commercial advantage and that copies show this notice on the first page or initial screen of a display along with the full citation. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, to redistribute to lists, or to use any component of this work in other works requires prior specific permission and/or a fee. Permissions may be requested from Publications Dept., ACM, Inc., 1515 Broadway, New York, NY 10036 USA, fax: +1 (212) 869-0481, or [permissions@acm.org](mailto:permissions@acm.org).

© 2004 ACM 1073-0616/04/1200-0329 \$5.00

state-of-the-art speech recognition under realistic conditions (e.g. Sears et al. [2001]). More importantly, several recent studies confirm that users of these speech recognition systems still experience significant difficulty correcting recognition errors [Karat et al. 1999; Sears et al. 2001]. In these studies, users spent as much as 66–75% of their time finding and correcting recognition errors. Further, a more detailed analysis confirms that users spend nearly 33% of their time (50% of the error correction time) simply navigating to the errors that need to be corrected.

When using a traditional graphical user interface to complete dictation-oriented tasks, the process of correcting recognition errors can be broken into three steps: users must determine that a recognition error has occurred (detection), navigate to the error, and then correct the error [Sears et al. 2003]. First, users must locate an error that needs to be corrected. Recognition errors result in the wrong, correctly spelled, word being inserted into the document. While this suggests that detecting recognition errors could be more difficult than detecting traditional typing errors, reports from users suggest that finding recognition errors is not particularly problematic [Sears et al. 2001].

After finding the error, users must convey this information to the system. Normally, this is accomplished by moving the cursor to the incorrect word via speech- or mouse-based navigation. Studies confirm that multimodal techniques tend to be superior to speech-only alternatives when completing this type of spatial task [Oviatt 1997; Oviatt et al. 2000]. Sears et al. [2001] found that experienced users spend nearly one-third of their time issuing speech-based navigation commands. While this certainly includes some time that would be more appropriately considered part of the detection phase, navigation is clearly a time consuming activity.

Finally, the focus shifts to removing the incorrect word and inserting the desired word. While many approaches exist, the most common technique is to delete the incorrect word and then redictate the desired word [Karat et al. 1999]. Other less frequently used techniques begin by selecting the incorrect word and then either redictating, spelling, or selecting the desired word from a list of alternatives. Sears et al. [2001] also found that users spent approximately one-third of their time removing incorrect words and inserting the desired word when using speech-based techniques.

These results clearly indicate that more efficient navigation and correction techniques are critical if speech recognition systems are to be widely accepted. Although multimodal techniques can be more effective than the speech only solutions, common multimodal solutions are not always viable options. For example, some individuals with physical disabilities have limited use of their hands but retain normal speech functions (e.g. some individuals with spinal cord injuries at or above C6 with American Spinal Cord Injury Association scores of A or B). For these individuals, hands-free speech-based solutions can allow for faster and more satisfying experiences than the traditional keyboard and mouse solutions that are available for the same population [Sears et al. 2001]. With these users in mind, as well as situations where the users' hands are unavailable due to conflicting tasks, our current focus is on more efficient hands-free interactions. In this article, we focus on the issues involved in navigating to the

errors that must be corrected during dictation-oriented activities, leaving the design and evaluation of more effective correction techniques as a focus for future research. Confidence scores are central to our approach. Confidence scores are numeric values generated by speech recognition engines that represent the level of certainty of the recognition engine that a particular word is correct. While many researchers have investigated the use of confidence scores, the primary focus has been on improving recognition accuracy (e.g. Setlur et al. [1996]; Kemp and Schaaf [1997]; Mou and Zue [2000]; Hazen et al. [2002]). Others discussed the use of confidence scores to support the detection of recognition errors, but these discussions typically focus on theoretical possibilities without reporting on the evaluation of implemented solutions (e.g. Chase [1997a]; Gillick et al. [1997]; Gunawardana et al. [1998]; Litman et al. [1999]; Bouwman et al. [2000]; Hazen and Bazzi [2001]; Maison and Gopinath [2001]). In one recent study, Suhm et al. [2001] did implement and evaluate a system that highlighted likely recognition errors using confidence scores. Unfortunately, their results suggest that confidence scores may not be effective for supporting error detection. Additional information regarding the nature of confidence scores, and the challenges that exist when using confidence scores to guide user interactions, is provided in the following section of the paper.

Our approach differs in that our goal is to help users navigate to, not detect, recognition errors. This shift in focus is motivated by the following reasons:

1. data confirming that the error detection and navigation can consume as much as one-third of the time experienced users spend creating documents [Sears et al. 2001],
2. data that indicate that users do not find error detection to be a major problem [Sears et al. 2001],
3. empirical results indicating that confidence scores may not prove effective for error detection [Suhm et al. 2001], and
4. our own analyses of confidence scores, reported below, which also suggest that confidence scores are not likely to be effective for error detection.

We describe a speech-based navigation technique that builds on the information available through confidence scores. This technique is intended for use with speech-based systems designed to support dictation-oriented activities using graphical user interfaces that are large enough to allow users to visually inspect the output of the speech recognition engine. The technique is useful when traditional pointing devices are either unavailable or inappropriate. For example, this approach may prove useful for individuals with physical disabilities that hinder the use of traditional pointing devices and for users working in environments where their hands are busy with other activities. We also present results from an empirical study that provides a preliminary assessment of the efficacy of this technique as well as directions for future research.

## 2. RELATED RESEARCH

Two areas of research are directly relevant to the current work: navigation and the use of confidence scores to facilitate the correction of recognition error.

While we acknowledge the potential benefits of multimodal solutions when specifying recognition errors, our current focus is on speech-based solutions. As a result, we do not attempt to provide a comprehensive review of the research on multimodal solutions. For a comprehensive review of this literature, we refer the reader to Oviatt [2003]. Similarly, there is a significant body of research on confidence scores, but most of these activities focus on improving the accuracy of the recognition algorithm as opposed to changing the way users interact with the resulting applications. As mentioned above, several researchers do discuss the possibility of using confidence scores to support the detection of recognition errors, but few have gone beyond theoretical discussions and even fewer have implemented and evaluated possible solutions. As a result, we provide only a brief overview of this literature to illustrate the nature of the existing research on confidence scores, as well as a detailed review of the two articles that are most directly related to the current project.

## 2.1 Navigation

Several researchers have evaluated alternative error correction processes. For example, Suhm et al. [2001] investigated several multimodal approaches for correcting recognition errors, including use of the keyboard, mouse, a stylus, and speech. While the solutions were multimodal, navigation was always accomplished using a touchscreen with other modalities (e.g. speech or keyboard). Similarly, Danis et al. [1994] reported on a speech-oriented editor where users could “Point and Speak” to change the insertion point while dictating, but cursor movements were accomplished using a mouse. Other studies also confirm that multimodal solutions tend to be superior to speech-only solutions for spatial navigation tasks such as selecting a word within a document [e.g. Oviatt 1997; Oviatt et al. 2000].

McNair and Waibel [1994] also explored error correction activities, but focused explicitly on speech-based selection of incorrect words. They described a target-based approach, where users spoke the word that needed to be corrected, with a success rate of 85%. Similar techniques are available in most commercial speech recognition applications. Recent studies confirm that 85% success continues to be representative of the performance of this target-based approach to correcting recognition errors (e.g. Sears et al. [2003]). More importantly, these studies confirm that failed target-based navigation commands resulted in significant difficulties for users as they worked to correct recognition errors (e.g. Sears et al. [2003]).

Manaris and Harkreader [1998] investigated the use of speech recognition as an alternative mechanism for generating keystrokes and mouse events with the goal of developing an alternative data entry technique for individuals with upper-body motor-control impairment. While several navigation mechanisms were implemented, and a pilot study was conducted using a Wizard-of-Oz simulation, no results are reported regarding the effectiveness of their speech-based navigation mechanism.

de Mauro et al. [2001] discussed the design of a voice-controlled mouse, which supported several different navigation mechanisms. However, navigation

within textual documents was only supported using a set of commands that required users to learn nonintuitive mappings between commands (e.g. “Move left”) and utterances that caused commands to be executed (e.g. “A”). No data were provided regarding the efficacy of this navigation mechanism.

Christian et al. [2000] explored speech-based navigation in the context of the web. Navigation was accomplished by speaking the words that served as links or numbers, generated by the browser, that were associated with these links. While errors were minimal, this approach required significantly longer time than traditional mouse-based navigation. More importantly, this task is not representative of the activities involved when users navigate to recognition errors within textual documents.

In a recent study, Sears et al. [2001] confirmed that experienced users spent nearly one-third of their time issuing speech-based navigation commands when completing standard dictation tasks using state-of-the-art speech recognition software. An analysis of the users’ navigation activities confirmed that users experienced significant difficulty with all of the navigation commands and significant effort was required to recover from some of the resulting consequences (e.g. when the content of the document was inappropriately altered).

While multiple researchers have investigated the issues involved in specifying recognition errors, few have focused on speech-based specification techniques. Results from those studies that did involve speech-based specification techniques suggest that the techniques are error prone and time consuming.

## 2.2 Facilitating Error Correction Using Confidence Scores

Confidence scores are recognizer dependent numeric values generated by speech recognition engines that represent how “confident” the recognition engine is that a particular word is correct. Confidence scores are computed using a variety of metrics that may differ from one recognition engine to another (see Chase [1997a] for a discussion). At the same time, researchers have confirmed that there is a strong relationship between confidence scores and recognition errors (e.g. Gillick et al. [1997]). In general, correctly recognized words tend to have higher confidence scores and recognition errors tend to have lower confidence scores. However, a significant percentage of the recognition errors will have relatively high confidence scores and some correctly recognized words will have low confidence scores. The absolute value of the confidence scores generated by different speech recognition engines may differ, but the general pattern described above appears to hold true regardless of the recognition engine being used (e.g. Bouwman et al. [1999]; Suhm et al. [2001]). Therefore, interaction mechanisms employing confidence scores are likely to require adjustment as they are ported from one recognition engine to another. At the same time, confidence scores tend to exhibit similar characteristics, regardless of the recognition engine being used. This suggests that, with appropriate evaluation and tuning, it will be possible to port confidence score-based interaction mechanisms between recognition engines. Of course, the effectiveness of the technique may vary when different recognition engines are used. As a result, it would be useful to evaluate the sensitivity of confidence score-based interaction

mechanisms to the quality of the information available through the confidence scores. Some techniques may be robust, porting between recognition engines with minimal work. Others may be highly sensitive, making their transfer to a new recognition engine more difficult.

Confidence scores are the focus of extensive research. Importantly, confidence scores have been used successfully to detect likely recognition errors in a variety of telephony applications. However, it is important to note that these applications tend to employ relatively small vocabularies. In the area of speech-based dictation, where the vocabulary is substantially larger, most confidence score research has focused on increasing the accuracy of the recognition algorithm as opposed to helping users find and correct recognition errors. For example, Chase [1997b] investigated using a variety of confidence scores to predict the occurrence and type of recognition errors with the goal of providing feedback to the recognition algorithm. Gillick et al. [1997] computed confidence scores using a probabilistic approach and established a model to predict the correctness of individual words. Gunawardana et al. [1998] extended the commonly used filler model for word-based acoustic confidence measures and compared the score of the word in question to the scores for words that are commonly confused with it. This approach reduced the false acceptance rate by 39% relative to the original filler model. Litman et al. [1999] used confidence measures to detect poor speech recognition at the dialogue level, resulting in a significant performance improvement. Bouwman et al. [2000] suggested that weighting different aspects of phone confidence measures can improve early detection of errors while Hazen and Bazzi [2001] combined confidence measures and out-of-vocabulary word detectors to detect errors. Each of these studies provides insight into possible uses of confidence scores, but all the studies focus on generating information that can provide feedback to the systems, developers, or system administrators as opposed to the users of the system.

Chase [1997a] did suggest that comparing confidence scores to a predefined threshold could be useful for detecting likely recognition errors, but the impact of varying the threshold was not considered and the suggested technique was not implemented or evaluated. In contrast, Suhm et al. [2001] implemented and evaluated a system that used confidence scores to highlight possible recognition errors. As suggested by Chase, a threshold was used to separate those words that were likely to be correct from those that were likely to be incorrect. The threshold was set such that classification errors were minimized, with classification accuracy defined as the percentage of words that were correctly classified (i.e. correct words marked as being correct and incorrect words marked as being incorrect). A classification accuracy of 89% was achieved using a threshold of 0.6, resulting in words with confidence scores below 0.6 being highlighted as likely recognition errors. Results from an empirical study indicated that automatically highlighting likely recognition errors actually slowed down the overall correction process. This finding suggests the confidence scores used in this system were not sufficiently reliable to support recognition error detection.

To date, no published reports of successful, implemented, systems use confidence scores to assist in the process of finding and correcting recognition errors. Efforts by Suhm et al. [2001] to facilitate the detection of recognition errors were

not successful. This, combined with data suggesting that users experience more difficulty during the specification phase of this process, motivated the activities described below.

### 3. RESEARCH OBJECTIVE

As discussed above, numerous researchers have discussed the use of confidence scores for detecting possible recognition errors (e.g. Chase [1997a]; Gunawardana et al. [1998]; Suhm et al. [2001]). The basic approach is simple: compare the confidence score of each word to an established threshold. Words with confidence scores below the threshold are considered likely to be incorrect. Some of these words are actually incorrect (i.e. correct detections), but others are correct words that happen to have low confidence scores (i.e. false alarms). Words with confidence scores equal to, or greater than, the threshold are considered likely to be correct. Similarly, some recognition errors will have confidence scores greater than the threshold and will be considered likely to be correct (i.e. missed detections). The number of false alarms and missed detections can be manipulated by varying the threshold; numerous techniques can be used to compute confidence scores, to set thresholds, and to compare confidence scores to these thresholds.

Our long term goal is to investigate the use of confidence scores to support the process of detecting recognition errors, navigating to those errors, and completing the required correction. Our initial goal, which motivates the current study, is to provide more effective support for navigation activities. Our investigation begins with an existing collection of approximately 67,000 spoken words with confidence scores. We begin by discussing the origin of these data. Next, we analyze the use of confidence scores to detect recognition errors. This is followed by a discussion of how confidence scores could be used to facilitate the specification of recognition errors. We present a new speech-based navigation technique that uses confidence scores to expedite the navigation process. We conclude by presenting results from a simulation and an empirical study that confirm the efficacy of our new technique.

### 4. CONFIDENCE SCORE DATA

Our investigation builds on data gathered as part of an earlier study [Feng et al. 2003]. In this study, 15 participants composed 120 documents using a custom speech recognition application (TkTalk Version 2.0) that employed IBM's ViaVoice speech recognition engine (Millennium Edition). For each task, participants were provided with a general topic to discuss as well as specific issues they may want to address in their response. Participants had to compose responses that addressed the general topic, but their responses did not have to address each of the specific issues that were mentioned in the task description.

In ViaVoice, each word is assigned an integer confidence score. The word output by the speech engine as the "most likely" alternative is actually selected based upon a variety of factors including the confidence scores of individual words as well as language models that consider the surrounding words. In addition, ViaVoice can generate alternatives that may be correct

if the “most likely” word is actually incorrect. The first alternative generated is considered the “best alternative” if the “most likely” word is not correct. Since multiple factors are considered when selecting the “most likely” word that a user spoke, the confidence score associated with this “most likely” word is occasionally lower than the confidence score associated with the “best alternative.”

When the 15 participants completed the study, the resulting 120 documents contained a total of 66950 words. This includes 55495 words that were correctly recognized and 11455 recognition errors (82.9% recognition accuracy). Confidence scores ranged from approximately  $-15$  to near  $+30$ . We recorded both the “most likely” word and the “best alternative” along with their confidence scores. As expected, in a few cases the speech engine did not generate any alternatives.

## 5. DETECTING RECOGNITION ERRORS

Most discussions regarding the use of confidence scores to facilitate the process of detecting, navigating to, and correcting recognition errors have focused on assisting the user in detecting likely recognition errors. Suhm et al. [2001] provide the only example of an evaluated implementation of this idea. They used classification accuracy to set the threshold. Again, classification accuracy is defined as the percentage of words that are classified correctly. The number of classification errors is the sum of false alarms and missed detections. Since the number of correct words is typically much larger than the number of recognition errors, using classification accuracy to set the threshold results in a bias toward minimizing false alarms.

### 5.1 Using Raw Confidence Scores

When discussing the use of confidence scores to detect recognition errors, the focus is typically on the raw confidence score associated with the “most likely” word generated by the speech engine. Using the raw confidence scores from our data set, and setting the threshold at  $-7$  maximizes classification accuracy. As illustrated in Table I and Figure 1, maximizing classification accuracy results in only 27% of the recognition errors being identified (73% missed detections). Classification accuracy is maximized because almost 98% of the correctly recognized words are classified properly (correct rejections). However, this threshold is unlikely to be effective if the goal is to support users in the detection of recognition errors.

A higher threshold is required to detect a reasonable fraction of the recognition errors. A threshold of  $-2$  increases recall such that over 50% of the recognition errors are identified, but false alarms also increase as evidenced by the decrease in precision. A threshold of 1 results in approximately two-thirds of the recognition errors being identified, but precision drops to only 37.4. These results highlight the difficulties that exist when trying to support recognition error detection using threshold-based approaches with raw confidence scores.

Table I. Recall, Precision, and Classification Accuracy when Using Various Confidence Score Thresholds. Recall Represents the Percentage of Recognition Errors Identified. Precision Represents the Percentage of Words Identified as Likely Recognition Errors that Were Actually Recognition Errors. Classification Accuracy is the Percentage of Words that Were Classified Correctly

Threshold	Recall	Precision	Classification Accuracy
-10	14.7	70.5	84.4
-9	18.2	67.5	84.5
-8	22.2	64.6	84.6
<b>-7</b>	<b>26.8</b>	<b>61.8</b>	<b>84.6</b>
-6	31.5	58.9	84.5
-5	36.2	56.0	84.2
-4	41.2	53.3	83.8
-3	46.0	50.3	83.0
<b>-2</b>	<b>50.8</b>	<b>47.8</b>	<b>82.1</b>
-1	55.1	45.1	80.8
0	62.7	39.7	77.3
<b>1</b>	<b>66.3</b>	<b>37.4</b>	<b>75.3</b>

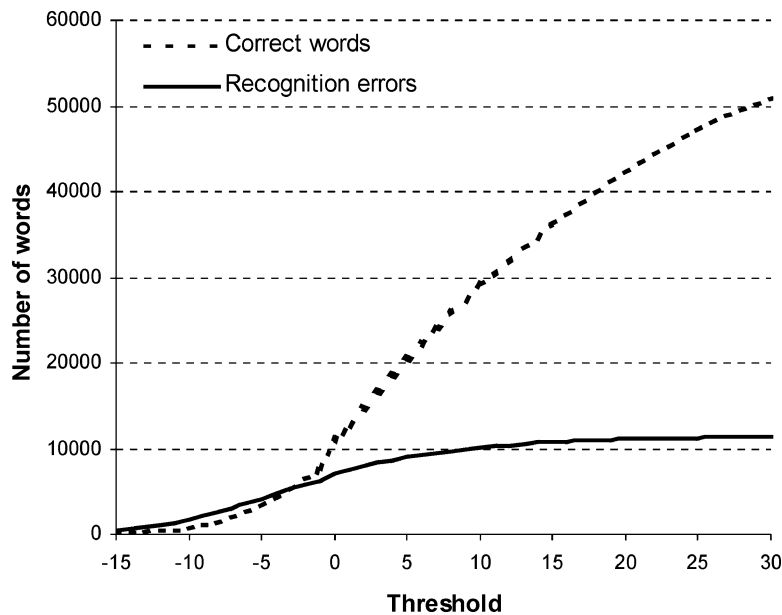


Fig. 1. Number of correct words selected and recognition errors selected using various thresholds for raw confidence scores.

## 5.2 Using Differences in Confidence Scores

While the previous analysis investigated the use of the raw confidence score associated with the “most likely” word, this section focuses on comparing the confidence scores of the “most likely” word and the “best alternative.” The underlying logic is that a large difference between these two words should imply greater certainty that the “most likely” word is correct. We acknowledge

Table II. Recall, Precision, and Classification Accuracy when Using Various thresholds for Differences in Confidence Scores

Threshold	Recall	Precision	Classification Accuracy
-8	4.2	59.3	83.1
-7	6.1	58.7	83.2
-6	8.7	56.8	83.2
<b>-5</b>	<b>12.0</b>	<b>55.5</b>	<b>83.3</b>
-4	16.4	53.6	83.3
-3	21.5	51.1	83.0
-2	27.7	48.8	82.7
-1	34.6	46.8	82.1
0	43.5	43.8	80.8
<b>1</b>	<b>52.5</b>	<b>41.0</b>	<b>79.0</b>
2	61.1	38.4	76.6
<b>3</b>	<b>68.7</b>	<b>35.7</b>	<b>73.5</b>

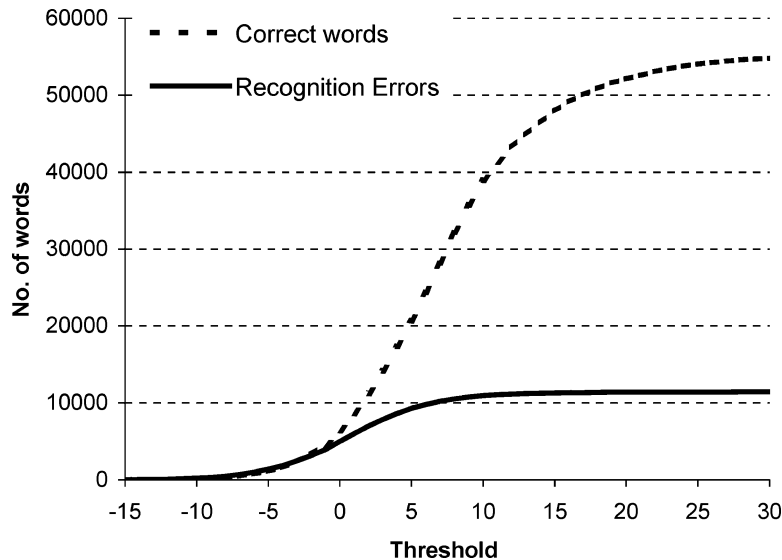


Fig. 2. Number of correct words selected and recognition errors selected using various thresholds for differences in confidence scores.

a strong relationship between confidence scores and the difference scores we are computing as well as the fact that some recognition algorithms use similar computations internally as part of the recognition process. However, we still believe these “difference” scores may prove useful, especially when more complex techniques, such as machine learning, are used.

When using the difference in confidence scores, classification accuracy is maximized at a threshold of  $-5$  with the current data set. As illustrated in Table II and Figure 2, maximizing classification accuracy results in only 12% of the recognition errors being identified (88% missed detections). In this case, classification accuracy is maximized because over 98% of the correctly

recognized words are classified properly (correct rejections). To detect at least 50% of the recognition errors (i.e. recall  $>50$ ), the threshold must be increased to 1, but this results in precision dropping to 41.0 (i.e. 59% false alarms). Over two-thirds of the recognition errors can be identified with a threshold of 3, but precision continues to decrease.

These results suggest that neither raw confidence scores nor differences in confidence scores are likely to be effective if the goal is to facilitate the detection of recognition errors. At the same time, the differences in the results obtained when using raw confidence scores and differences in confidence scores (compare Figures 1 and 2) suggest that any additional exploration should include both approaches. Of course, the algorithms used to compute confidence score could possibly be improved such that recognition error detection, using either raw scores or differences, could be more effectively supported. An existing confidence score algorithm, different from the one used in the current study, could also possibly provide better results but there are no published reports that would suggest that a substantial improvement is possible with existing confidence score algorithms. Improved confidence score algorithms, either existing or developed in the future, would be welcome as they may not only support recognition error detection but they would likely improve our proposed speech-based navigation mechanism, which is described below.

## 6. NAVIGATING TO RECOGNITION ERRORS

The inability to facilitate recognition error detection activities using either raw confidence scores or differences in confidence scores, combined with data suggesting that navigation is a more significant problem for users, motivated a shift in attention from detection to navigation. In the discussion that follows, we assume that the users are responsible for detecting the word to be corrected and that the goal is to facilitate the users' efforts to navigate to the word they want to correct.

We begin by analyzing the distribution of recognition errors. Next, we describe the design of a new speech-based navigation technique intended to facilitate the navigation process. This technique integrates information about confidence scores, the distribution of recognition errors, and results from our earlier efforts to improve speech-based navigation.

### 6.1 Recognition Error Sequences

We begin by analyzing the distribution of recognition errors to determine if errors tend to occur in clusters or separately. A recognition error sequence is defined as one or more consecutive words that are recognized incorrectly. In our sample of almost 67000 words, the length of recognition error sequences varied between one and twenty words. Figure 3 illustrates the distribution of recognition error sequences by length.

As illustrated in Figure 3, less than 30% of the recognition errors occur in isolation. More importantly, over 70% of the recognition errors appear immediately adjacent to at least one other recognition error. This provides new opportunities given the goal of navigating to, as opposed to detecting, recognition

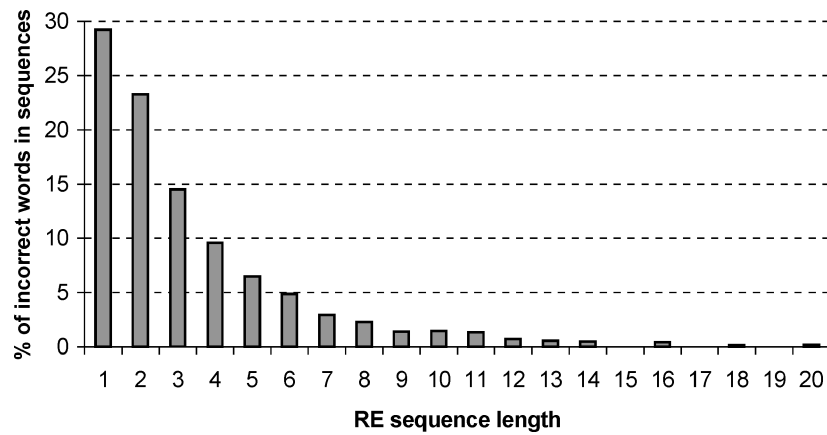


Fig. 3. Percentage of incorrect words in recognition error sequences of different lengths.

errors. Under these circumstances, helping users navigate to a recognition error sequence may be sufficient.

## 6.2 Navigation Anchors

Our earlier studies of speech-based navigation highlighted the potential benefits and important limitations of simplified direction-based navigation (e.g. move up). Given short, fixed, phonetically distinct commands, failure rates can be minimized [Feng et al. 2003]. The disadvantage of these commands is that they allow the user to move the cursor just one word or line at a time. Target-based navigation (e.g. select Friday) allows users to move the cursor larger distances with a single command as do some more powerful, but error prone, direction-based navigation commands (e.g. move right four words). Since these target-based commands typically allow users to select any word that is visible on the screen, the number of possible targets, which determines the vocabulary size, can be quite large. The difficulty of command construction (i.e. determining exactly which words to say) is also significantly higher than simple direction-based commands. As a result, recognition errors are quite common, causing 10–20% of all target-based navigation commands to fail [Feng et al. 2003].

To summarize:

- Short, fixed, phonetically distinct navigation commands minimize errors, but tend to provide limited power [Feng et al. 2003].
- Longer, or constructed, commands can provide greater power, but tend to be associated with higher failure rates [Feng et al. 2003].

Our goal is to design a speech-based navigation technique that:

- allows for greater efficiency than the simple direction-based navigation discussed above, and
- minimizes failure rates by employing short, fixed, navigation commands.

Navigation anchors are central to the proposed solution. Strategically selected words within the document will serve as navigation anchors. Using one of two short, fixed, phonetically distinct commands (i.e. next and previous) that should rarely fail, users can easily move to the nearest anchor that precedes or follows the current cursor location. Since the navigation anchor may not be the exact word the user wants to correct, our technique also provides four short, fixed, direction-based navigation commands that allow users to move to adjacent words or lines (i.e. move up, move down, move left, and move right).

When compared with existing target- and direction-based navigation, next and previous appear to be hybrid commands. They can be viewed as target-based commands where the target is predefined by the system (i.e. the nearest navigation anchor in the correct direction). They could also be viewed as direction-based commands where the system defines the distance each time the command is issued. The key is that they are short, fixed, phonetically distinct commands and therefore, they should be highly reliable.

While developers of commercial speech applications have worked to reduce the number of modes users must employ, some special modes still exist. Examples include modes to support navigation (e.g. some voice-mouse techniques) or certain editing activities (e.g. a correction dialog box could be viewed as a mode). Modes can be useful in speech applications since they may allow the size of the vocabulary to be reduced and this typically results in lower failure rates. We propose a system that uses a new “proof document” mode. The intended method for using this technique involves dictating a portion of the document and then proofreading the resulting text to correct errors. While users could dictate just a word or two before the proofreading operation, this would be inefficient. Users could also dictate multiple pages before proofreading, but they would be more likely to forget some relevant details and the proofreading process could be more difficult. We anticipate that users will dictate a paragraph or two, proofread the resulting text, and then continue dictating. The exact quantity of text dictated before proofreading is likely to evolve with experience and will probably depend on the recognition accuracy individual users experience. The proposed system is designed to be flexible, allowing users to proofread their dictation as frequently, or infrequently, as desired.

On the surface, there are two arguments against introducing a proofreading mode. First, speech recognition engines often employ contextual information when deciding which words the user most likely spoke. This may come in the form of trigrams, which indicate how likely a specific set of three words are to occur in a particular order. As a result, one recognition error could lead to cascading errors as subsequent words are recognized in an incorrect context. In contrast, it has been suggested informally that speech recognition users are better served by dictating larger quantities of text. The underlying justification is that this results in users speaking more fluently, thereby reducing recognition errors. In fact, as our results will illustrate, which of these two approaches an individual chooses does not seem to have an impact on the number of recognition errors. In our earlier study, the recognition error rate was approximately 17% [Feng et al. 2003]. As illustrated below, participants in the current study

completed the same tasks using our new proofreading mode with a recognition error rate of 17%.

A second argument against a proofreading mode could be that it would result in an unnatural approach to completing the document generation task. However, Karat et al. [1999] found that speech-based interactions are profoundly different from keyboard/mouse input. When producing text documents, users employ two approaches for correcting errors. With inline corrections, errors are corrected as they occur. With delayed corrections, users continue producing more text after an error and navigate back to correct errors at a later time. Karat et al. [1999] found that inline corrections were quite common when people created documents using a keyboard and mouse. Results were quite different when users created documents with speech recognition. Novices corrected some errors as they occurred, but numerous errors were corrected during a proofreading process. Interestingly, as users gained experience, inline corrections became less common and correcting errors during a proofreading phase became more common. As this result was explored, it became clear that users of speech technologies do not have a clear model of when errors are likely to occur. As a consequence, they must either constantly monitor the display for errors or rely more heavily on a proofreading pass to detect and correct errors.

The question that remains is how to specify navigation targets such that users can move between recognition errors efficiently. While confidence scores do not appear likely solutions when trying to detect errors, they may prove useful for defining navigation anchors to aid in the specification of errors. This belief is based, in part, on the way recognition errors are clustered. As discussed above, over 70% of all recognition errors appear immediately adjacent to another recognition error. This suggests that a technique that can successfully detect a subset of the recognition errors may prove useful when defining navigation anchors. Interestingly, about 5900 carefully selected navigation anchors could allow for efficient navigation to all 11455 recognition errors in the current data set. In the next section, we discuss our method for defining navigation anchors.

### 6.3 Defining Navigation Anchors

The goal when defining navigation anchors is to facilitate the process of navigating to the words that must be corrected, not necessarily to detect recognition errors. Navigation anchors may, or may not, also be recognition errors. In fact, our analysis suggests that it is highly unlikely that a one-to-one correspondence between navigation anchors and recognition errors could be established at this point. However, some navigation anchors are likely to be recognition errors, which will simplify navigation to those particular errors. This may also simplify navigation to some other errors since we know that more than 70% of all recognition errors are immediately adjacent to other recognition errors. Clearly, navigation anchors will be more useful if they are near recognition errors. To effectively analyze the efficacy of a set of navigation anchors, we must consider how they will be used. Therefore, we developed a simulation to determine which

commands would be required to navigate to a predefined set of recognition errors.

For our preliminary analysis, we used the data described earlier. This provided 120 documents including approximately 67000 words with a recognition error rate of approximately 17%. The recognition errors that served as navigation targets for our simulation included both substitutions (where the incorrect word was displayed on the screen) and deletions (where the word was omitted from the resulting document). The simulation was designed as follows.

While the proposed navigation mechanism includes six commands (i.e. *next*, *previous*, *move up*, *move down*, *move left*, *move right*), only three were used in the initial simulation: *next*, *move right*, and *move left*. Since we do not anticipate that “*move up*” and “*move down*” will be used often and their effects are determined by the screen size and font size, these commands were not included in the initial simulation. Since “*previous*” is only useful if the “*move down*” command is available, this command was also excluded from the initial simulation.

When proofreading a document, navigation anchors are highlighted and the cursor is positioned such that it corresponds to the first navigation anchor. For the simulation, the cursor is moved to each recognition error, in the order they appear in the document, until the final error is reached. Three approaches can be used when moving to a recognition error:

1. The “*move right*” command is issued repeatedly until the next recognition error is highlighted.
2. The “*next*” command is issued one or more times until the cursor moves to the last navigation anchor before the next recognition error, or to the recognition error itself if that is also a navigation anchor. Then “*move right*” is used as necessary to move the cursor to the recognition error.
3. The “*next*” command is issued one or more times until the cursor moves to the first navigation anchor following the next recognition error. Then “*move left*” is used as necessary to move the cursor to the recognition error. A variant of this approach, which only involves the “*move left*” command, is used to select any recognition errors that may exist before the first navigation anchor. The “*move left*” command addresses situations where the cursor must be moved backwards from a navigation anchor to a recognition error.

All three solutions are evaluated, with the results being recorded for the approach that requires the fewest commands. When two or more solutions require the same number of commands, the following preferences are used to break ties. “*Next*” is preferred over both “*move left*” and “*move right*” for two reasons. First, we anticipate a lower failure rate for “*next*” because it is a single word and does not share the common word “*move*” with the other commands. Second, given an observed recognition error rate of 17%, a “*move*” command should place the cursor on a recognition error only 17% of the time. In contrast, we anticipate much higher precision for our navigation anchors. Assuming the use of the “*next*” command does not break the tie, we prefer solutions that use “*move right*” instead of “*move left*.” This is due to the simple fact that “*move left*” means that the cursor was moved beyond the target and is being moved

back to the target. We believe this will be less natural than alternatives that use “move right.”

In addition to the 11455 recognition errors for which we have confidence scores, 677 more words were omitted from the resulting documents due to recognition errors. When words are omitted, we expect users to navigate to the word immediately before or after where the omitted word should have been. The simulation was designed to navigate to omitted words in the same way. Therefore, our simulation navigates to 12132 targets within a set of 66950 words.

Three approaches were evaluated for defining navigation anchors. The first method uses the raw confidence score (CS) of the “most likely” word. Thresholds of  $-6$  to  $+2$  are reported for raw confidence scores. The second uses the difference between the confidence score (DCS) of the “most likely” word and the “best alternative.” Thresholds of  $-2$  to  $+3$  are reported for differences in confidence scores. A third approach, where every  $n$ th word (Fixed) is used as a navigation anchor, was included for comparison purposes. Results are reported for values of  $n$  ranging from 2 to 10. We did not anticipate that the “fixed” strategy would be selected for use, but included it to allow the potential benefits of the CS and DCS techniques to be highlighted. The key difference is that the CS and DCS techniques attempt to exploit the limited information that is available through confidence scores while the fixed approach does not.

We began by using two major criteria to evaluate the different technique/threshold combinations. First, we considered the number of successful commands required to reach all recognition errors. Clearly, an approach that requires fewer commands has the potential to be more efficient. Next, we examined the composition of the navigation commands required. While we evaluated the number of commands first, the specific commands that are required can have a significant impact on the efficacy of any given solution. For example, in earlier studies we confirmed that different types of commands result in different failure rates and that the consequences of these failures can vary dramatically between commands [Sears et. al. 2003]. Assuming that a similar number of commands is required, higher failure rates or more severe consequences can result in less efficient interactions and decreased satisfaction. As discussed above, solutions that make greater use of the “next” command are preferred. Assuming “next” is used with equal frequency, we prefer solutions that use “move right” over those that use “move left.” The importance of command composition is further illustrated below in the analysis of failure rates from our empirical study.

We do not claim that these are the only two criteria that should be considered, and we acknowledge that other criteria may ultimately prove more important. In this paper, we present these criteria as those that we believe will ultimately prove valuable when deciding how to define navigation anchors. In the sections that follow, we use these criteria to select a specific technique/threshold combination for further investigation. Given the difficulties others have experienced using confidence scores to support error correction activities, we aim to demonstrate the efficacy our new speech-based navigation technique. We do not view this as a definitive evaluation of this technique and acknowledge that additional studies will be required to compare it to other solutions, explore

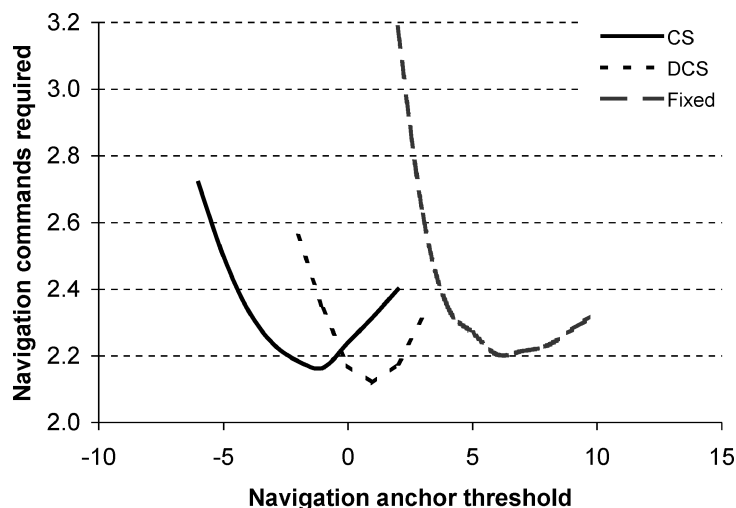


Fig. 4. Average number of navigation commands required when using different navigation anchor thresholds and techniques.

alternative methods of defining navigation anchors, and evaluate the effectiveness of the technique when integrated into a complete dictation system.

#### 6.4 Number of Commands Required

Figure 4 illustrates the average number of commands required for per target when using various thresholds with each of the three anchor-definition techniques described above. All three techniques result in a “U” shaped curve, confirming that a threshold that is too high or too low would result in additional commands being required.

The DCS technique, with a threshold of 1, minimized the number of navigation commands required (2.12 commands on average). Since the recognition errors cannot be considered independent events, we used the Wilcoxon Z score pairwise non-parametric test and identified four additional thresholds that provided results that did not differ significantly from this minimum. The four additional thresholds included the threshold of 0 when using the DCS technique and the thresholds of  $-3$ ,  $-2$ , and  $-1$  when using the CS technique. The average number of navigation commands required ranged from 2.12 to 2.24 for this set of five thresholds.

Since the number of data points for the statistical analysis is extremely large, the statistical test is sensitive to small differences. For example, the difference between the thresholds of 1 and 2 using the DCS technique is statistically significant. However, for a document of 400 words (the average size of the documents created by our participants) with 17% recognition error rate, increasing the threshold to 2 results in less than one additional command being required to reach all of the recognition errors. Since users are extremely unlikely to notice such small differences, we expanded the set of technique/threshold combinations that we investigated to include all combinations where the average

Table III. Thresholds that Provide Similar Average Navigation Distance

Technique/Threshold	Average Number of Navigation Commands Required
CS/-3	2.24
CS/-2	2.18
CS/-1	2.17
DCS/0	2.17
DCS/1	2.12
DCS/2	2.18
Fixed/6	2.20
Fixed/7	2.21
Fixed/8	2.23

number of commands is less than or equal to 2.24 (the upper bound of our original set of solutions). This provided a set of nine technique/threshold combinations to consider as illustrated in Table III.

### 6.5 Command Composition

To select a specific threshold to use for our initial implementation, we analyzed the commands used for each technique/threshold combination. As discussed above, the “next” command is preferred over either of the “move” commands. As illustrated in Table IV, the DCS/2 combination provided the best results with “next” accounting for over 71% of all of the commands issued and “move left” accounting for less than 6% of the commands. Given these results, we employed the DCS/2 combination to define navigation anchors in the current investigation. However, it is important to note that given the magnitude of the differences that exist between some of the technique/threshold combinations, we do not believe that users would necessarily be able to differentiate between all of the possible solutions. In fact, we believe it is unlikely that any single solution will prove optimal. Instead, we assert that a set of techniques can be used to define navigation anchors that will allow for more effective interactions as compared to those that result when other techniques are used to define navigation anchors. Through our analysis, we suggest that the number of commands as well as the composition of those commands will help determine the efficacy of a given technique/threshold combination. Given our current analysis, we present the DCS/2 combination as one approach that likely belongs to the set of solutions that proves to be effective. Determining which other solutions will also allow for effective interactions, and which will not, will require multiple empirical comparisons with other technique/threshold combinations.

## 7. AN EMPIRICAL EVALUATION

The purpose of this study was to provide a preliminary evaluation of the proposed navigation technique. This can only be viewed as a preliminary evaluation for three reasons. First, our participants completed the error detection and navigation activities, but did not actually correct the errors. While we recognize the limitations that result from this decision, we felt it was necessary given the

Table IV. Command Composition Using Different Technique/Threshold Combinations (Represented as Percentages of the Total Number of Commands Required)

Technique/Threshold	Next	Move Right	Move Left
CS/-3	42.4	44.5	13.1
CS/-2	49.3	39.1	11.5
CS/-1	56.2	33.9	9.8
DCS/0	47.2	40.8	12.0
DCS/1	60.3	31.3	8.4
DCS/2	71.1	23.3	5.6
Fixed/6	46.0	46.3	7.7
Fixed/7	39.1	54.3	6.6
Fixed/8	35.9	53.0	11.1

additional complexities that exist when users complete the entire error correction process. More specifically, existing solutions for actually correcting the errors often result in cascading errors which, in turn, result in additional navigation and error correction activities. At this point, given the difficulties that other researchers experienced using confidence scores to support error correction activities, we aim to provide an initial validation of this approach. If our results confirm that users can navigate within their documents effectively, additional studies will investigate alternative techniques for defining navigation anchors as well as approaches for integrating improved error correction support into our navigation technique.

Second, we chose a navigation anchor selection technique and threshold based upon our expectations of how users will react to this interface. Since it is possible that user reactions will differ from our expectations, a different selection technique or threshold may result in improved performance. While we believe that results from the current study will provide insights into the efficacy of this technique, additional studies will be required to further evaluate alternative technique/threshold combinations. If the current study provides negative results, this may indicate that the approach itself is flawed or that a different technique/threshold combination is required.

Third, our participants only interacted with this new navigation technique twice: one practice session and one experimental session. Our results represent novice performance and any comparisons with earlier studies must take this into consideration.

## 7.1 Methods

**7.1.1 Participants.** Twelve participants took part in this study. All participants were native English speakers with no documented physical, cognitive, visual, hearing, or speech impairments. None of the participants had prior experience using commercial speech recognition products for dictation-oriented activities. Five participants were female and the average age was 19.4 years.

**7.1.2 Apparatus.** Participants interacted with a custom version of an internally developed speech recognition application, TkTalk 3.0, which uses IBM's ViaVoice speech recognition engine (millennium edition). TkTalk 3.0 was a

modified version of the software used in our earlier studies (e.g. Sears et al. [1999]; Feng et al. [2003]). TkTalk 3.0 provided two modes of interaction. In the first, users could dictate new text. In this mode, the only commands were “Go to sleep” and “Proof document.” The speech recognition software could be disabled temporarily using the “Go to sleep” command. To review new dictation (since the document was last proofread), users issued the “Proof document” command.

Once users entered the “Proof document” mode, navigation anchors were highlighted by making the text gray instead of the normal black. The highlighting for navigation anchors was intended to be subtle to avoid attracting too much attention. If any navigation anchors existed, the first anchor was selected. If there were no navigation anchors, the first word dictated since the last time the user proofread the document was selected. A small dialog box, representing the error correction mechanism that would be integrated into a complete version of this system, was displayed next to the selected word. In the current implementation, this dialog box simply listed the available commands.

Users could issue any of the six navigation commands (i.e. next, previous, move right, move left, move up, move down) to change the word that is currently selected. Once a navigation anchor had been used (i.e. selected as a result of one or more navigation commands), the highlighting was removed and it no longer served as a navigation anchor. When a recognition error was selected, the user issued the “correct” command that simply marked that word as being corrected. For words omitted due to a recognition error, the user was supposed to select the word immediately preceding or following where the omitted word should have been and then issue the “insert” command. This highlighted the selected word to indicate that the insertion was acknowledged. The “clear” command could be used to undo erroneous “correct” or “insert” commands. The “close dialog” command allowed the user to discontinue proofreading and continue dictating.

*7.1.3 Tasks and Procedure.* Each participant completed a practice session followed by a single experimental session. In each session, participants composed documents of approximately 400 words, marking any recognition errors using the proofreading mode. Prior to the practice session, participants were guided through the ViaVoice enrollment process. Next, the commands available within TkTalk were demonstrated and participants were allowed to become acquainted with the software for about 10 minutes. Finally, the participants completed their practice task. Users were finished as soon as they dictated the entire document and indicated that they had marked all of the recognition errors. Participants returned the next day to complete the experimental session. No additional training or practice was provided before the experimental session.

Pilot study results suggested that users found it difficult to remember and correct what they had dictated if they waited until the entire 400-word document was complete. Therefore, we instructed participants to delay error correction and proofread the document whenever they finished dictating 20% to 25% of the document or approximately one paragraph. While processing the input speech, the recognition engine depends on both the acoustic model and the grammar to generate the best guess of the words that users speak. While

acknowledging that delaying error correction may result in some loss of context (due to earlier recognition errors that have not been corrected), this should also allow for more natural speaking patterns (by reducing the number of times speech is interrupted). More importantly, our empirical results confirm that this tradeoff appears to result in recognition error rates remaining stable at 17%. Of course, the specific approach we employed may not be the most effective solution; questions remain as to how much text should be dictated before proofreading the document and correcting the errors. Additional studies will be required to address these questions.

## 7.2 Results

Similar to our earlier studies, we found that approximately 17% of all words dictated were recognized incorrectly. In this section, we discuss the average number of navigation commands required to reach each target, the composition of the commands utilized, and failure rates for these commands. Interestingly, while our participants found and marked 85% of the recognition errors that existed, they also marked several words that were not recognition errors. While some of the missed recognition errors should have been marked for correction, others represented situations where the resulting text was actually correct. Two examples include

User: "The. . . the cat was black"	System: "The cat was black"
User: "The cat is. . . was black"	System: "The cat was black"

In both situations, recognition errors occurred but the resulting text matched what the user wanted. To ensure objectivity, our analysis required participants to mark both of these errors. However, it is important to note that finding this type of error is difficult since the resulting text looks, and is, correct. Similarly, some of the non-errors that users marked represented situations where they changed their mind. The system recognized what they said, but the user decided that different words would have been better. These initial observations confirm that it is critical for future studies to support the complete error detection, navigation, and correction processes.

*7.2.1 Number of Navigation Commands Required.* A total of 1614 successful navigation commands were issued as users navigated to and marked a total of 716 words as needing to be corrected. On average, participants issued 2.25 navigation commands to reach each of the targets. This is very close to the results of the simulation reported above where 2.18 commands were required.

*7.2.2 Command Composition.* As illustrated in Table V, while "next" accounted for more than half of the navigation commands issued, it was used less frequently than predicted by the simulation. The "move down" command, which was not included in the simulation, accounted for 10% of all navigation commands. This suggests that, with the current screen size and font size, it is not uncommon for one full line of text with no recognition errors. Given the recognition error rates observed (17%) and our earlier analysis of how recognition errors are clustered, we would expect to find one recognition error sequence

Table V. Command Composition and Failure Rates of the Empirical Study Result

Command	% of Commands	Failure Rate
Next	52.9	0.8
Move right	20.4	8.6
Move left	13.5	7.2
Move down	10.0	7.4
Move up	0.6	30.8*
Previous	2.5	4.7

\*This is not considered a representative value given how infrequently “Move Up” was used.

every 11 words. The current screen configuration results in approximately eight words per line, suggesting that it is reasonable to expect the “move down” command to be used. These results also suggest that the “move down” command should be integrated into future simulations. “move up” and “previous” are used less frequently and are therefore, less important for the simulation.

*7.2.3 Navigation Command Failure Rates and Consequences.* Table V also includes failure rates for each of the six navigation commands used in this study. For comparison purposes, an earlier study with enhanced navigation commands still found failure rates of over 20% for target-based navigation (e.g. select Friday) and approximately 5% for simple direction-based navigation (e.g. move up) when users created similar sized documents [Feng et al. 2003]. Using the new technique, on average only 3.23% of the commands issued failed, but failure rates varied between commands. As expected, “next” resulted in very few failures. “previous” also had a low failure rate, as it was unlikely to be confused with any of the other commands. “move up” command was used infrequently. Because of the limited number of times this command was issued, and the substantially lower failure rates observed for all of the other commands, the 30% failure rate reported in Table V is not considered representative of what would occur during normal use.

Our earlier studies also highlighted the importance of the consequences users experience when commands fail [Feng et al. 2003]. For example, failed target-based commands moved the cursor to the wrong location approximately 35% of the time and actually changed the content of the document almost 37% of the time. Failed direction-based navigation commands moved the cursor to the wrong location only 12% of the time, but changed the content of the document almost 75% of the time. In contrast, when the navigation commands used in the current study failed, they moved the cursor to the wrong location only 8% of the time and only changed the content of the document approximately 1% of the time. Over 90% of the failed navigation commands were simply ignored by the system, suggesting that the consequences are less severe in the current study.

The differences observed in Table V provide further support for using command composition as one of the factors when selecting the technique to be used when defining navigation anchors. For example, it could be argued that CS/–2,

Fixed/6, and DCS/2 all require about the same number of commands, and as a result, users would not notice any difference in performance. However, when the distribution of commands is considered in conjunction with the failure rates listed above, we can begin to differentiate these solutions. Given the predictions in Tables III and IV, as well as the results in Table V, we can predict average command failure rates of 4.6% for CS/-2 and 4.9% for Fixed/6. In contrast, the predicted failure rate for DCS/2 is only 3.0%.

*7.2.4 Time Spent Dictating and Navigating.* Our current goal is to confirm the potential of this new speech-based navigation mechanism. Assuming the potential of the technique is confirmed, the technique would be refined such that we have additional confidence in the specific set of navigation anchors being presented, and additional empirical studies would be conducted. The average number of commands required, the composition and failure rates of those commands, and the satisfaction results described below are all encouraging. One additional set of measures, time spent dictating and on navigation activities, may also provide useful insights.

In an earlier study, we had users complete the same composition tasks using improved versions of traditional navigation commands [Feng et al. 2003]. Note that, in this earlier study, users completed the entire correction process. As a result, direct comparisons must be viewed with caution. For comparison purposes, we extracted the time users spent dictating as well as the time users spent on navigation activities from both our earlier study and the current study. In the earlier study, users spent an average of 28.6 minutes on dictation and navigation to create 400 word documents. More precisely, they spent 19.3 minutes dictating text and 9.3 minutes on navigation. In contrast, participants in the current study completed the same task, spending only 19.6 minutes on dictation and navigation combined with 12.0 minutes of dictation and 7.6 minutes of navigation. These results are encouraging for two reasons. First, encouraging users to dictate larger quantities of text before correcting errors appears to significantly reduce the total time users spent composing text. Second, there was an 18% reduction in the amount of time spent on navigation.

*7.2.5 User Perceptions.* Subjective measures of participant attitudes toward the new proofreading-based solution were collected at the end of the experimental trial. Participants were asked

- how easy it was to complete the task,
- how easy it was to complete the required navigation activities,
- how satisfied they were with the time required to complete the task,
- how easy this solution was to use compared to their normal data entry solution, and
- how fast this solution was compared to their normal data entry solution.

Responses were provided using a Likert-style scale ranging from 1 (most positive response) to 5 (least positive response). All participants defined their “normal method” as being the keyboard and mouse. Participant responses were surprisingly positive, especially when compared with earlier results where

Table VI. Average Number of Commands and Command Composition of the Empirical Study Result and Simulation Results

Command	Empirical Results	Revised Simulation 100% Optimal Solution (Test Data)	Revised Simulation 100% Optimal Solution (Test Data)	Revised Simulation 75% Optimal (Empirical Results)	Revised Simulation Without any Anchors
Next	53	55	50	56	0
Move right	20	26	28	22	70
Move left	14	7	13	12	15
Move down	10	11	8	8	15
Move up	1	0	0	0	0
Previous	3	2	2	2	0
Average # commands	2.25	1.92	2.24	2.34	2.92

traditional computer users felt that speech recognition was much slower and more difficult to use than a keyboard and mouse when completing these same tasks [Sears et al. 2001]. Overall, participants felt that it was easy to complete this task (mean = 1.92, stdev = 0.79) and that the navigation activities were easy to complete (mean = 2.00, stdev = 0.85). They also rated the total time required to complete this task as acceptable (mean = 1.75, stdev = 0.62). Comparisons with their “normal” data entry technique (all participants identified the keyboard and mouse as their normal solution) were also encouraging when compared with earlier results. Our speech-based solution was rated as easy to use (mean = 2.00, stdev = 1.13) and similar in speed (mean = 2.67, stdev = 1.30) as compared to a keyboard and mouse.

## 8. SIMULATION REVISITED

Since the empirical study result demonstrated that participants did use the “move down” command frequently, our initial simulation design, which did not include the “move down” command, must be refined. Therefore, we revised our original simulation to integrate the use of the “move down” command. We assumed the same screen size and font size as was used in the experiment. Whenever the cursor is not on the same line as the next recognition error, we consider a fourth navigation solution, which includes the “move down” command. With this approach, “move down” is used to reach the line that the error is on, and then “next”, “previous”, “move right”, and “move left” are used to navigate to the error in the most efficient way possible.

We ran the revised simulation on the original data pool of more than 67,000 words. As illustrated in Table VI, the predicted command composition is substantially closer to what was observed in the empirical study. However, the predicted number of commands required is 15% lower than the empirical result. We believe the reason for this discrepancy is that the simulation always adopts the optimal approach (i.e. minimizes the number of commands), but users are unlikely to choose the optimal approach every time. To test this hypothesis, we adjusted the simulation to randomly select a subset of the recognition errors for which the second best navigation solution is used. Using the optimal

solution for 75% of the errors, and the second best solution for the remaining 25% of the errors, results in effective predictions for both the average number of commands required and the composition of those commands (see Table VI).

Based on these results, we applied the revised simulation—selecting the optimal solution 75% of the time—to the results of our empirical study. As illustrated in Table VI, the prediction for the average number of commands required is within 4% of the observed result. Further, the predicted command composition is similar to the observed results. These results suggest that the revised simulation, tuned to select the optimal solution 75% of the time, may prove effective when modeling user interactions with this navigation solution.

To further validate the efficacy of navigation anchors, we also used the simulation to predict the outcomes when no anchors are defined. As shown in Table VI, users need approximately 2.9 commands to reach each error when using only the four “move” commands. This alone, suggests that navigation anchors should reduce the number of commands required for navigation. Further, in the absence of navigation anchors, only “move” commands can be used, which is likely to result in increased recognition errors.

To summarize, the empirical evaluation confirmed that confidence scores could be used to facilitate navigation. However, it does not suggest that the threshold or technique we used to define navigation anchors is the best alternative. A set of techniques or thresholds could define navigation anchors that will prove effective. It is also possible that criteria beyond the two we considered (i.e. number of commands and the composition of those commands) could prove useful when selecting an approach for defining navigation anchors. Some criteria that may prove useful include the number of words highlighted, the distribution of the words highlighted, and the number of recognition errors highlighted. Determining the most appropriate method for defining navigation anchors will require additional empirical studies and is suggested as a promising direction for future research.

## 9. CONCLUSIONS

In the context of dictation-oriented activities, existing speech-based error correction techniques tend to be error prone and time consuming. While numerous researchers have expressed interest in using confidence scores to support the error correction process, even the most recent attempt to do so failed to show the desired benefits [Suhm et al. 2001]. Our data suggest, as did Suhm et al. [2001], that confidence scores are not likely to be effective if the focus is on error detection. More importantly, users appear to find the error detection process acceptable [Sears et al. 1999], suggesting that error specification and correction activities may be a more appropriate focus. The current article focused explicitly on error specification, leaving error correction as the focus of future studies.

Our earlier results confirm that the challenge is to develop efficient navigation techniques that use short, fixed, commands. Navigation anchors were introduced as a means of providing increased efficiency without introducing significantly higher failure rates. Various methods could be used to define navigation

anchors. Our simulation results indicated that the DCS technique is promising when using a threshold of 2. The subsequent empirical study confirmed that users could specify erroneous words efficiently, the required navigation commands resulted in low failure rates, and the consequences of failed navigation commands were minimal. Users generally felt that the technique is easy to use and efficient. A preliminary comparison shows that the new technique outperforms the existing navigation approaches in terms of command failure rate, failure consequences, efficiency, and user satisfaction. A revised simulation further validates the new technique, providing an accurate tool for predicting users' command usage activities.

This initial evaluation highlights the potential of our new anchor-based solution for speech-based navigation as well as the potential of using confidence scores when defining navigation anchors. However, these results do not necessarily confirm that the anchor definition technique/threshold combination explored is the best solution. We do not believe it is likely that there will be a single optimal method of defining navigation anchors. Instead, we believe it is likely that there will be a collection of approaches that will allow for effective speech-based navigation in the context of dictation-oriented applications.

Results from this study also highlighted several directions for future research. First, future studies should support the entire error correction process. This is motivated by two observations: (1) some recognition errors should not be corrected and (2) some words that are correct will need to be changed. It is also motivated by the possibility that difficulties users experience during correction activities may influence their navigation activities. Controlled empirical studies, comparing the new technique with existing solutions, such as McNair's navigation technique, will provide useful information into the efficacy of this new technique.

The results reported above highlight a practical, implemented, example of how confidence scores can be used to support the navigation process in the event that a common desktop-sized display and traditional pointing devices are either unavailable or inappropriate. Perhaps even more important is the initial validation of our new navigation anchor-based approach to speech-based navigation. While interesting, we do not believe these results are definitive or final. Additional studies are required, not only to address the issues highlighted above, but to more thoroughly evaluate the efficacy of various navigation anchor definition technique/threshold combinations. While we focused on the number of commands required and the composition of those commands, other factors may also be important. Marking words as navigation anchors may distract users, suggesting that using fewer anchors could prove more effective. Fewer anchors could also result in users issuing more commands, but this may be an acceptable tradeoff if anchors are sufficiently distracting. It is possible that having some anchors correspond to recognition errors and others correspond to words that were correct could also lead to confusion. Therefore, it is important that additional studies provide insight into how navigation anchors facilitate, and hinder, the process of detecting recognition errors. Further, we completely separated dictation from editing. As a result, users were not allowed to do any inline

correction (e.g. using “scratch that” to delete the last word if it was incorrect) as they dictated. A hybrid system that provides users with limited inline editing capabilities along with more comprehensive proofreading capabilities may prove interesting. Finally, our navigation anchor definition technique/threshold was defined prior to the study and did not change. It would be interesting to develop a technique that adapts as users proofread documents, with the goal of defining more effective navigation anchors. While many interesting issues remain to be studied, the results reported above clearly demonstrate the potential of navigation anchors as well as the potential of using confidence score data to guide the speech-based navigation process.

#### ACKNOWLEDGMENTS

We would like to thank the anonymous reviewers and the editor for thoughtful feedback, which allowed us to clarify a number of important issues.

#### REFERENCES

- BOUWMAN, G., STURM, J., AND BOVES, L. 1999. Incorporating confidence measures in the dutch train timetable information system developed in the ARISE project. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*, Vol. 1, 493–496, Phoenix.
- BOUWMAN, G., BOVES, L., AND KOOLWAALJ, J. 2000. Weighting phone confidence measures for automatic speech recognition. In *Proceedings of the COST249 Workshop on Voice Operated Telecom Services*. 59–62.
- CHASE, L. 1997a. Error-responsive feedback mechanisms for speech recognizers. *Ph.D. Thesis*. School of Computer Science, Carnegie Mellon University, Pittsburgh, PA.
- CHASE, L. 1997b. Word and acoustic confidence annotation for large vocabulary speech recognition. *Eurospeech-97*. 815–818.
- CHRISTIAN, K., KULES, B., SHNEIDERMAN, B., AND YOUSSEF, A. 2000. A comparison of voice controlled and mouse controlled web browsing. In *Proceedings of Assets 2000*. 72–79.
- DANIS, C., COMERFORD, L., JANKE, E., DAVIES, K., DEVRIES, J., AND BERTRAND, A. 1994. Storywriter: A Speech Oriented Editor. In *CHI 94 Conference Companion*. 277–278.
- DE MAURO, C., GORI, M., MAGGINI, M., AND MARTINELLI, E. 2001. Easy access to graphical interfaces by voice mouse. demauro@dii.unisi.it.
- FENG, J., SEARS, A., AND KARAT, C.-M. 2003. A longitudinal investigation of hands-free speech-based navigation during dictation. UMBC Tech. Rep. available from the authors.
- GILLICK, L., ITO, Y., AND YOUNG, J. 1997. A probabilistic approach to confidence estimation and evaluation. In *Proceedings of International Conference on Acoustics, Speech, and Signal Processing*. 879–882.
- GUNAWARDANA, A., HON, H., AND JIANG, L. 1998. Word-based acoustic confidence measures for large-vocabulary speech recognition. In *Proceedings of ICSLP-98, Sydney, Australia*, Vol. 3, 791–794.
- Hazen, T. J. and Bazzi, I. 2001. A comparison and combination of methods for OOV word detection and word confidence scoring. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 397–400.
- HAZEN, T. J., POLIFRONI, J., AND SENEFF, S. 2002. Recognition confidence scoring for use in speech understanding systems. *Computer Speech and Language* 16, 1, 49–67.
- KARAT, C.-M., HALVERSON, C., KARAT, J., AND HORN, D. 1999. Patterns of entry and correction in large vocabulary continuous speech recognition systems. In *Proceedings of CHI 99*. 568–575.
- KARAT, C.-M., VERGO, J., AND NAHAMOO, D. 2003. Conversational interface technologies. In J. Jacko and A. Sears, Eds. *The Human-Computer Interaction Handbook*. LEA: NJ. 169–186.
- KEMP, T. AND SCHAAF, T. 1997. Estimating confidence using word lattices. In *Eurospeech, 1997*, Vol. 1, 371–373.

- LITMAN, D., WALKER, M., AND KEARNS, M. S. 1999. Automatic detection of poor speech recognition at the dialogue level. In *Proceedings of the 37th Annual Meeting for Computational Linguistics*. 309–316.
- MAISON, B. AND GOPINATH, R. A. 2001. Robust confidence annotation and rejection for continuous speech recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 389–392.
- MANARIS, B. AND HARKREADER, A. 1998. SUITEKeys: A speech understanding interface for the motor-control challenged. In *Proceedings of the 3rd International ACM SIGCAPH Conference on Assistive Technologies (ASSETS'98)*. 108–115.
- MCNAIR, A. AND WAIBEL, A. 1994. Improving recognizer acceptance through robust, natural speech repair. In *Proceedings of the International Conference on Spoken Language Processing*. 1299–1302.
- MOU, X. AND ZUE, V. 2000. The use of dynamic reliability scoring in speech recognition. In *Proceedings of the 6th International Conference on Spoken Language Processing*.
- OVIATT, S. L. 1997. Multimodal interactive maps: Designing for human performance. *Human-Computer Interaction* 12, 93–129.
- OVIATT, S. L. 2000. Taming speech recognition errors within a multimodal interface. *Comm. ACM* 43, 9, 45–51.
- OVIATT, S. L. 2003. Multimodal interfaces. In J. A. Jacko and A. Sears, Eds. *The Human-Computer Interaction Handbook*. Mahwah, NJ: Lawrence Erlbaum Assoc. 286–304.
- OVIATT, S. L., COHEN, P. R., WU, L., VERGO, J., DUNCAN, L., SUHM, B., BERS, J., HOLZMAN, T., WINOGRAD, T., LANDAY, J., LARSON, J., AND FERRO, D. 2000. Designing the user interface for multimodal speech and gesture applications: State-of-the-art systems and research directions. *Human Computer Interaction* 15, 4, 263–322.
- SEARS, A., FENG, J., OSEITUTU, K., AND KARAT, C.-M. 2003. Hands-free speech-based navigation during dictation: Difficulties, consequences, and solutions. *Human Computer Interaction* 18, 3, 229–257.
- SEARS, A., KARAT, C.-M., OSEITUTU, K., KARIMULLAH, A., AND FENG, J. 2001. Productivity, satisfaction, and interaction strategies of individual with spinal cord injuries and traditional users interacting with speech recognition software. *Universal Access in the Information Society* 1, 4–15.
- SETLUR, A. R., SUKKAR, R. A., AND JACOB, J. 1996. Correcting recognition errors via discriminative utterance verification. In *Proceedings of ICSLP'96*, Vol. II, 602–605.
- SUHM, B., MYERS, B., AND WAIBEL, A. 2001. Multimodal error correction for speech user interfaces. *ACM Trans. Comp.-Hum. Interact.* 8, 1, 60–98.

Received February 2003; revised October 2003; accepted October 2003 by Brad Myers