INTERACTIVE RECOVERY FROM SPEECH RECOGNITION ERRORS IN SPEECH USER INTERFACES

Bernhard Suhm [1], Brad Myers [2], Alex Waibel [1]

[1] Interactive Systems Laboratories
Carnegie Mellon University and University of Karlsruhe
[2] Human Computer Interaction Institute
Carnegie Mellon University
Email: {bsuhm,bam,ahw}@cs.cmu.edu

ABSTRACT

We present a multimodal approach to interactive recovery from speech recognition errors for the design of speech user interfaces. We propose a framework to compare various error recovery methods, arguing that a rational user will prefer interaction methods which provide an optimal trade off between accuracy, speed and naturalness. We describe a prototypical implementation of multimodal interactive error recovery and present results from a preliminary evaluation in form filling and speech to speech translation tasks.

1. INTRODUCTION

Although intensive research over recent years has boosted the performance of speech recognition technology significantly, the automatic interpretation of speech is inherently unreliable. There are limiting factors beyond the control of the designer of any speech based application, such as variability in speaker and the acoustic environment. Furthermore, even human performance is limited, due to inherent, both acoustic and semantic, ambiguities of natural language. For the design of speech user interface we therefore need methods to gracefully recover from interpretation errors - in addition to further improving baseline performance of spoken language technology.

The field envisions a wide range of applications for spoken language technology, ranging from speech-only applications over noisy channels (e.g. telephone services) to interactive walk-up-and use applications (e.g. speech controlled ATMs). Of course, the approach to error recovery has to take into account this application context. Our research focuses on application settings where multiple input and output modalities are available, at least speech and a touch sensitive display. In particular, we are exploring data entry (form filling), dictation and speech-to-speech translation in a scheduling domain.

Some researchers have acknowledged that for the design of speech user interface, the problem of the inherently unreliable automatic interpretation of human signals, in particular speech, has to be addressed. However few studies have investigated issues relevant to error recovery. Zajicek [1] emphasizes the importance of the user's conceptual model of the interface to the design of speech applications in general, and error recovery in particular. Ainsworth and Pratt [2] designed and compared two error-correcting strategies for a very small vocabulary (14 word) speech recognition system: repetition with elimination (of incorrect prior recognition output from the current vocabulary) and elimination without repetition (i.e. eliminating successively incorrect hypotheses from the N best list of hypotheses). Baber and Hone [3] developed an approach to define requirements for error correction dialogue based on a model of both the task-related dialogue and the underlying speech recognition system. Zoltan-Ford [4] investigated how user queries are influenced by the vocabulary and phrase structure used in the system's messages, suggesting an approach to reduce errors by biasing users towards interaction patterns which are less likely to cause interpretation errors. Oviatt et al. [5] also explored this approach in the context of multimodal interaction. Furthermore, Danis [6] showed that it is possible to increase speech recognition accuracy by training the user, trying to eliminate speaking behaviors which tend to cause recognition errors. Brennan and Hulteen [7] emphasize the importance of context sensitive feedback to facilitate the detection of communication problems.

In prior work at our laboratories [8], a rescoring approach to interactive error recovery based on the information available in N-best lists was implemented and evaluated on the Resource Management task. The drawback of this approach is that error recovery fails when no significantly better alternative can be found in the N-best list. In this paper we describe an approach to error recovery which attempts to leverage multiple repair modalities. Motivated by linguistic research about strategies humans employ to deal with communication problems, we identify three approaches to deal with errors in spoken language systems. We propose a framework for comparing and evaluating interactive error recovery methods. We describe a prototypical implementation of multimodal error recovery, and present results from pilot evaluations on speech to speech translation and form filling tasks.

2. STRATEGIES TO DEAL WITH SPEECH RECOGNITION ERRORS

The three main strategies employed in human to human conversation are avoiding communication problems, initiation of a repair dialogue as soon as a communication problem has been detected, and collaborative work on the repair [10]. Applied to speech user interface design, these strategies correspond to the following three approaches to dealing with interpretation errors,:

1. Reduce the number of interpretation errors by training or channeling the user towards speaking styles and spoken input patterns which the automatic interpretation system can interpret more accurately (through the interface and dialogue design).
2. Facilitate the detection of interpretation errors through context sensitive feedback messages.
3. Recover from interpretation errors by involving the user in interactive error recovery dialogues.

Our research focuses on design of interactive error recovery methods.

3. A FRAMEWORK TO EVALUATE INTERACTIVE ERROR RECOVERY

Given a set of error recovery methods feasible with current technology in a certain application setting, an important question for the designer of a speech user interface is to predict which method users will prefer. Applying the principle of least collaborative effort which governs error recovery in human to human conversations [11] suggests users will prefer methods which minimize the effort necessary to recover from errors.

We argue the effort is determined by the following three dimensions:

- Time required by the user to provide the input, and by the system to process it
- Accuracy of the system in interpreting the input
- “Naturalness” of interaction

“Naturalness of interaction” is meant to capture factors which are dependent on the user and task. For instance, some people have trouble typing, and tasks lend themselves more or less to speech input.

We propose to combine the time required by user to provide input, the time required to interpret the input automatically, and the interpretation accuracy into a single measure which characterizes the overall information output time, given a certain level of accuracy. The latter is of course highly dependant on the task: for instance, in a dictation task, close to 100% will be necessary, whereas in a speech translation task, getting the point across might be sufficient.

First, we estimate the number of attempts necessary to get *alpha%* of the input correct. We make the simplifying assumptions that multiple repair attempts of the same error(s) are stochastically independent, and that the word accuracy WA stays constant over multiple repair attempts. Then, the cumulative word accuracy after N attempts can be estimated using a geometric series:

$$WA \cdot \sum_{i=0}^{N-1} (1-WA)^i = \frac{1-(1-WA)^N}{1-(1-WA)} = 1-(1-WA)^N > alpha$$

Therefore, an input modality which can be interpreted at accuracy WA% will require

$$N > \frac{\log(1-alpha)}{\log(1-WA)}$$

attempts to get *alpha%* correct.

Then, the overall time including repair can be estimated as

$$T = N \cdot T_1$$

where T_1 is the time to input and automatically interpret some input, normalized by the length. Since we are dealing with the input of natural language, and words differ greatly in length, we normalize by the number of letters in the correct transcription of the input.

The proposed measure captures the speed versus accuracy trade-off for a single input modality, ignoring other factors like the “naturalness” of interaction mentioned before, and the overhead time the user spends planning his next action and fiddling with the interface. Of course, these factors will eventually play an important role in the design of the speech user interface. Once we have gained a better understanding of the different modalities and respective error recovery methods, we will address these issues. However we feel they can be captured using established human factor research procedures.

4. A PROTOTYPICAL IMPLEMENTATION OF INTERACTIVE ERROR RECOVERY

4.1 Interactive Repair Dialogue

As already outlined in [8], interactive error recovery can be organized in a dialogue between user and system. It consists of two main phases: first, the system interpretation errors have to be identified, then the user collaborates with the system to correct the error.

The *error detection* can be either initiated by the system, for instance based on some confidence measure, or by the user. Assuming a graphical user interface (GUI) is available, we currently simply require the user to highlight erroneous words (the *reparandum*) in the recognition hypothesis presented to him visually.

For *error correction*, we require the user to help out the system by providing additional input, choosing among various error recovery methods: repair by repeating the reparandum by respeaking, spelling out loud, or handwriting, repair by paraphrasing the reparandum, repair by selecting among N-best alternatives and repair by gestures (e.g. to delete or insert).

This multimodal approach to error recovery takes advantage that different input modalities are orthogonal: where one modality failed, we hope the same input can be reliably recognized in a different modality.

4.2 Repair by Respeaking

We extended the “spoken hypothesis correction method” described in [8] by dropping the strong constraint that the repair utterance has to be one of the N-best alternatives of the reparandum. Instead, we don’t impose any language modeling constraint on the decoding of the repair utterance other than carrying over the trigram context preceding the reparandum. For example:

User speaks: <u>I’m sorry I have a conference there</u>
System recognizes: i’m sorry **off** a conference all day
User highlights “off” and respeaks: <u>I have</u>
System recognizes: i’m sorry **or if** a conference all day

The decoding of the first word of the repair utterance will assume a trigram context “i’m sorry <word>” - instead of the standard beginning of sentence trigram “<s> <s> <word>”. As can be seen, the repair wasn’t successful in the example above.

4.3 Using the Repair Context

From a human factors point of view it can be expected that users won’t tolerate much more than one attempt to recover from errors. Therefore it is crucial to develop highly accurate error recovery methods. In this section we describe how we exploit the repair and interaction context to improve interpretation accuracy.

To improve accuracy of repair by respeaking, we developed a rescoring algorithm which takes advantage not only of the context before the reparandum, but also of the context after the reparandum. Assuming the words in the vicinity of the highlighted reparandum are correct, the lattice generated for the repair utterance is rescored enforcing the two words preceding and the two words following the reparandum as trigram context. In the above example, this technique would enforce the context “i’m sorry {repair utterance} a conference”. That way, in the example above, the correct hypothesis “i have” could be retrieved from the lattice:

System recognizes and displays: i’m sorry i have a conference all day

Thus, by using the reparandum context, repair by respeak may help in cases where no better alternative is available in the N-best of the reparandum.

By excluding words highlighted as erroneous from the vocabulary, we make sure that the same error can’t occur in the interpretation of following repair interactions.

In form filling tasks, knowledge of the current field provides a powerful constraint on both vocabulary and language model. We exploit this contextual knowledge by switching language models and vocabularies dynamically as the user switches between fields.

4.4 Non-Speech Repair Methods

Repair by spelling: To correct an erroneous word, the user spells out loud a sequence of letters. To maximize accuracy, the input is recognized with a specialized connected letter recognizer[13] constrained to the same vocabulary as used in continuous speech decoding.

Repair by handwriting: To repair, the user provides cursive handwriting. As for spelling, a specialized recognizer[14] is used to interpret the handwriting, and the hypothesis is constrained to any word within the speech recognizer’s vocabulary.

Selection among N-best: A list of N-best alternatives for the highlighted words is generated and displayed in a pop-up menu. To avoid cognitive overload, we limit the number of N-best alternatives displayed (typically 6).

Gesture repair: Deletion and insertion of words can be indicated by certain pen gestures.

5. PILOT EVALUATION

Knowledge of the speed versus accuracy trade-off is crucial to be able to design a speech user interface that minimizes the effort necessary to recover from system interpretation errors. We therefore conducted preliminary evaluations to assess the effectiveness of our repair methods, and to explore the speed versus accuracy trade-off given current technology.

5.1 Experimental Design

Based on the most recent version of the JANUS system we have implemented prototypical interactive error recovery interface for a speech to speech translation application in an appointment scheduling domain (Setting 1), and for a form filling task in a military application domain (Setting 2).

In *setting 1*, two subjects are given fictitious calendars and asked to schedule a meeting. The necessary hardware includes two workstations equipped with audio hardware and a touch sensitive display, which are located in different offices. The display is partitioned into three windows: for the sentence hypothesis (the output of the recognizer), for the paraphrase (i.e. the translation of the input back into the input language), and for the most recent “message” from the conversation partner. A typical interaction scenario looks like: The user pushes the “record” button and speaks a “first” utterance (in English). During speech decoding, partial hypotheses are displayed as they become available. Upon completion, the final sentence hypothesis is parsed, and the paraphrase and translation (in German) are generated and displayed. If necessary, the user highlights an erroneous region and corrects by respeaking, spelling, handwriting or N-best choice. Upon completion of repair, i.e. once the user is satisfied with the paraphrase displayed, the “translate” button is pressed and the translation “sent” to the partner, i.e. displayed in the partner’s “message” window. The speech recognizer was trained for human to human conversational speech on our ESST (English Spontaneous Scheduling) database (c.f [12]). The vocabulary size is around 2000.

In *setting 2*, the display consists of the various fields of the form to be filled with data: day and time, sender and addressee (name, grade and phone number), location (names of cities in Bosnia), etc. To get closer to realtime performance, we allowed repair only by spelling, handwriting and N-best, and not by respeaking, thus eliminating the currently quite time consuming step to decide whether spoken input was continuous speech or spelling. A typical interaction scenario consists of the user selecting a field and speaking the desired input, then highlighting errors word by word, and correcting them by spelling, handwriting or N-best selection. For the various small vocabulary (less than 500 words) recognition tasks in this application, we used acoustic models trained on the Wall Street Journal (WSJ) task.

5.2 Repair Accuracies

We conducted preliminary evaluations with 5 subjects in setting 1 and 1 subject in setting 2. All subjects had prior experience with speech recognition technology.

In setting 1, of a total of 57 turns (484 words), 39 needed repair. The decoding of the initial utterances yielded a word accuracy of 78%, Table 1 shows the repair accuracies and on how many interactions it was measured. As can be seen, repair by spelling was the most accurate method, and selection among N-best alternatives the least. The latter reflects the fact that in most cases, no better alternative was found among the system’s top 10 choices. Additionally, the rescoring algorithm described in section 4.3 improves the accuracy of repair by respeak significantly.

TABLE 1: Repair Accuracies (Speech Translation Task)

	Respeak	Respeak + Rescore	Spell	Handwrite	N-best
Repair Accuracy	58%	66%	93%	85%	9%
# Inter.	29	29	15	20	37

Table 2 shows the repair accuracies obtained in setting 2. There were 43 turns (276 words). The decoding of the initial, continuously spoken utterances yielded a word accuracy of 77%. Comparison with Table 1 confirms that spelling and handwriting repair are very accurate. N-best choice appears to be more effective in this setting, probably due to the small vocabularies.

TABLE 2: Repair Accuracies (Form Filling Task)

	Spell	Handwrite	N-best
Repair Accuracy	86%	82%	44%
#Interactions	21	68	9

5.3 Input Time including Repair

In addition to accuracy of interpretation, for setting 2 (form filling task) we measured the time it took the user to provide the speech, spelling or handwriting input, and the time the system took to initially interpret it, normalized by the number of letters in the input, as shown in Table 3. Choosing among N-best lists took on the average 0.75 seconds per letter.

TABLE 3: Input and Interpretation Time [sec/letter]

	Speech	Spell	Handwrite
Input Time	0.2	0.8	0.6
Interpretation Time	0.3	0.6	1.1

To derive trends how effective different repair methods will be considering both accuracy and speed, we applied the method of combining input and interpretation time with accuracy described in section 3 for the data obtained on the form filling task. Given the initial, continuous speech decoding was 77% accurate, we estimated the number of attempts necessary to get 99% correct using spelling, handwriting and N-best repair (row 1 in Table 4), and the total time necessary including repair, in seconds per letter (row 2 in Table 4).

TABLE 4: Total Time including Repair [sec/letter]

	Spell	Handwrite	N-best
Repair Attempts	1.6	1.8	5.4
Total Input Time	1.3	2.3	4.4

As can be seen, given the constraints of current technology, repair by spelling seems to be faster than handwriting and N-best.

6. CONCLUSIONS

Our pilot evaluation suggests that given current technology, repair by spelling and handwriting will be very effective. We have shown that using the context from the repair dialogue, e.g. in rescoring techniques, can substantially improve the accuracy of repair.

Although results are still preliminary, they show that our multimodal approach to interactive error recovery is very promising. Future work will focus on achieving realtime performance, generalizing the repair algorithms to get rid of some simplifying assumptions we currently use, and exploring other repair methods. We will reported updated results at the conference.

7. ACKNOWLEDGEMENTS

This research was sponsored in part by the Department of the Navy, Office of Naval Research.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Navy or the U.S. Government.

The authors would like to thank all members of the Interactive System Laboratories, especially Herman Hild, Stefan Manke and Monika Woszczyna for providing the newest versions of the various recognizers used (continuous speech, connected letters, handwriting).

8. REFERENCES

[1] Zajicek, M., and Hewitt, J. "An investigation into the use of error recovery dialogues in a user interface management system for speech recognition", *Proceedings of the Conference on Human Factors in Computing Systems*: 755-760, 1990

[2] Ainsworth, W.A., and Pratt, S.R. "Feedback strategies for error correction in speech recognition systems", *International Journal of Man-Machine Studies* 36: 833-842, 1992

[3] Baber. C., and Hone, K.S. "Modeling error recovery and repair in automatic speech recognition", *International Journal of Man-Machine Studies* 39: 496-515, 1993

[4] Zoltan-Ford, E. "How to get people to say and type what computers can understand", *International Journal of Man-Machine Studies* 34: 527-547, 1991

[5] Oviatt, S.L., Cohen, P.R., and Wang, M. "Toward interface design for human language technology: Modality and structure as determinants of linguistic complexity", *Speech Communication* 15: 283-300, 1995

[6] Danis, C.M. "Developing Successful Speakers for and Automatic Speech Recognition System", *Proceedings of the Human Factors Scociety 33rd Annual Meeting*, 1989

[7] Brennan, S.E., and Hulteen, E.A. "Interaction and feedback in a spoken language system: a theoretical framework", *Knowledge Based Systems* 8: 143-151, 1995

[8] McNair, A.E., and Waibel, A. "Improving Recognizer Acceptance through Robust, Natural Speech Repair", Proceedings of the International Conference on Spoken Language Processing - ICSLP III: 1299-1302, 1994

[9] Schegloff, E.A., Jefferson, G. and Sacks, H. "The preference for self--correction in the organization of repair in conversation", *Language* 53: 361-382, 1977.

[10] Clark, H.H., and Schaefer, E.F. "Contributing to Discourse", *Cognitive Science* 13:259-294, 1989

[11] Clark, H.H., and Wilkes-Gibbs, D. "Referring as collaborative process", *Cognition* 22: 1-39, 1986

[12] Waibel, A., Finke, M., Gates, D., Gavalda, M., Kemp, T., Lavie, A., Levin, L., Maier, M., Mayfield, L., McNair, A., Rogina, I., Shima, K., Sloboda, T., Woszczyna, M., Zeppenfeld, T., Zhan, P. "JANUS-II - Advances in Spontaneous Speech Recognition, *International Conference on Speech, Acoustics and Signal Processing ICASSP*, 1996

[13] Hild, H., and Waibel, A. "Speaker-Independent Connected Letter Recognition with a Multi-State Time Delay Neural Network", EUROSPEECH II: 1481-1484, 1993

[14] Manke, S., Finke, M., and Waibel, A. "NPen++: A Writer Independent, Large Vocabulary On-Line Cursive Hand writing Recognition System", Proc. Int. Conf. on Document Analysis and Recognition, Montreal, 1995