Available online at www.sciencedirect.com

SCIENCE DIRECT®

Speech Communication 45 (2005) 325–341

SPEECH COMMUNICATION

www.elsevier.com/locate/specom

Exploring human error recovery strategies: Implications for spoken dialogue systems

Gabriel Skantze *

Department for Speech Music and Hearing, KTH, Lindstedtsvägen 24, SE-100 44 Stockholm, Sweden

Received 6 February 2004; received in revised form 10 November 2004; accepted 16 November 2004

Abstract

In this study, an explorative experiment was conducted in which subjects were asked to give route directions to each other in a simulated campus (similar to Map Task). In order to elicit error handling strategies, a speech recogniser was used to corrupt the speech in one direction. This way, data could be collected on how the subjects might recover from speech recognition errors. This method for studying error handling has the advantages that the level of understanding is transparent to the analyser, and the errors that occur are similar to errors in spoken dialogue systems. The results show that when subjects face speech recognition problems, a common strategy is to ask task-related questions that confirm their hypothesis about the situation instead of signalling non-understanding. Compared to other strategies, such as asking for a repetition, this strategy leads to better understanding of subsequent utterances, whereas signalling non-understanding leads to decreased experience of task success.

Keywords: Error handling; Miscommunication; Spoken dialogue systems; Wizard-of-Oz

1. Introduction

One of the greatest challenges when building dialogue systems is to deal with uncertainty. Uncertainty comes not only from the ambiguity of language, but in the case of spoken dialogue, from imperfect speech recognisers, from which the system designer must expect a certain amount of errors. However, as Brown (1995) points out, apparently satisfactory communication may often take place between humans without the listener arriving at a full interpretation of the words used. One explanation for this is the redundancy of language and that information often is repeated by the speakers in order to ensure understanding. Furthermore, when humans speak to each other, there is a collaborative process of recovery from non-understanding or misunderstanding that often

* Tel.: +46 8 790 78 74.
E-mail address: gabriel@speech.kth.se

0167-6393/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.specom.2004.11.005

goes unnoticed (Clark, 1994). The question is how the seemingly smooth handling of miscommunication in human–human dialogue can be transferred to human–computer dialogue. In this paper, an experiment on human–human communication is presented, where the error recovery strategies employed after miscommunication are explored and analysed, and their applications to spoken dialogue systems discussed.

One approach to explore such strategies is to look at problems as they occur in human–human dialogue, and transfer this knowledge to human–computer dialogue. There are two potential problems with this approach. Firstly, human–computer dialogue and human–human dialogue have been shown to have different properties (Fraser and Gilbert, 1991). This is partly due to the user's idea of the system, not being a human, and partly due to the limitations of the system's conversational capabilities. A second, less obvious, problem is that the participants' actual understanding of what is said is not always transparent to the analyser. Just because a speaker does not give any sign of non-understanding does not necessarily mean that every word was understood correctly.

A method that is commonly used to collect data on human–computer interaction before a dialogue system is actually built is the Wizard-of-Oz method, where an operator is simulating parts of the system, most often assuming a perfect speech recogniser. The advantage compared to studying human–human dialogue is that people will act is if they spoke to a computer. This method could be used to collect data on miscommunication, but the problem is that the collected corpus will not contain any data on how the speakers handle typical speech recognition errors (since the recogniser is usually simulated by the operator).

In the experiment presented in this paper, a speech recogniser has been used in human–human dialogue to introduce errors. The problems of miscommunication that these errors give rise to, and the effects they have on the dialogue and the subjects' experience of it, have been analysed. This approach has three advantages. Firstly, the speech recogniser imposes some limitations similar to those of a dialogue system, which makes the dialogue more similar to human–computer dialogue. Secondly, the kind of errors that occur are probably more similar to those that occur in spoken dialogue systems than those that occur in ordinary human–human dialogue or those that may be simulated in Wizard-of-Oz studies. Thirdly, since the operator's understanding is limited to that of the speech recognition result, the level of understanding is more transparent to the analyser.

It is important to remember that a dialogue system, in most cases, is not just a conversational partner, but also a tool that is used by a human to solve a task. Thus, error handling strategies should aim at improving factors that are important from a usability point of view. In common view, these factors are the objective measures of effectiveness and efficiency, and subjective measures of user satisfaction (e.g. Walker et al., 2000; Larsen, 2003). Although this experiment is not a study of human–computer interaction, the metrics that have been used to evaluate the outcome of different strategies are influenced by these factors, in order to make the results more useful in the design of spoken dialogue systems.

2. Background

2.1. Miscommunication and error handling

Miscommunication is a general term that denotes all kinds of problems that may occur in dialogue. A common distinction is made between misunderstanding and non-understanding (e.g. Hirst et al., 1994; Weigard, 1999). *Misunderstanding* means that the listener obtains an interpretation that is not in line with the speaker's intentions. If the addressee fails to obtain any interpretation at all, or obtains more than one interpretation, with no way to choose among them, a *non-understanding* has occurred. One important difference between non-understandings and misunderstandings is that non-understandings are recognized immediately by the addressee, while misunderstandings may not be identified until a later stage in the dialogue. Some misunderstandings might never be detected at all. McRoy (1998) adds *misinterpretation* to this list, which occurs when the speakers have different beliefs about the world. The current study will focus on the first two kinds of miscommunication, primarily on non-understanding.

Error handling is the process of preventing, detecting and recovering from errors. It might seem a bit awkward to talk about "errors" in human–human communication. However, in human–computer dialogue, miscommunication situations, at least when they are caused by the system not understanding or misunderstanding the user, can be regarded as deviations from a desired system performance given the user's actions.

Given a speech recognition result, the system must decide what parts of the utterance should be taken as correct. If there seems to be too many errors, the system should consider it a non-understanding. Thus, non-understanding should not be regarded as something that just happens, but is something the system has to detect (unless there is no plausible interpretation at all). If the system detects that there may be errors in the result, it must decide if it should make an interpretation and thereby risk a misunderstanding or if it should decide upon a non-understanding.

Schegloff (1992) makes a distinction between second turn repair and third turn repair. In Schegloff's account, second turn repair means that the detection and repair is made in the second turn (counted from the turn when the utterance was spoken), whereas third turn repair is detected and initiated in the third turn. Thus, second-turn repair is done after non-understanding and third-turn repair after misunderstanding. It is questionable whether the term "repair" is appropriate when it comes to non-understanding and misunderstanding, since it suggests that the understanding of the previous utterance must be repaired. Instead, the speakers try to improve the understanding of subsequent utterances (which may be repetitions or rephrasing of the problematic utterance), i.e. they try to resume understanding, or to get "back on track", as Shin et al. (2002) puts it. Thus, the term "error recovery" may be more suitable.

The study of error recovery after non-understanding has mainly focussed on how users' repetitions of utterances should be handled better. For example, Ainsworth and Pratt (1992) investigates how the system could eliminate the recognised word from the vocabulary to improve recognition of repetitions. Several studies have also focussed on the problem of hyperarticulation of repetitions (e.g. Oviatt et al., 1996; Levow, 1998; Bell and Gustafson, 1999). Many speech recognisers are not built for hyperarticulation, which makes the understanding of repeated utterances even more difficult. A common assumption seems to be that after non-understanding, the system has no option but to signal non-understanding, and thereby encourage repetition. Balentine et al. (2001) argues that such requests for repetition tend to be very tedious if there are a lot of errors, and that they should be avoided.

2.2. Studying human–computer error handling

In order to design dialogue systems that can handle the varieties of situations that occur in human–computer dialogue—such as miscommunication—data of such interaction needs to be collected. The Wizard-of-Oz method, in which parts of the system are controlled by an operator (the "wizard") (Fraser and Gilbert, 1991), is traditionally used for tasks like this. One problem concerning studies of error handling is that it is hard to get an accurate account of what happens when speech recognition errors occur, since they are often ignored when the experiment is conducted. The optimistic assumption tends to be that these things can be added later on when the rest is solved, or that the problem will disappear automatically as speech recognisers get better. One approach to get data on miscommunication is to simulate errors, for example by randomly substituting words in the input. But, as Fraser and Gilbert (1991) points out, this is an almost impossible task in a Wizard-of-Oz environment. Firstly, the wizard is working under time pressure and it may be hard to make the right substitutions while controlling other components. Secondly, the kind of errors that really do occur are hard to simulate. Just substituting random words may be too simplistic a model. Out-of-vocabulary words will often give rise to unexpected results, as the speech recogniser is trying to fit what has been said into the language model using in-vocabulary words. Another approach considered by Fraser and Gilbert (1991) is using a speech recogniser as a filter between the user and the operator, but they argue that it would be too hard for the operator to read the speech recognition results, and that the poor speech recognition performance would constrain the dialogue too much. Paek (2001) suggests that a speech recogniser could be used in a Wizard-of-Oz setting to establish a gold standard for other components, which are simulated by the operator.

A fundamental assumption behind the Wizard-of-Oz-method is that users are thought to behave differently when talking to a machine compared to talking to a human (Dahlbäck et al., 1993). There are two reasons for this. Firstly, the system has conversational limitations that will constrain the dialogue. Secondly, the user has a model of the partner that will affect the linguistic constructs used. Many studies that compare human–human and human-wizard conditions (see Fraser and Gilbert, 1991 for an overview) do not make this distinction and these variables are not controlled systematically and independently of one another. Thus, they do not tell us which one of them is important for the differences that appear. However, in an experiment conducted by Amalberti et al. (1993), the effect of the user's conceptions about the other speaker was tested independently of the limitations of the system. Two groups of subjects were asked to obtain information about air travel via spoken dialogue with a remote travel agent. One group was told that they were talking to a computer, while the other was told that they were talking to a human operator. In both cases, the voice of the operator was distorted. The amount of distortion was carefully tuned, so that the human group could be told that they were testing communication through a noisy channel, while the other group were told that they were talking to a computer. Thus, the experimental setting was exactly the same for the two groups, apart from their conceptions about the other speaker. The results showed that there were differences in the users' linguistic behaviour. However, the differences were most noticeable initially, and a lot of differences tended to disappear in subsequent sessions. Furthermore, the differences that were found between the groups were mainly related to problem solving, where the users in the human group were more cooperative towards the operator. This suggests that the experience the user has of a system will affect the user's behaviour in future interactions. If users are faced with more cooperative systems, they may start to take advantage of this. In order to make advance in the development of dialogue systems, it could be dangerous to adapt to users' current beliefs of the capabilities of such systems, especially users who have a very limited experience of them.

This leads us to another problem concerning explorative Wizard-of-Oz experiments. Numerous studies show that the behaviour of a dialogue system has great impact on the user's behaviour (e.g. Brennan, 1996). Thus, the way the wizard acts will influence the data that is collected. This can be a problem, since the collected data might be based on a priori assumptions about the users' behaviour and how a system is supposed to react to them, and might not cover other, unanticipated interaction patterns. Using a speech recogniser in a controlled Wizard-of-Oz setting would also make it hard to prescribe how the operator should behave depending on different levels of comprehensibility.

All in all, it seems as if the Wizard-of-Oz method might be difficult to use for studying error handling strategies, even if a speech recogniser is included in the setting, unless the experimenter has a very clear idea of the different errors that will occur and which specific error handling strategies should be tested. In order to deceive the subject, the wizard must work fast and accurately. This does not only require a good design of the experimental setting and operator environment, but also much training of the operator. Since the method is very costly to perform, it may be unfeasible to use it for conducting more explorative experiments on such strategies.

3. Method

3.1. Using speech recognition for exploring human error handling strategies

In this study, an explorative experiment was conducted in which subjects were asked to give route directions to each other in a simulated campus, using a speech recogniser to corrupt the speech in one direction. Data was collected on how the subjects reacted to and recovered from speech recognition errors. The goal was to get clues as to how errors may be handled in spoken dialogue systems, not to test specific error handling strategies. The task was chosen to resemble that of a fairly complex spoken dialogue system and the two subjects were given the role of operator (corresponding to the "dialogue system") and user. The operator could not hear what the user said, and had to read the speech recognition result from a screen. A number of different naive operators were used in order to get varied data on error handling strategies. As opposed to the Wizard-of-Oz method, the operator was treated as a subject as well and the users were openly informed about the setting.

As discussed previously, the problem of transferring results from human–human studies to human–computer dialogue has been shown to be dependent mainly on the limitations that real systems impose on the dialogue. In the current study, the users were told that a speech recogniser was used, and were therefore aware of the fact that complex utterances might not get through. Since speech recognition is regarded as the bottleneck of most complex spoken dialogue systems, the results of this study may be more easily transferred to dialogue systems than results taken from ordinary human–human dialogue.

One thing that does differ in the conversation with humans and machines, even when the channel is equally noisy in both cases, is the amount of common ground (Clark, 1996) that the speakers have before engaging in the conversation. To minimise common ground, the operator and the user were not allowed to meet before or during the experiment. However, both subjects were fully informed about the experimental setting. If the user should not be allowed to form any assumptions about the operator, the question is how the operator should reply. One possibility could be to let the operator type a message, synthesize it and play it back to the user, using a text-to-speech system. However, pilot studies showed that this would be too slow, and that the operator might behave in a "lazy" way, not typing the whole message as it would have been spoken. Another solution could be to let the operator choose or compose the answer from a set of templates, but this would restrict the operator's output, and unexpected behaviour would not be captured. The proposed solution is to let the operator speak freely and distort the speech through a vocoder. The final setting is illustrated in Fig. 1.

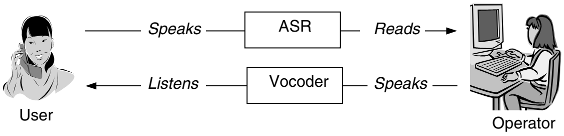


Fig. 1. The setting used in the experiment.

It should be noted that this experimental setting lacks the control that the consistent behaviour of a trained operator would give. Still, this method may be good for explorative studies, which aim at finding new ideas on dialogue behaviour, and especially on how error situations could be handled. It should be followed up by more controlled experiments (preferably with a real system) in order to test the derived hypotheses.

3.2. The domain for the experiment

A general distinction can be made between problem solving and information seeking dialogue (e.g. Flycht-Eriksson, 2001). Most dialogue systems built today are designed for information seeking, such as travel information and stock quotes. The domain used in this experiment is not about information seeking, but direction-giving, which would be classified as problem solving. In such a domain, the user's goal is to get to a specific location and the dialogue system (in this experiment the operator) is used to get route directions. The system (operator) does not know where the user is, and must rely on the user's descriptions of the environment. One important difference from information seeking is that the dialogue system (operator) can establish the user's goal at an early stage in the dialogue and then work towards this goal. In information seeking, the system rarely knows the user's final goal. The dialogue type can affect which types of error handling strategies might be used by speakers, which should be kept in mind when analysing the results.

Dialogue about route descriptions have been studied extensively in so-called Map Task experiments (see Anderson et al., 1991, for a description of a corpus, and Brown, 1995, for an extensive analysis). The question is to what extent these data are applicable to dialogue systems for navigation, since the user (the "follower" in Map Task) has access to the whole map and can talk about absolute directions (such as "north", "south", "up" and "down"). For this experiment, a simulation environment, which is described in the next section, was built to prevent the user from using such information. Although miscommunication has been studied in Map Task experiments previously (Brown, 1995; Carletta, 1996), such experiments have not, to the author's knowledge, been conducted using a speech recogniser.

3.3. Experimental design

3.3.1. Subjects

Sixteen subjects were used, 8 users and 8 operators. All subjects were native speakers of Swedish. The subjects were paired in groups of operator/user. There were 8 women and 8 men, equally balanced as operators and users. Users with low to medium computer experience were chosen (to represent ordinary users), while the operators were chosen with a somewhat higher computer experience and some experience of speech technology, on the assumption that this would make the learning of the operator interface faster. However, since the purpose of the study was to collect data on "natural" human error handling, people with experience in dialogue system design were not used as operators.

3.3.2. Scenarios

The users were given the task to get from one department to another in a simulated campus. The operators' task was to guide the users. The operators had access to a map showing the entire campus to help them with their task. In order to solve the task, the users had to state the goal and continuously describe their current location. When guiding the users, the operators had no direct access to their position, but had to rely on their descriptions of surrounding landmarks. Five different scenarios were given to each pair of subjects, which resulted in 40 dialogues. The order of the scenarios was changed and balanced between pairs, so that general trends after several sessions could be studied independently of scenario.

3.3.3. Material

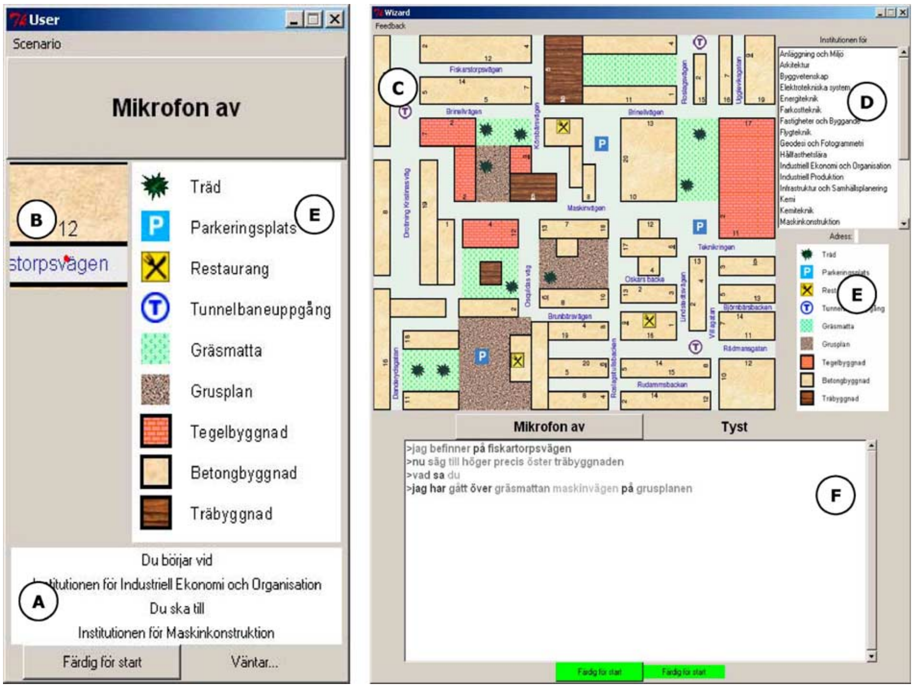
A system for handling the simulation of the campus and for managing the communication between the subjects was built. The user's and the operator's interfaces to the systems are shown in Fig. 2.

At the bottom of the user's screen (A), the scenario was presented. Only a small fraction of the map (B) surrounding the current position was shown (seen from above). The user "walked" in the campus by using the arrow keys on a keyboard. When the user changed direction, the whole map rotated, so that the user always was facing "up", which made it hard for the subjects to talk about "up", "down", "north" or "south". Instead, they had to use landmarks and relative directions.

The operator's map (C) was identical to the user's, except for some street names that were missing on the user's map. The operator could easily look up where the departments were located (D). The user's position was not shown on the operator's map, so the operator had to rely on the user's descriptions of the environment. On each screen, there was a legend explaining the landmarks (E).

Both the user and the operator were wearing headsets and a push-to-talk mechanism was used. The operator's speech was processed through a vocoder and played back directly to the user. The processed speech was fairly easy to understand, according to post-interviews. However, a lot of prosody was distorted, and the users could not tell whether it was a male or female voice.

The user's speech was recognised by a speech recogniser and the recognised string was displayed on the operator's screen (F). An off-the-shelf speech recogniser was used with built-in acoustic models of Swedish. A tri-gram language model was used, trained on a small corpus of invented



Du börjar vid
Institutionen för Industriell Ekonomi och Organisation
Du ska till
Institutionen för Maskinkonstruktion

>jag befinner på fiskartorpsvägen
>nu säg till höger precis öster träbyggnaden
>vad sa du
>jag har gått över gräsmattan maskinvägen på grusplanen

Fig. 2. The user's interface to the left and the operator's interface to the right.

dialogues and transcriptions from pilot studies. The vocabulary was of about 350 words. As Fraser and Gilbert (1991) points out, using speech recognition in a Wizard-of-Oz setting might be tough for the operator, since the recognition result may be hard to present and interpret. In this experiment, the words were coloured in greyscale according to each word's confidence score, so that they should be more easily readable, allowing the operator to get an immediate and overall understanding of the confidence scores of the words. Words that were coloured in darker tones had higher confidence scores, while lighter tones reflected lower confidence scores. Since the operator could not hear the user, an indicator on the screen showed whether the user was speaking or not, in order to facilitate turn taking.

3.3.4. Procedure

The user and operator were informed separately about the experiment and the setting, and the respective computer interfaces were explained to them. After the instructions, the subjects were placed in different rooms and were not allowed to see each other until the experiment was over. They got no information about each other before the experiment. During the experiment, the conductor of the experiment was sitting behind the operator and could see what the operator was doing and hear what the participants said (using headphones). The conductor also assisted in answering any technical questions about the system from the operator during the experiment. The user was sitting alone. The task was interrupted if the subjects did not complete it within ten minutes. There were no instructions on who should start the conversation, who should take the initiative and "lead" the dialogue, or on possible error handling strategies.

After each scenario, both operator and user filled out a questionnaire about the interaction. The questionnaire consisted of a number of statements, for each of which the subjects stated to what extent they agreed. Only one of the statements from the users' questionnaire is discussed in this report: "we did well in solving the task". There was a choice of seven levels of agreement, ranging from "strongly disagree" to "strongly agree". After the whole experiment, both user and operator were interviewed. For the users, the questions mainly concerned how well they thought that they had been understood and if they had understood the vocoder.

3.4. Data analysis

All utterances of the users and operators were transcribed and manually annotated. Each utterance was annotated with regard to which dialogue acts it contained and how well it was understood. Some annotation examples are given in Table 1.

The dialogue act scheme was not intended to be general. Instead, it was constructed to cover the most frequent types of dialogue acts in the current experiment. The purpose of annotating dialogue acts was to find relations between these and the understanding of utterances. Unlike more general schemes (such as Carletta et al., 1997), distinctions were also made between questions and assertions concerning different subtasks in the domain, such as establishing a goal, finding out the users' positions and giving directions. No distinction was made between assertions and answers (which can be seen as subset of assertions), since this would introduce unnecessary ambiguity, especially as there was a lot of miscommunication. The dialogue acts were encoded based on the spoken, not the recognised, utterances. The dialogue acts are presented in Table 2.

Each user utterance was also annotated with regard to how well it was immediately "understood" by the operator. "Understood" here means that the operator continued the dialogue with one interpretation, knowing that it may turn out to be incorrect. It does not necessarily mean that that the operator believed in the interpretation. For example, a clarification question from the operator such as "do you have a tree on your left?" shows that the operator understands that there is a tree on the left from the previous utterance. Based on the user's reaction to this question, the operator may later reject this interpretation. However, it is still annotated as (partially) understood or misunderstood, since this was the immediate interpretation. To estimate the level of understanding, the speech recognition result and the operator's reaction to the utterance were considered. The degree of understanding was classified into four categories:

- *Full understanding* (FULL): The full intention of the utterance was understood.
- *Partial understanding* (PARTIAL): Only a fragment or a part of the full intention was understood.
- *Non-understanding* (NON): No part or fragment of the intended message (with the possible exception of a single vague word) was understood.
- *Misunderstanding* (MIS): The operator continued with an interpretation that was not in line with the user's intention.

Of course, the annotator had no direct access to the speaker's intention for each utterance. However, the interpretation was highly constrained by the task context. If there were speech recognition errors that could lead to misunderstanding, these were only marked as misunderstanding if the operator seemed to continue with an interpretation of them. An example of this is utterance Ua.4 in Table 1. The utterance Oa.5 suggests that the operator interprets the word "nineteen" as correct, so Ua.4 is classified as a misunderstanding. In contrast, there is nothing in utterance Oa.7 or Ob.2 (or subsequent utterances) that suggest that the previous utterance was misunderstood (in this sense), even though there are many misrecognised words.

The data was transcribed and annotated by one main annotator. To check the reliability of the annotation scheme used, two other persons annotated 1/5 of the dialogues (i.e. full dialogues, randomly selected). The annotators were instructed to annotate according the definitions of understanding, understanding levels, and dialogue acts, as they are described above. For dialogue acts, the main annotator agreed with one of the other annotators in 99.1% of the cases and with both of them in 97.5% of the cases. For the understanding levels, the scores were 94.9% and 89.8%. The most common disagreement was between partial and full understanding. These figures were judged good enough to base the analysis on the main annotation.

Table 1
Example annotation

Id	Utterance	Understanding	Dialogue act
O.a1	Take to the right in the crossing and continue. You will have a gravel pitch on your right and then a concrete building and then you will get to a crossing and you can stop there		AssertRoute
U.a2	**I AM THERE** (*I am there*)	full	AssertPosition
O.a3	Okay, then continue, eeh lets see, the concrete house that you have on your right side when you have passed the crossing and continued straight forward, well you should pass it and then you should take to the right directly after that house		Acknowledge AssertRoute
U.a4	**NUMBER FOUR NINETEEN HOUSE NUMBER FIVE** (*number ten in the corner of the house, should I round it?*)	mis	RequestRoute
O.a5	Number nineteen is Machine Construction		AssertRoute
U.a6	**AT A GRAVEL PITH WITH THIRTEEN** (*At a gravel pitch. Am I right then?*)	partial	AssertPosition RequestRoute
O.a7	Okay, then there is a little problem, I must check, wait a moment		Acknowledge AssertProblem RequestActWait
U.b1	**HELLO ELEVEN TWENTY ONE TWELVE** (*yes I am there and where shall I go now*)	non	AssertPosition RequestRoute
O.b2	Repeat		RequestActRepeat
U.b3	**ELEVEN COME TO A WOODEN BUILDING TWENTY NINETEEN** (*eeh now I have come to the wooden building how should I go then*)	partial	AssertPosition RequestRoute
O.c1	Do you have a brick building on your right?		RequestPosition
U.c2	**WHAT HEARD** (*yes I have*)	non	Acknowledge
O.c3	Do you have a brick building on your right?		RequestPosition (Repeat)
U.c4	**WHAT HAS** (*yes I have*)	non	Acknowledge
O.c5	What else can you see than the wooden building?		RequestPosition

The columns denote (from left to right): the speaker (Operator or User) and utterance id, the utterance (in case of a user utterance, first the recognised utterance and then the spoken utterance in italics), the understanding, and the dialogue act label. All examples are translated from Swedish. The confidence shading of the words in the speech recognition results have been transferred to the corresponding English words in the translation.

4. Results

4.1. General results

The 40 dialogues resulted in 736 user utterances (18.4 per dialogue on average). The mean utterance length was 6.7 words. 80% of the 40 scenarios were solved within ten minutes. There were a lot of errors in the recognition results, about 42% word error rate (WER). This was partly due to the users' relatively free speech and partly due to the limited training of the language models (an average 7.3%

Table 2
The dialogue act categories

Label	Example
RequestGoal	Where do you want to go?
AssertGoal	I want to go to the department for machine construction
RequestPosition	Can you see a wooden building?
AssertPosition	I have a tree to my left and a concrete building to my right
RequestRoute	How should I go now?
AssertRoute	After the building, you should take to the left
SignalNonUnderstanding	I do not understand/Please repeat/What did you say?
RequestReady	Are you ready?
AssertProblem	There seems to be a problem
RequestActWait	Please wait
AssertActWait	I am waiting
Acknowledge	Okay/Yes
No	No
Greeting	Hello
Thanks	Thank you

out-of-vocabulary rate). There were large individual differences in terms of understanding. The different operators' average understanding is shown in Fig. 3. In the rightmost bar, the average understanding for all subjects is shown.

As can be seen in the figure, very few of the utterances resulted in misunderstanding. This means that when misrecognitions occurred, the operators were very good at deciding which words were correct and which were not. When there were a lot of misrecognitions, this resulted in partial understanding or non-understanding, instead of misunderstanding. Thus, the operators were very good at error detection. More than 50% full understanding in general may seem to be high compared to the high WER. However all words do not have to be correct for full understanding. Moreover, the WER was not equally distributed between utterances. Some had very low WER and some very high.

In order to find out whether the subjects improved at the task during the five sessions, a trend analysis was tested on several factors. As shown in Fig. 3, there was a large between-pair variance, which makes it hard to find general trends. However, it turned out that the proportion of non-understanding and the number of user utterances (a measure of the length of the dialogue) changed after subsequent sessions (one-way repeated measures ANOVA; $p < 0.05$). An analysis of polynomial contrasts showed that both variables decreased in a linear fashion. The trends are shown in Fig. 4.

The post interviews revealed that, despite of the numerous non-understandings, the users in general experienced that they were almost always understood. It turned out that in many cases, instead of signalling non-understanding—which may seem like the obvious choice—the operators employed other strategies.

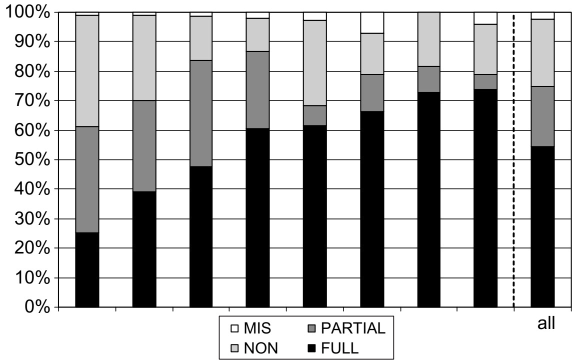


Fig. 3. The different operators' average understanding of the users' utterances across all five sessions.

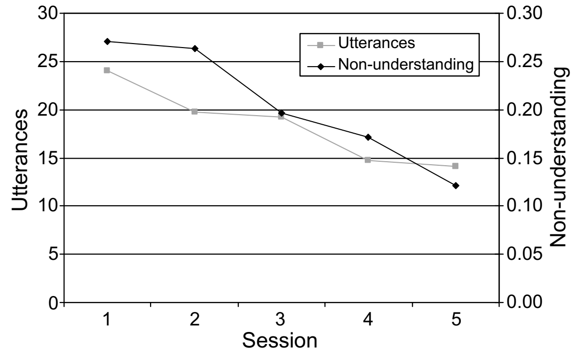


Fig. 4. The average number of user utterances and proportion of non-understanding in each dialogue, as they decrease after subsequent sessions.

4.2. Strategies after non-understanding

The operators' strategies after non-understanding were divided into three categories: SignalNonUnderstanding, AssertRoute and RequestPosition. The three groups were of approximately equal size.

4.2.1. SignalNonUnderstanding

This category includes all reactions to non-understanding where the operator somehow signalled that the utterance was not understood. This includes explicit requests for repetition ("please repeat", "what did you say"), assertions of non-understanding ("I didn't understand"), and repetitions of the same utterance (O.c1–O.c3 in Table 1 is an example of a repetition). Consider the following example:

U.d1: WEST WITH (*that's right*)
O.d2: Please repeat what you said
U.d3: THAT THERE WITH (*that's right*)

In the example, the problem with the recognition is that the expression "that's right" isn't covered by the language model. Therefore, the repetition doesn't lead to a recovery from the problem. Example U.b1–U.b3 in Table 1 is an example where the request for repeat instead triggers the user to rephrase, which leads to a minor recovery (partial understanding).

4.2.2. AssertRoute

This category contains all reactions to non-understanding where the operator gave a new route description without any of the signals of non-understanding mentioned above. Here is an example:

O.e1: Continue a little bit forward
U.e2: STREET THAT THERE HOUSE (*past the wooden house?*)
O.e3: Now, walk around the wooden house. Take left and then right

The only thing that the operator seems to rely on, in this example, is something about a house. Since it is impossible to interpret what the user is trying to say and since the word house does not contribute much (in this domain) to the understanding, this has been classified as a non-understanding. Although it seems like the operator is totally ignoring the user's contribution, the operator utterance implicitly verifies the user's position by referring to a wooden house (something that there are only a few of on the map). If the hypothesis is incorrect and the operator's utterance was out of place, the user has a chance to react so that the recovery process may continue. If it is correct (as in this case), the user will probably perceive the situation as if the utterance was fully understood.

4.2.3. RequestPosition

This category contains all reactions to non-understanding where the operator asked a question about the user's position without any of the signals of non-understanding mentioned above. Here is an example:

O.f1: Do you see a wooden house in front of you?
U.f2: YES CROSSING ADDRESS NOW (*I pass the wooden house now*)
O.f3: Can you see a restaurant sign?

The operator seems unsure of whether the user really can see a wooden house, but instead of asking the user to repeat, another question is asked that is confirming the same hypothesis as the operator wanted to confirm by asking the first question. Another example is Oc.3–O.c5 in Table 1. After the non-understanding, the operator asks about a wooden building (which has been mentioned previously in the dialogue). Since the question is task-related (and not related to what has been said), it implicitly confirms the operator's hypothesis about the user's position without signalling non-understanding (just as with AssertRoute).

4.3. Error recovery

As discussed and exemplified previously, miscommunication may often lead to error spirals, where the user just repeats the non-understood utterance or starts to hyperarticulate. A good error recovery strategy should therefore aim at coming to understanding, or get “back on track”, as quickly as possible after a non-understanding has occurred. In order to evaluate the different strategies based on this criterion, the operator’s understanding of the user’s utterance following a reaction to a non-understanding was studied. As an example, take the sequence b.1–b3 in Table 1. After the first non-understanding, the operator selects the strategy SignalNonUnderstanding. This strategy leads to a partial understanding of the next utterance. The distribution of the operators’ understanding following the different strategies is presented in Fig. 5. The top bar shows the expected distribution, which is the general distribution for all utterances, also shown in the rightmost bar in Fig. 3.

Statistical tests showed that there was no deviation from the expected distribution after AssertRoute and SignalNonUnderstanding, but after RequestPosition there were significantly less non-understandings and instead significantly more partial understandings (goodness-of-fit test; dF = 3; $\chi^2 = 12.52$; $p < 0.01$). This suggests that RequestPosition leads to better recovery from the problem.

Why does RequestPosition lead to less non-understanding? To answer this question, the types of questions that were posed and the reactions to the strategies were analysed further. Approximately 1/3 of the RequestPosition utterances were wh-questions and 2/3 yes/no-questions. This may suggest that the questions constrain the length of the answers from the user and thereby increase the speech recognition performance. However, yes/no-questions do not always result in simple yes/no answers, as the example U.f2 illustrates. Table 3 shows the utterance length following the different strategies. As seen in the table, the utterances following RequestPosition, are not shorter, but longer. The better understanding of these utterances is probably explained by the fact that they constrain the response to the domain and the language models, which increase speech recognition performance. Moreover, the specific question that precedes the response may also constrain the interpretation of the speech recognition result even if it is poor. This is not true for AssertPosition, and may explain why RequestPosition, works better.

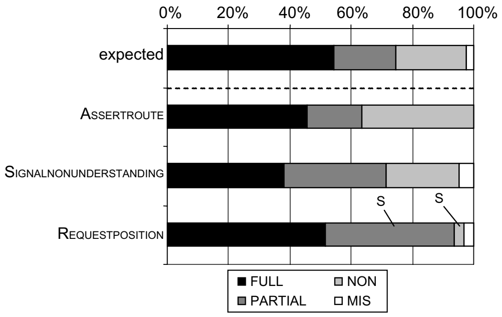


Fig. 5. The understanding of the user’s utterance that follows the operator’s reaction to a non-understanding. “S” marks significant deviation from the expected value.

Table 3
Mean and median length of the utterances (number of words) following the different strategies

	Mean utterance length	Median utterance length
SignalNonUnderstanding	7.4	4
AssertRoute	6.4	5
RequestPosition	8.6	8

4.4. User experience of task success

Fast error recovery can be regarded as a measure of efficiency. But from a user-centred point of view, the experience of using the system should come first, and efficiency should be a means for improving the experience of using the system. Thus, it is interesting to examine how different recovery strategies and other objective measures contribute to the user’s experience. Since there was a large between-pair and within-pair variance (as shown in Figs. 3 and 4), regarding the subjects’ performance, it should be possible to correlate the user’s experience with objective measures for different pairs and sessions. To investigate this, a multiple regression analysis was used in a way similar to the PARADISE evaluation framework for dialogue systems (Walker et al., 2000). The idea behind PARADISE is to find out the relation

Table 4
Results from the regression analysis

Contributing factors	Coeff	SE	*t*-Stat	*p*-Value
Total time	−0.456	0.083	−5.499	<0.001
SignalNonUnderstanding	−0.560	0.262	−2.142	0.039
Non-contributing factors				
Total path				
WER				
Non-understanding				
AssertRoute				
RequestPosition				

between the subjective measure of the user's satisfaction (which can be collected by using a questionnaire) and a number of objective measures (the task success and dialogue costs, such as number of repetitions, WER, etc.). If this relation can be estimated, it is possible to predict the effect on user satisfaction that the tuning of objective parameters will have (such as improving the WER), without having to run expensive user tests (Walker et al., 2000). The method can also be used to give insights into which parameters are important for the user satisfaction of dialogue systems in general and which are not.

The input to the regression analysis is a criterion variable (in the case of PARADISE, the user satisfaction) and a set of predictor variables (the objective measures). The output is a set of coefficients for the predictor variables that describe the relative contribution of each variable for the variation in the criterion variable. Since the user's task in the current study was given beforehand and was quite artificial, it was hard to get a measure of the "user satisfaction". Instead, the user's experience of task success was used. The question "how well do you think that you did in solving the task?" from the questionnaire was used as the dependent factor, which was a rating from 0 to 6. As predictor variables, factors that were likely to affect the user's experience were selected: time to solve the task, the length of the path that the user went, the mean WER, the number of non-understandings, and the number of uses of the error recovery strategies (SignalNonUnderstanding, AssertRoute, RequestPosition). All 40 dialogues were used as data points.

If several predictor variables correlate, they will explain the same variation in the criterion variable, and the result will depend on which predictor variables are selected. It is therefore important to select the variables systematically. Hinkle et al. (1994) describes three procedures for doing this: backward solution, forward solution and stepwise solution. All three were tested, and they all resulted in a significant correlation between the criterion variable and two of the predictor variables ($R^2 = 0.56$; $p < 0.0001$). The contribution of the different variables to the user's experience of task success is shown in Table 4.

As can be seen in the table, the only factors that contributed were time for task completion and the number of non-understandings that the operator had *signalled* (which both had a negative effect). It is interesting that neither the number of non-understandings nor the WER per se had any effect on the user's experience, but only the cases where the user was made aware of the non-understanding.

5. Conclusions and discussion

In the experiments, the high WER caused only a few misunderstandings, but many non-understandings. This suggests that different knowledge sources (such as confidence score, syntactic structure and context) can be used (at least by humans) for detection of errors in the speech recognition result, and for deciding upon appropriate reactions to them. Despite the numerous non-understandings, users reported that they were almost always understood. Unlike most dialogue systems, the operators did not often signal non-understanding. If they did display non-understanding, this had a negative effect on the user's experience of task success. Non-understandings per se had no such effect.

An alternative reaction to non-understanding was to ask task-related questions that were confirming the operator's hypothesis about the user's position. This strategy led to fewer non-understandings of the subsequent user utterance, and thus to a faster recovery from the problem.

On average, the speech recognition performance was poor. As mentioned previously, this was partly due to the users' relatively free speech and partly due to the limited training of the language models. This may seem as non-representative for most dialogue systems. However, without the poor performance, it would not have been possible to collect enough data on non-understanding from a reasonable number of dialogues for quantitative analysis. It is also important to stress that the WER varied a lot between utterances and subjects (as can be seen in Fig. 3), which is often the case in real applications. Some dialogues were smooth and successful, while others were dominated by errors. This also made the experience of task success more varied, which is important for regression analysis. Humans are also probably better at interpreting the bad recognition results than what could be accomplished with a robust interpreter. Thus, the distribution of the levels of understanding may be more representative than the WER.

One question is whether it was the signal of non-understanding per se that led to a lower experience of task success, or if it was the repeated non-understanding of subsequent utterances. However, like SignalNonUnderstanding, AssertRoute did not lead to decreased non-understanding, but unlike RequestPosition, it did not lead to decreased experience of task success. This suggests that it is the signalling of non-understanding that is frustrating and gives the user an experience of task failure. This also shows that efficiency might not be the sole predictor for the user's experience of task success.

The fact that the operators were good at error detection is interesting. The question is to what extent different features (such as confidence scores and contextual information) contributed to this performance. Another experiment has been performed to find the answer to this question, where human subjects were given the task to detect errors in the speech recognition results, given different amount of information (Skantze and Edlund, 2004a).

It would be interesting to find out why the subjects get better at the task after repeated sessions. It is probably due to the fact that they get better at formulating descriptions and route directions. It would also be interesting to find out whether they learn any new error handling strategies. Another question is whether it is mainly the user or the operator that adapt. That is, does a dialogue system have to adapt to the user, or is it enough that the user adapts to the system? No general trends in choice of strategies could be found, probably due to the large inter-subject variance.

The results from this study confirm Brown's (1995) argument that it may be problematic to study understanding by just analysing ordinary human–human dialogue, since the signals of understanding that the speakers send apparently do not have to reflect their true understanding. As Brown points out, the problem of studying understanding in ordinary conversation analysis is that the analyst has no access to what goes on inside peoples' heads. The analyst has to rely on the record of the speakers' behaviour, such as grounding and signals of non-understanding. However, it is not certain at all that these signals reflect the true understanding of the speakers. This is a serious problem if the analyst wants to relate the level of understanding to the speaker's behaviour during conversation. In psycholinguistic laboratory experiments, the comprehension of subjects can be studied by carefully controlling the stimuli and measure the level of understanding after each utterance or fragment. The problem with such experiments is that it is not possible to relate the understanding to an ongoing dialogue that the subject is engaged in. Brown argues that the Map Task method provides a solution to this problem, since the speakers' beliefs about the world are controlled by the experimenter (i.e. what is printed on the maps). Thus, it becomes possible to study miscommunication that arises from the misalignments of the speakers' models of the world. While this facilitates the study of *misinterpretation* (in the sense used by McRoy, 1998), it does not provide information on how people react to *non-understanding*, which the experimental set-up used in this study supports.

Another conclusion drawn by Brown (1995), which is augmented by these results, is that listeners do not primarily strive to arrive at a correct interpretation of utterances. They merely use the utterances as a knowledge source among others to solve the task at hand. In a problem solving task such as guiding, the goal is established early in the dialogue and the listener can focus on solving the task by working towards this goal. For the design of spoken dialogue systems in similar domains, the results suggest that when non-understandings occur, a good domain model and robust parsing techniques should be used to pose relevant questions to the user (instead of signalling non-understanding), so that errors can be efficiently resolved without the user experiencing the dialogue as problematic and dominated by explicit error handling.

One important question is if these results can be applied to domains that are not about navigation. In tasks where a single slot has to be filled by using specific words, there may not be any other option than to signal non-understanding and thereby encourage repetition. In certain other, more complex domains, strategies similar to RequestPosition, are likely to be applicable. To illustrate the possible applications, some examples from different domains will be given. The most obvious are dialogues where the operator is diagnosing a problem. If the system does not understand the answer to one question, it might be better to ask another one instead of signalling non-understanding, given that there are several ways to pinpoint the problem. A similar strategy may also be useful if speech technology is to be used in games, where non-understanding may be frustrating for the user and task-related questions may be used to guide the conversation along certain paths. It should also be possible to ask task-related questions after non-understanding in information-browsing domains. As an example, take the apartment broker domain, which is the domain for the AdApt spoken dialogue system (Gustafson et al., 2000). The following (invented) dialogue illustrates:

U.g1: Tell me about the bathroom. (*full understanding*)
S.g2: It is a tiled bathroom and it has a bathtub
U.g3: Is there anything else you can tell me about the apartment? (*non-understanding*)
S.g4: Do you want me to tell you more about some specific part of the apartment?
U.g5: Yes, tell me about the kitchen

In this case, the system's response after the non-understanding (S.g4) happens to be in place and does not signal non-understanding. Just like RequestPosition, it is a task-related question that may constrain the interpretation of the user's next utterance. The results from this study suggest that it may have a greater potential for recovering from the error than an explicit signal of non-understanding would have. If it had not been in place, the user would still have a chance to correct the system. Furthermore, the user may not always have a fixed idea of what she wants to know and may experience a question such as S.g4 as helpful. In a multimodal system (such as AdApt), it is also often possible to switch modality and let the user provide the information in an alternative way. Oviatt and VanGent (1996) have shown that modality switching is a successful method for recovering from error. The results from this study suggest that it may be better to do this without signalling non-understanding:

S.h1: Which area are you interested in?
U.h2: I would like to live near the water (*non-understanding*)
S.h3: Can you mark exactly on the map

Utterance S.h3 could start with "Sorry, I couldn't understand", but leaving this out may improve the user experience of task success. The operators in this study often used some word in the poor recognition results when formulating their requests after non-understandings. This is likely to increase the probability of posing a question that seems relevant.

Of course, these are just invented examples of what could possibly be done in other domains to recover from non-understanding without signalling non-understanding. It would be interesting to perform experiments similar to this in other domains to find out if humans benefit from similar strategies, or if they have to signal non-understanding. Our next step is to apply the results from this study to a complete dialogue system (Edlund et al., 2004). To do this, we have to develop better techniques for world-level error detection (Skantze and Edlund, 2004a) and robust interpretation (Skantze and Edlund, 2004b), so that the system may detect non-understandings correctly and possibly find some correct words that it may use to increase the likelihood of posing relevant error recovery questions.

Acknowledgement

This research was carried out at the Centre for Speech Technology, a competence centre at KTH, supported by VINNOVA (The Swedish Agency for Innovation Systems), KTH and participating Swedish companies and organizations.

References

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

Amalberti, R., Carbonell, N., Falzon, P., 1993. User representations of computer systems in human–computer speech interaction. International Journal of Man–Machine Studies 38 (4), 547–566.

Anderson, A.H., Bader, M., Bard, E.G., Boyle, E.H., Doherty, S.C., Garrod, S.C., Isard, S.D., Kowtko, J.C., McAllister, J., Miller, J., Sotillo, C.F., Thompson, H.S., Weinert, R., 1991. The HCRC map task corpus. Language and Speech 34 (4), 351–366.

Balentine, B., Morgan, D.P., Meisel, W.S., 2001. How to Build a Speech Recognition Application: Second Edition: a Style Guide for Telephony Dialogues. Enterprise Integration Group, San Ramon CA.

Bell, L., Gustafson, J., 1999. Repetitions and its phonetic realizations: Investigating a Swedish database of spontaneous computer-directed speech. In: Proceedings of ICPhS '99, pp. 1221–1224.

Brennan, S.E., 1996. Lexical entrainment in spontaneous dialog. Proceedings of ISSD, 41–44.

Brown, G., 1995. Speaker, Listeners and Communication. Cambridge University Press, Cambridge.

Carletta, J., Mellish, C.S., 1996. Risk-taking and recovery in task-oriented dialogue. Journal of Pragmatics 26, 71–107.

Carletta, J.C., Isard, A., Isard, S., Kowtko, J., Doherty-Sneddon, G., Anderson, A., 1997. The reliability of a dialogue structure coding scheme. Computational Linguistics 23 (1), 13–31.

Clark, H.H., 1994. Managing problems in speaking. Speech Communication 15, 243–250.

Clark, H.H., 1996. Using Language. Cambridge University Press, Cambridge.

Dahlbäck, N., Jönsson, A., Ahrenberg, L., 1993. Wizard of Oz studies—why and how. In: Proceedings from the 1993 International Workshop on Intelligent User Interfaces, pp. 193–200.

Edlund, J., Skantze, G., Carlson, R., 2004. Higgins—a spoken dialogue system for investigating error handling techniques. In: Proceedings of ICSLP.

Flycht-Eriksson, A., 2001. Domain knowledge management in information-providing dialogue systems. Lic. Thesis, Thesis No. 890, Linköping University.

Fraser, N.M., Gilbert, G.N., 1991. Simulating speech systems. Computer Speech and Language 5, 81–99.

Gustafson, J., Bell, L., Boye, J., Carlson, R., Edlund, J., Granström, B., House, D., Wirén, M., 2000. AdApt—a multimodal conversational dialogue system in an apartment domain. In: Proceedings of ICSLP 2, pp. 134–137.

Hinkle, D.E., Wiersma, W., Jurs, S.G., 1994. Applied Statistics for the Behavioral Sciences, third ed. Houghton Mifflin Company, Boston.

Hirst, G., McRoy, S., Heeman, P., Edmonds, P., Horton, D., 1994. Repairing conversational misunderstandings and non-understandings. Speech Communication 15, 213–230.

Larsen, 2003. On the usability of spoken dialogue systems. Ph.D. Thesis, Report No. R03-1011, Aalborg University.

Levow, G., 1998. Characterizing and recognizing spoken corrections in human–computer dialogue. In: Proceedings of COLING/ACL '98.

McRoy, S.W., 1998. Preface—detecting, repairing and preventing human-machine miscommunication. International Journal of Human–Computer Studies 48, 547–552.

Oviatt, S., Levow, G., MacEachern, M., Kuhn, K., 1996. Modeling hyperarticulate speech during human–computer error resolution. In: Proceedings of ICSLP '96.

Oviatt, S., VanGent, R., 1996. Error resolution during multimodal human–computer interaction. In: Proceedings of ICSLP, 1, pp. 204–207.

Paek, T., 2001. Empirical methods for evaluating dialog systems. In: ACL 2001 Workshop on Evaluation Methodologies for Language and Dialogue Systems.

Schegloff, E.A., 1992. Repair after next turn: the last structurally provided defence of intersubjectivity in conversation. American Journal of Sociology 97 (5), 1295–1345.

Skantze, G., Edlund, J., 2004a. Early error detection on word level. In: ISCA Tutorial and Research Workshop on Robustness Issues in Conversational Interaction.

Skantze, G., Edlund, J., 2004b. Robust interpretation in the HIGGINS spoken dialogue system. In: ISCA Tutorial and Research Workshop on Robustness Issues in Conversational Interaction.

Shin, J., Narayanan, S., Gerber, L., Kazemzadeh, A., Byrd, D., 2002. Analysis of user behaviour under error conditions in spoken dialogs. In: Proceedings of ICSLP.

Walker, M., Candace, A., Kamm, A., Litman, D., 2000. Towards Developing General Models of Usability with PARADISE. Natural Language Engineering 6 (3-4), 363–377.

Weigard, E., 1999. Misunderstanding: the standard case. Journal of Pragmatics 31, 763–785.