User Responses to Speech Recognition Errors: Consistency of Behaviour Across Domains

Stephen Choularton and Robert Dale

Centre for Language Technology
Macquarie University
{stephenc|rdale}@ics.mq.edu.au

Abstract

The problems caused by imperfect speech recognition in spoken dialogue systems are well known: they confound the ability of the system to manage the dialogue, and can lead to both user frustration and task failure. Speech recognition errors are likely to persist for the foreseeable future, and so the development and adoption of a well-founded approach to the handling of error situations may be an important component in achieving general public acceptability for systems of this kind. In this paper, we compare two studies of user behaviour in response to speech recognition errors in quite different dialog applications; the analysis supports the view that user behaviour during error conditions contains a large component that is independent of the domain of the dialogue. The prospect of a consistent response to errors across a wide range of domains enhances the prospects for a general theory of error recognition and repair.

1 Introduction

The problems caused by imperfect speech recognition in spoken language dialogue systems are well known; they both confound the ability of the system to understand the dialogue, and lead to user frustration, at worst resulting in task failure. At a recent industry conference,[1] Fausto Marasco, CEO of Premier Technologies, indicated that his company's market research showed that half of all complaints about spoken language dialogue systems concerned recognition errors. Less anecdotally, the published literature suggests that word error rates of between 10% and 40% are not uncommon [Greenberg and Chang, 2000; Bennet and Rudnicky, 2002].

In the commercially deployed systems available today, this problem is approached by having the recognizer generate a confidence measure for each recognition hypothesis. This estimates the likelihood that the hypothesis accurately reflects what was said. Hypotheses whose confidence measure is above some threshold will be accepted as correct; those below some lower threshold will be rejected as likely to be incorrect, resulting in a reprompt; and those in the region between these two thresholds will be considered questionable, resulting in a request that the user confirm the hypothesis. The setting of these thresholds is a fine balancing act: a cautious approach of setting the 'accept as correct' threshold very high will result in an increased number of unnecessary confirmations, lengthening the call and causing frustration for the user; on the other hand, setting this threshold too low will result in the mistaken acceptance of a higher proportion of incorrect hypotheses.

Generally speaking, confidence-estimation algorithms work well: Figure 1 shows the incidence of errors at different levels of confidence taken from an analysis of our Pizza Corpus, discussed further below. Statistics like these are used in tuning system parameters: here, the data would provide support for setting the 'accept as correct' threshold in the 70–80% range. There will still be cases, of course, when the system misrecognizes even at these high confidence levels.

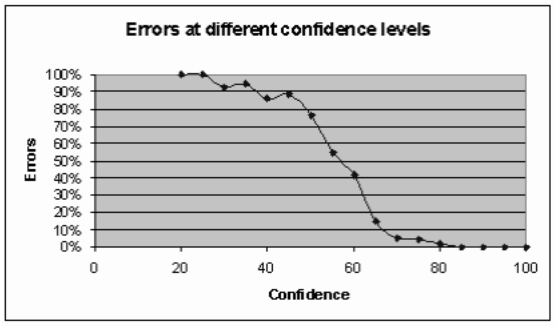

Figure 1: Error rates at different levels of confidence.

Unless some new paradigm for speech recognition becomes available, users of spoken language dialog systems will therefore continue to be faced with occasions where the system they are interacting with either incorrectly thinks it may have misheard and thus requests some confirmation, or, indeed, is quite unaware that it has misheard, and continues the dialogue assuming some incorrect piece of information or instruction that the mishearing has introduced.

If a recognition error is introduced into the dialog, it is obviously preferable if the dialog system is able to determine that this has happened as soon as possible, so that steps can be taken to get the dialog back on track with the minimum of disruption. Consequently, the ways in which people become aware that they have been misrecognized, and how they react to this, are important in determining the best strategies to adopt to repair and recover from these errors.

The aim of this paper is to explore the extent to which user behaviour in response to the awareness of error is

[1] Voice World Australia 2003, Sydney, 25th–27th February 2003.

Proceedings of the 10th Australian International Conference on Speech Science & Technology
Macquarie University, Sydney, December 8 to 10, 2004.

Accepted after full paper review

predictable, and in particular, to explore the question of whether user reaction to error is impacted significantly by the specific application being used, or whether it is generally the same across different applications. Our evidence so far is that the latter is the case, which has positive consequences for the development of a domain-independent theory of error recognition and repair.

In Section 2, we briefly describe relevant previous work in the area. In Section 3, we introduce the corpus analysis which is the subject of the present study, and in Section 4 we present the results of a comparison of the analysis of error-induced behaviour in the two corpora studied. Finally, in Section 5 we draw some conclusions and point to future work.

2 Related Work

There are a number of strands of work that are concerned with understanding a user's response to the awareness of error. One strand is concerned with relatively high-level models of communication in dialog, and requires inferring the particular speech acts being performed and reasoning about their role in the ongoing dialog. SharedPlan Theory [Grosz and Sidner, 1990] attempts to model the sharing of information through dialogue, and Traum [1994] explores the notion of a grounding speech act as a means of ensuring that information has been correctly introduced into a dialogue; and a number of other authors since this earlier work have tried to model language at a higher level than words and grammars in order to catch and repair errors (see, for example, Allen [1995]; McRoy and Hirst [1995]; Danieli [1996]; Perlis and Purang [1996]; Purang [2001]).

There is also work that we might consider as operating at a relatively low level, that of the analysis of the prosody of user utterances. Work on hyperarticulation[2] (see Stifelman [1993]; Oviatt et al. [1998]; Levow [1998, 1999]) concludes that speakers change the way they are talking in principled ways when speech recognition problems arise. There has also been work on the association of prosodic features with sites in dialogues where errors occur; Shriberg and Stolcke [2001] provides a very good summary of the approach and points to other work in the field. The work of Hirschberg, Litman and Swerts is particularly significant here (see, for example, Hirschberg et al. [2000]; Litman et al. [2001]; Hirschberg et al. [2004]).

Our concern, however, is with analyses that fall between these two, and which are more concerned with the particular lexical and syntactic phenomena that are indicative of an error having been recognised. A few existing studies have approached this aspect of the problem.

Bousquet-Vernhettes et al. [2003] report work-in-progress on corrective sub-dialogues and error handling in a study of a human–human dialogue corpus in the air traffic control domain, consisting of conversations between apprentice controllers and persons playing the role of pilots. They identify a number of properties of the utterances produced by users when they realise that they have been misheard. These utterances contain:

- specific lexico-syntactic phenomena (signals of illocutionary acts of refusal (e.g., *no* and *I don't want*) and of preference (e.g., *I would prefer* and *rather*), and command words (e.g., *cancel* and *start again*);
- phenomena due to spontaneous speech, including hesitations and pauses, incoherent user utterances (owing to anger, irritation, or upset), and no answer at all; and
- *non sequiturs*, such as utterances that are away from the focus of the dialogue.

In addition, these utterances often generate a high number of recognition errors.

The work on human–machine dialogue, although not large, clearly supports the idea of established patterns of behaviour. In two studies nearly ten years apart, Stifelman [1993] and Shin et al. [2002] both noted several typical user responses to error awareness, such as exact repetition, repetition with editing cues, and re-wording, as demonstrated respectively in the following examples:

(1) U: I'd like to fly to New York.
S: OK, Newark, and where are you departing from?
U: I'd like to fly to New York!

(2) U: I'd like to fly to New York.
S: OK, Newark, and where are you departing from?
U: No, I'd like to fly to New York!

(3) U: I'd like to fly to New York.
S: OK, Newark, and where are you departing from?
U: I want to go to New York from Boston.

Stifelman analyzed data from an Air Travel Information System (ATIS) developed by MIT's Spoken Language Systems Group; the corpus contained 45 user sessions, 388 user commands and 103 speech recognition errors. The errors were analyzed and classified as one of substitution, insertion, and deletion, plus a multiple error type for combinations. The user's responses to recognizing the error were classified into the following categories:

- Exact repeat;
- Partial repeat, including possibly an editing expression;
- Reword, which itself is subscategorized into (a) simplify; (b) elaborate; or (c) break into multiple queries.

An example is shown in Table 1.

Not all errors led to repair; users sometimes missed the error, and in some other cases the dialogue simply failed at that point (Stifelman does not provide a breakdown of these other cases). The distribution of repair strategies Stifelman found is shown in Table 2.

Shin et al.'s study, also an analysis of ATIS data, is by far the most elaborate study of error in human–machine dialog carried out to date. In the late 1990s, the DARPA Communicator Program sponsored the development of a number of sophisticated spoken language dialog systems operating in the domain of air travel planning; as part of this program, in 2000 a major corpus of dialogues was collected

[2] Hyperarticulated speech is delivered more slowly, with spaces between words accentuated and pronunciation more pronounced.

Accepted after full paper review

Turn #	Utterance	Comment
1.1 System	What date will you be returning on?	
1.2 User	September twenty nine	
1.3 System	Here are the Continental flights from Denver to Boston on Sunday September twenty	Error type: Deletion
1.4 User	No, I said September twenty nine	Response: Partial repeat with an editing expression

Table 1: An example of Error and Repair [from Stifelman, 1993].

Repair Strategy	Percentage of Occurrences
Exact repeat	7.77%
Partial repeat with an editing expression	1.94%
Partial repeat, no editing expression	4.85%
Reword	50.49%
Error missed or dialogue failed	34.95%
Total	100.00%

Table 2: Stifelman's breakdown of repair strategies.

from nine participating institutions [Walker et al., 2001].[3] Shin et al. [2002] designed a tagging scheme for marking up speech recognition errors and used this to analyze how participants in this collection became aware of errors and responded to them. This scheme, and the results of Shin's analysis, are described in more detail in the next section.

Krahmer et al. [2001] looked specifically at lexical clues in users responses to implicit and explicit verification questions along two dimensions. They used a corpus of 120 dialogues from a Dutch train timetable application, and tagged system prompts for the nature of the verification question, explicit or implicit; the number of information items being verified; the presence or absence of any default assumption such as the user wishing to travel today; and the presence of recognition errors and whether the error has persisted from a previous turn. User utterances were tagged for number of words; whether a user response was detected; whether the word order was marked (as in *To Amsterdam I want to travel* and *Where I want to go to is Amsterdam*, as opposed to *I want to travel to Amsterdam*); the presence of confirmation markers (such as *yes*, *yup*, and *right*) and disconfirmation markers (such as *no*, *nope*, and *wrong*); and the number of repeated and/or corrected information items. They found that various combinations of these cues were predictive of error.

3 Corpus Analysis

The studies just described demonstrate that there are a range of typical user behaviours found in those situations where a user realises that the application they are conversing with has misheard them. However, the published studies of human-machine dialogue have generally been of laboratory-based research systems, rather than commercially-deployed systems; only Krahmer et al. [2001] studied a fielded system. This has potentially important consequences; in particular, live users of commercial systems have a vested interest in achieving the goal they had in mind when initiating a dialogue with the application. This is not necessarily true for research systems, where there is generally no real outcome, and so the user may be inclined to acquiesce in the face of repeated misunderstandings by the system. If a real user wants to fly to Boston rather than Austin, for example, then, faced with an application that mistakenly recognizes *Boston* as *Austin*, they will be inclined to either persist until they succeed in communicating their intention to the system, or they will hang up in frustration. A subject in an experiment using a research prototype does not have this investment in the achievement of a particular outcome, and so is more likely to give in to the system, accepting the system's misunderstanding as correct just to complete the transaction.

It is also notable that, whereas the recognizers in laboratory systems tend to be built using n-gram language models, commercially deployed systems almost exclusively use hand-crafted and hand-tuned grammars. This has an impact on the overall sophistication of the system and its 'user feel'; and we might expect it also to have an impact on the kinds of errors that are produced by the recognizer, and thus possibly on how the user reacts to them.

Our interest, then, is in seeing how the kinds of analysis carried out so far on laboratory systems might carry across to real commercially-deployed systems. To this end, we have obtained a substantial corpus of user data for a pilot deployment of an application designed to take pizza orders. This corpus is described in the next section; then, we describe Shin's tagging scheme, and our application of this scheme to a subset of the Pizza Corpus, presenting the comparative results for the Pizza Corpus and the DARPA Communicator data analysed by Shin.

3.1 The Pizza Corpus

The Pizza Corpus is a large corpus arising from a pilot deployment of a real pizza ordering system, provided to us for research purposes by an Australian provider of speech recognition solutions. The system employed Nuance technology for speech recognition. The corpus consists of 2486 dialogues containing 32728 utterances. Using a strict measure of errors (that is, defining a speech recognition error as any recognition that differs from the human transcription of the original utterance), 19.6% of the utterances contain errors.

[3] This corpus, together with a subsequent further DARPA data collection, will be published by the Linguistic Data Consortium in the near future.

Feature	Shin	Pizza
Dialogues	141	176
Turns	2528	1942
Words Per Turn	2.64	1.68
Error Segments	235	219
Back on Track	78%	58%

Table 3: Corpus Subset Comparison.

3.2 Shin's Tagging Scheme

Shin et al. [2002] devised a tagging scheme consisting of 19 tags with which to monitor three dimensions of dialogues: system behaviour, user behaviour, and task status. The system behaviour tags characterize the clues the system gives the user (intentionally or otherwise) that a recognition error may have occurred. There are six of these: *explicit confirmation*, *implicit confirmation*, *reject*, *aid*, *non sequitur*, and *system repeat*. The user behaviour tags characterize the nature of the user's reaction to these clues. There are ten of these: *repeat*, *rephrase*, *contradiction*, *frustration*, *change request*, *start over*, *ask*, *scratch*, *acquiescence*, and *hang up*. The task status tags characterize features of the state of the dialogue. There are three of these: *error*, *back-on-track*, and *success*.[4]

Shin et al. present a number of results of applying this analysis to a fragment of the DARPA Communicator Corpus, which we will review below. We took the same tags and applied the same kind of analysis to a sample of the Pizza Corpus roughly comparable in size to the corpus subset analysed by Shin; the comparative figures are shown in Table 3. Note that many more of Shin et al.'s got 'back on track'.[5] This may be accounted for by the fact that the commercial system allowed users to easily switch to a human agent in the face of problems, and, as has already been mentioned, users of laboratory systems have a greater tendency to acquiesce in the face of errors.

Applying the tags from Shin's scheme to a different corpus is not entirely straightforward, in large part because some of the tags are idiosyncratic consequences of the underlying design of the application analysed: some of Shin's system behaviour tags reflect the strategies designed into the application, and the user behaviour tags reflect responses to these particular system functionalities. Thus, Shin provides tags like *scratch* and *start-over*, which are seen less frequently in deployed systems than user responses like *help* or *operator*; these responses, on the other hand, are not used at all in the DARPA systems.

However, the major categories appear to be broadly applicable. Table 4 shows examples from each of the two corpora of the most common tags used in the tagging scheme.

[4]There is insufficient space here to provide examples of each of these tags: see `http://sail.usc.edu/dialog/model_tags` for examples.

[5]A dialog is considered to be back on track at the point where the human annotator sees evidence of both the user and the system having completely recovered from the error.

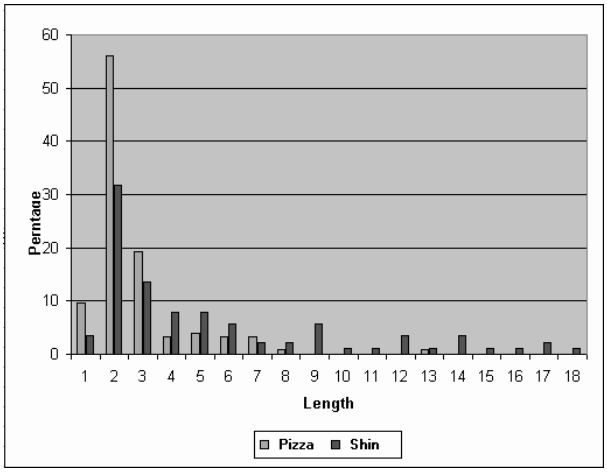

Figure 2: Length of error segments in the two corpora; Shin's figures from Shin et al. [2002].

4 A Comparison

4.1 Length of error segments

The first question we might ask is how long it takes to recover from an error once it has been recognised. The tagging scheme allows the extraction of the length of **error segments**, where an error segment commences with an substantive error and continues until the dialogue is either back on track or ends. Figure 2 shows the results from both Shin et al.'s study and the same analysis performed on the Pizza Corpus.

The similarity of these distributions is marked. Shin et al.'s has a longer tail, but as it did not offer the facility to switch to an operator for completion, it is not surprising users continued to endeavor to get back on track with the system for longer periods. The peak at two turns indicates that, in both cases, errors are generally spotted and recovered from quickly. Examples (4) and (5) are typical examples of this behaviour from the Pizza and Communicator corpora respectively:

(4) S: What would you like for your first pizza?
U: [*no recognition*] [Actual: one pan ham hawaiian]
S: For standard menu pizzas just say for example, 'A large pan super supreme'. For a personalized pizza, ask for help. Now, what would you like for your first pizza?
U: one large pan ham and pineapple
[Actual: one large pan ham and pineapple]
S: OK, one large pan ham and pineapple. Is that right?
U: yes [Actual: yes]

(5) S: Would you like a car in Indianapolis?
U: thanks [Actual: no thanks]
S: I don't understand.
U: thank you [Actual: no thank you]
S: I don't understand
U: no thank you [Actual: no thank you]

Note that in both cases, the fast recovery is simply due to the system providing immediate feedback that there may have been a problem in recognition; obviously a system

Response Type	Communicator Corpus	Pizza Corpus
Rephrase	S: what day do you want to leave denver U: july thirteenth two thousand *july two thousand* S: what day do you want to leave denver U: i want to leave denver july thirteenth	U: one pan ham hawaiian *[recognition rejected]* S: For standard menu pizzas just say for example 'A large pan super supreme'. Now, what would you like for your first pizza? U: one large pan ham and pineapple
Repeat	S: would you like to make any corrections U: no thank you *thank you* S: i don't understand U: no thank you	S: How many pizzas would you like? U: three *[recognition rejected]* S: Please tell me just how many pizzas you would like to order U: three
Contradict	S: i have you going from boston texas to washington d c on september twenty sixth . is that correct? U: no austin texas to phoenix arizona	U: a large house @hes@ [fragment] ham and pine hawaiian *@hes@ large house supreme minus olives* S: ok a large house supreme but no black olives U: incorrect no
Change	S: what time do you want to leave cairo U: ++uh++ sometime about noon *++uh++ how 'bout about don't* S: what time do you want to leave cairo U: two o'clock	S: Was it home delivery or pickup from a store? U: i i want an operator *no i want an operator* If you want your pizza delivered to you say delivery. If you want to pick-up your pizza from a store, say pick-up. U: delivery

Table 4: Examples of the principal response types: S = system, U = user; recognition hypotheses are shown in italics.

which pushed on without giving either explicit or implicit confirmation risks the possibility of only discovering that an error has occurred much later. This emphasises the design decision discussed earlier regarding the setting of appropriate confidence thresholds.

4.2 User behaviour in the face of error

The most striking outcome of the analysis is that user behaviour showed strong similarities across both Shin et al.'s and the Pizza Corpus samples. Figure 3 shows rephrase and repeat as strong favorites in both corpora. The reactions grouped together as 'Other' are sometimes system-specific, like *help* and *scratch* and cover some eight less-used tags.

The lesson from this seems remarkably clear, and in some regards is not particularly surprising, although nonetheless it is useful to find empirical support for it: when a user suspects that the system they are conversing with has made a mistake, the most common response is to either reword the previous utterance, or to repeat it verbatim.

The use of editing expressions is less common than one might imagine. This may be in part due to the fact that the general awareness amongst the public of the nature of speech recognition systems has improved over the years (and certainly since Stifelman's study, which was carried out at a time when there were no commercial speech applications in use). A large proportion of today's users are aware that speech recognition may simply fail to work at times, and so a repetition (in case the failure was due to some other factor such as noise) or rewording (to find an alternative formulation that the recogniser does accept) are the best strategies to pursue. Users quickly discover that complex discussions over the problem with the system are bound to fail.

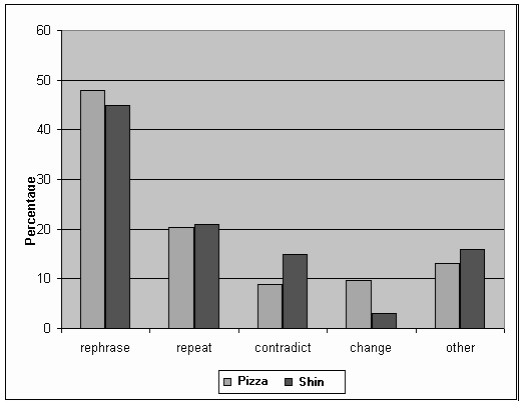

Figure 3: User behaviour in response to errors. Shin's figures from Shin et al. [2002]

5 Conclusions

Speech recognition errors are a significant and common problem for spoken dialogue systems. In this paper, we have looked at two studies of human behaviour when errors are introduced into spoken dialogue systems. We have noted that repair times and user behaviour are remarkably consistent over two very different systems with very different users, supporting the view that these behaviours contain a significant non-domain specific component.

The fact that rewording and repetition account for by far the largest proportion of utterances made after an error is obviously of importance when considering how to handle errors. It is well accepted that the probability of further errors doubles after a first recognition error [Oviatt et al., 1998; Levow, 1998, 1999]. Hyperarticulation plays its part in this, but many subsequent utterances are out-of-grammar, as users elaborate their replies. Of course one could simply suggest careful prompt design that directs

users towards in-grammar utterances, but that reduces the solution to a design issue when it would appear there is growing evidence that there is a significant domain independent component to user behaviour after errors. If this behaviour extends to the forms of language employed, it may be possible to ensure that the language models used to assist recognition are better fitted to the task at these critical points in the dialogue; as a next step we intend to carry out a further study of our corpus to determine if the lexical and syntactic nature of rewording is predictable. Turning to repetition, it would be useful to see if it is possible to determine if an utterance is a repetition of a previous utterance simply by statistical comparison of the features that can be extracted from the sound signal such as formants, pitch and intensity. This knowledge could be used, for example, to stop repeated misrecognitions of the same utterance.

References

J. Allen. Robust Understanding in a Dialogue System. In *34th Meeting of the Association for Computational Linguistics*, 1995.

C. Bennet and A. I. Rudnicky. The Carnegie Mellon Communicator Corpus. In *Proceedings of ICSLP 2002*, pages 341–344, Denver, Colarado, 2002.

C. Bousquet-Vernhettes, R. Privat, and N. Vigouroux. Error handling in spoken dialogue systems: toward corrective dialogue. In *Error Handling in Spoken Langugage Dialogue Systems*, pages 41 – 45. International Speech Communication Association, 2003.

M. Danieli. On the use of expectations for detecting and repairing human-machine miscommunication. *Computational Linguistics*, 13:11, 1996.

S. Greenberg and S. Chang. Linguistic Dissection of Switchboard-Corpus Automatic Speech Recognition. In *NIST Speech Transcription Workshop, College Park, MD*. NIST, May 2000.

B. J. Grosz and C. L. Sidner. *Intentions in Communication*, chapter Plans for Discourse, pages 417 – 444. MIT Press, Cambridge, MA, 1990.

J. Hirschberg, D. Litman, and M. Swerts. Generalizing prosodic prediction of speech recognition errors. In *International Conference on Spoken Language Processing (ICSLP)*, Beijing, China, September 2000. ICSLP.

J. Hirschberg, D. Litman, and M. Swerts. Prosodic and other cues to speech recongition failures. *Speech Communication*, (43):155–175, 2004.

E. Krahmer, M. Swerts, M. Theune, and M. Weegels. Error Detection in Spoken Human-Machine Interaction. *International Journal of Speech Technology*, 4(1):19 – 23, 2001.

G.-A. Levow. Characterizing and Recognizing Spoken Corrections in Human-Computer Dialogue. In *COLING-ACL '98*, Montreal, Canada, 1998. COLING-ACL.

G.-A. Levow. Understanding recognition failures in spoken corrections in human-computer dialogue. In *ESCA Workshop on Dialogue and Prosody*, Eindhoven, Netherlands, 1999. ESCA.

D. J. Litman, J. Hirschberg, and M. Swerts. Predicting User Reactions to System Error. In *Meeting of the Association for Computational Linguistics*, pages 362–369, Toulouse, France, 2001.

S. W. McRoy and G. Hirst. The repair of speech act misunderstanding by abductive inference. *Computation Linguistics*, 21(4):435–478, 1995.

S. Oviatt, M. MacEachern, and G.-A. Levow. Prediciting Hyperarticulate Speech During Human-Computer Error Resolution. *Speech Communication*, 24(2):87 – 110, 1998.

D. Perlis and K. Purang. Conversational Adequacy: Mistakes are the Essence. Technical report, Department of Computer Science, University of Maryland, 1996.

K. Purang. *Systems that detect and repair their own mistakes*. PhD thesis, University of Maryland, 2001.

J. Shin, S. Narayanan, L. Gerber, A. Kazemzadeh, and D. Byrd. Analysis of User Behavior under Error Conditions in Spoken Dialog. In *Proc. ICSLP*, pages 2069 – 2072, Denver, Colorado, 2002.

E. Shriberg and A. Stolcke. Prosody Modeling for Automatic Speech Understanding: An Overview of Recent Research at SRI. In *Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and Understanding*, pages 13 – 16. ISCA, Red Bank, NJ., 2001.

L. J. Stifelman. User Repairs of Speech Recognition Errors: An Intonational Analysis. Technical report, Speech research Group, MIT Media Laboratory, May 1993.

D. R. Traum. *A Computational Theory of Grounding in Natural Laguage Conversation*. PhD thesis, University of Rochester, New York, 1994.

M. Walker, J. Aberdeen, J. Boland, E. Braat, J. Garofolo, L. Hirschman, A. Lee, S. Lee, S. Narayanan, K. Papineni, B. Pellom, J. Polifroni, A. Potamianos, P. Prabhu, A. Rudnicky, G. Sanders, S. Seneff, D. Stollard, and S. Whittaker. DARPA Communicator Dialogue Travel Planning Systems: The June 2000 Data Collection. Aalborg, Denmark, 2001. Eurospeech.

Accepted after full paper review