CHAPTER 3

Miscommunication and error handling

In the previous chapter, conversation and spoken dialogue systems were described from a very general perspective. In this description, a fundamental issue is missing: how to deal with *uncertainty* and *errors*. Understanding is not something that speakers can take for granted, but something they constantly have to signal and monitor, and something that will sometimes fail. In this chapter, we will first review how humans ensure understanding in communication and what happens when miscommunication occurs. We will then discuss the concept of error in the contexts of human-human and human-computer dialogue, and review research done on how errors in spoken dialogue systems may be detected and repaired.

3.1 Miscommunication and grounding

3.1.1 Miscommunication

Miscommunication is a general term that denotes all kinds of problems that may occur in dialogue. One reason for miscommunication being fairly frequent in dialogue may be explained by *the Principle of Parsimony*[3] (Carletta & Mellish, 1996):

> The Principle of Parsimony states that people usually try to complete tasks with the least effort that will produce a satisfactory solution. In task-oriented dialogue, this produces a tension between conveying information carefully to the partner and leaving it to be inferred, risking a misunderstanding and the need for recovery. (p. 71)

[3] Also called Ockham's Razor.

For example, speakers may produce ambiguous referring expressions, use fragmentary utterances which can only be understood assuming a certain common ground between the speakers, and may use extremely reduced phonetic realisation of utterances. These are all different ways of increasing efficiency and introducing risk – there is always the possibility that listeners will not interpret them correctly. However, it may not be worth the effort to produce unambiguous expressions and canonical pronunciations, if the intended messages usually are interpreted correctly or if it is easy to diagnose and correct the problem when they are not.

There are different ways of analysing miscommunication phenomena. 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 listener fails to obtain any interpretation at all, or is not confident enough to choose a specific interpretation, a *non-understanding* has occurred. One important difference between non-understandings and misunderstandings is that non-understandings are noticed immediately by the listener, while misunderstandings may not be identified until a later stage in the dialogue. Some misunderstandings might never be detected at all. The same utterance may, of course, give rise to both misunderstanding and non-understanding, that is, parts of an utterance may be misunderstood while others are not understood. Successful communication may be referred to as *correct understanding* or just *understanding*[4]. Misunderstanding and correct understanding are similar in that the listener chooses a specific interpretation and assumes understanding, which is not the case for non-understanding.

A second way of analysing miscommunication is by the *action level* with which the problem is associated. Both Allwood et al. (1992) and Clark (1996) make a distinction between four *levels of action* that take place when a speaker is trying to communicate something to a listener. The authors use different terminologies, but the levels are roughly equivalent. The terminology used here is a synthesis of their accounts. Suppose speaker A proposes an activity for listener B, such as answering a question or executing a command. For communication to be "successful", all these levels of action must succeed (listed from higher to lower):

- Acceptance: B must accept A's proposal.
- Understanding: B must understand what A is proposing.
- Perception: B must perceive the signal (e.g., hear the words spoken).
- Contact: B must attend to A.

More fine-grained analyses are of course also possible. The understanding level may for example be split into discourse-independent meaning (e.g., word meaning) and discourse-dependent meaning (e.g., referring expressions). The order of the levels is important; in order

[4] Brown (1995) prefers the term *adequate* interpretation (or understanding). According to her, every utterance is understood for a particular purpose on a particular occasion. There is, in most conversational settings, not a single interpretation which is "correct", but a number of adequate interpretations which will serve to fulfil the purpose of the speakers' joint project.

to succeed on one level, all the levels below it must be completed. Thus, we cannot understand what a person is saying without hearing the words spoken, we cannot hear the words without attending, and so on. Clark calls this the *principle of upward completion*.

Now, misunderstanding and non-understanding may occur on all these levels of action. B might correctly hear the words spoken by A, but misunderstand them or not understanding them at all. B might also attend to A speaking, but misrecognise the words spoken, or not hear them at all. As Dascal (1999) notes, this is reflected in the different names for misunderstanding in the English language, such as: *mishear*, *misrecognise*, *misinterpret*, *misinfer*, *misconclude*. In this thesis, however, we will stick to the terms misunderstanding and non-understanding to denote the general phenomena, and state which level is concerned if necessary.

It is questionable, however, whether failure on the level of acceptance really should be classified as miscommunication. If someone rejects a request or does not accept a proposal, we could easily say that the participants have succeeded in their communication. If A and B engage in a dialogue about possible activities and A suggests that they should go and see a movie, and B then rejects this proposal because he has already seen the film, we may say that they have successfully communicated that this is not an option.

A third distinction can be made depending on the *scope* of the miscommunication. Misunderstanding and non-understanding may concern not only the whole utterance, but also parts of it, resulting in *partial misunderstanding* and *partial non-understanding*:

(19) A: I have a red building on my left.
B (partial misunderstanding):
How many stories does the blue building have?
B (partial non-understanding):
What colour did you say?
Did you say red?

3.1.2 Grounding

Communication can be described as the process by which we make our knowledge and beliefs *common*, we add to our *common ground*. Clark (1996) defines the notion of common ground as follows:

> Two people's common ground is, in effect, the sum of their mutual, common, or joint knowledge, beliefs, and suppositions. (p.92)

When engaging in a dialogue, two people may have more or less in their common ground to start with. During the conversation, they try to share their private knowledge and beliefs – to add them to the common ground. As Clark (1996) points out, however, the process by which speakers add to the common ground is really a joint project, in which the speakers have to cooperatively ensure mutual understanding. A speaker cannot simply deliver a message and hope that the listener will receive, comprehend and accept it as correct. They have to constantly send and pickup signals about the reception, comprehension and acceptance of the information that is communicated. This is the process of *grounding*.

3.1.2.1 Evidence of understanding

In order to ground information, people give *positive* and *negative evidence of understanding* to each other. According to Clark, each contribution to the common ground requires a *presentation phase* and an *acceptance phase*. In the presentation phase, speaker A presents a signal for the listener B to understand; in the acceptance phase, B provides evidence of understanding. However, speaker B may in the same turn start a new presentation phase. Thus, each utterance can be said to communicate on two different tracks. On Track 1, knowledge and beliefs about the topic at hand are exchanged. At the same time, communication about understanding is (implicitly or explicitly) performed on Track 2. In Clark's words:

> Every presentation enacts the collateral question "Do you understand what I mean by this?" The very act of directing an utterance to a respondent is a signal that means "Are you hearing, identifying, and understanding this now?" (Clark, 1996, p.243)

The term *evidence of understanding* is closely related to the term *feedback*. The latter term is generally used to denote the information that an agent may receive about the consequences of the agent's actions. In this thesis, we will use the term evidence of understanding, which more precisely denotes feedback that concerns the management of understanding. For example, Allwood et al. (1992) use the term *linguistic feedback* to denote mechanisms by which interlocutors signal their understanding, but also attitudinal reactions and answers to yes/no-questions.

In Clark's account, some kind of positive evidence of understanding is required for each contribution to be considered as common. Clark & Schaefer (1989) list five different types of positive evidence:

1. The hearer shows *continued attention*.
2. An initiation of a *relevant next contribution*, for example an answer to a question.
3. An *acknowledgement* like "uh huh" or "I see".
4. A *demonstration* of understanding, for example a paraphrase.
5. A *display* of understanding, i.e., a repetition of some (or all) of the words used.

Evidence can be more or less strong. The types are listed above roughly from weak to strong: Evidence 1 and 3 only shows that the listener *thinks* that he understands; there is no real *proof* that the content of the utterance is really understood. In the words of Schegloff (1982): "'uh huh', 'mm hmm', head nods and the like at best *claim* attention and/or understanding, rather than *showing* it or *evidencing* it" (p. 78). Evidence 2 may indicate that some of the contents are understood correctly, but it is only evidence 4 and 5 that actually *prove* that (some of) the contents were correctly understood or perceived. As Traum (1994) points out, evidence 4 might actually be stronger than evidence 5, since the listener shows that he has processed the content on some deeper level.

Display of understanding may be given as *separate* communicative acts, with the main purpose of displaying understanding. These may be called *display utterances* or *echoic responses* (Katagiri & Shimojima, 2000). Here is an example:

(20) A: I have a red building on my left.
B: *A red building, ok, what do you have on your right?*

Display utterances and acknowledgements may also be given without keeping the turn, so-called *backchannels* (Yngve, 1970) or *continuing contributions* (Clark, 1996):

(21) A: I have a red building ...
B: *a red building*
A: ... on my left ...
B: *mhm*
A: ... and a blue building on my right.

An important function of such mid-utterance evidence is that it may denote which parts of the presentation utterance it concerns.

Display of understanding is very often *integrated* in the next communicative act, with its main purpose belonging to Track 1:

(22) A: I have a red building on my left.
B: *How many storeys does the red building have?*

3.1.2.2 The grounding criterion

In example (22) above, B could have used the pronoun "it" to refer to the building, but instead chooses the full definite description, which displays B's understanding. Thus, there is always a range of different realisations of the same propositional content (on Track 1), but which may provide different amounts of evidence (on Track 2). How do we, then, choose what strength of evidence to give? Clark (1996) defines grounding as follows:

> To ground a thing […] is to establish it as part of common ground *well enough for current purposes*. (p.221, italics added)

Thus, the requirements on how much evidence is needed vary depending on the current purposes. Clark calls these requirements the *grounding criterion*. There are at least three important factors that should govern the choice of what evidence to give. First, the level of uncertainty is of course an important factor. The more uncertain we are, the more evidence we need. A second important factor is the *cost of misunderstanding* and *task failure*. As less evidence is given, the risk that a misunderstanding occurs will increase – thereby jeopardizing the task the speakers may be involved with. However, a task failure may be more or less serious. Consider the following example:

(23) A: *Welcome to the travel agency. Ann here. How may I help you?*
B: Hi there, I would like to book a trip to Paris.
A: *Ok, to Paris, from where do you want to go?*

In this example, B's statement about the destination requires strong evidence (such as the display in the example), since booking a ticket with the wrong destination has serious effects. On the other hand, when Ann is presenting her name in the beginning of the conversation, there is typically no need for B to provide any evidence.

Why do we not always provide strong evidence, just to be certain, then? This is explained by the Principle of Parsimony, as discussed previously – people strive to be economical and efficient in their language use. Clark (1996) calls this the *principle of least effort*:

> All things being equal, agents try to minimize their effort in doing what they intend to do. (p224)

Thus, the third important factor for choosing what evidence to provide is the cost of actually providing the evidence and the possible reactions to the evidence.

Since miscommunication may occur on different levels of actions, evidence may also be given on these different levels. For example, the utterance "I heard what you said, but I don't understand" is an explicit way of giving positive evidence on the perception level, but negative evidence on the understanding level. When positive evidence is given on one level, all the levels below it are considered complete. Clark (1996) calls this the *principle of downward evidence*.

3.1.2.3 The requirement of positive evidence

As pointed out by Traum (1994), there is a problem with Clark's strong requirement of positive evidence. Since an acceptance utterance also can be regarded as a presentation (of some evidence) and all contributions require positive evidence, not just lack of negative evidence, the acceptance should require another acceptance (with positive evidence), and so on ad infinitum. Clark's solution to this problem is that each piece of evidence provided by one speaker in turn requires less evidence from the other speaker, so that the need for evidence eventually fades out. However, it is not entirely clear when, why and how the requirement for positive evidence disappears.

This problem is due to the explicit requirement that each contribution needs some sort of positive evidence:

> What distinguishes this model is the requirement of positive evidence. In traditional accounts, Roger could assume that Nina understood him unless there was evidence to the contrary. (Clark, 1996, p. 228)

But there are a number of communicative situations when we clearly do not require positive evidence. For example, a lecturer does not need continuous positive evidence from all hearers to assume that they are listening. We may also send an email without requiring positive evidence that it is received and read. In these cases, we may instead monitor that we do not get negative evidence (such as someone in the audience falling asleep or an error message from the mail server). In other cases, we do indeed require positive evidence. This, of course, depends on the grounding criterion, as discussed previously. If lack of negative evidence may be sufficient in these situations, why would it never be sufficient in spoken dialogue? Clark states that every contribution needs positive evidence, but it is quite unclear what is meant by a contribution. Is it the whole communicative act? Or is each semantic concept a contribution? Example

(23) above illustrates that there are certainly pieces of information for which a speaker does not require positive evidence.

As indicated in the quote above, the reason that Clark puts this strong constraint into his model is to distinguish the account from the naive view that speakers always assume understanding as long as there is no negative evidence. However, there is a middle way – we could assume that people sometimes require positive evidence and sometimes just lack of negative evidence, depending on the grounding criterion. If speaker A presents some signal, he may require positive evidence of some strength (such as a display of understanding). When this evidence is given by B, the participants may determine that the signal has been grounded sufficiently, unless A gives some sort of negative evidence in return. If the grounding criterion would have been even higher, further positive evidence may have been required. It is also important to remember that once a piece of information has been considered as being grounded, there may also an option to go back and repair it later on if it turns out to be wrong.

3.1.3 Repair and recovery

Negative evidence may be given when some sort of miscommunication has occurred. If speaker B has a problem hearing, understanding or accepting a contribution from speaker A (i.e., some sort of non-understanding), speaker B may give negative evidence of understanding:

(24) A: I have a blue building on my left.
B: *What did you say?*

On the other hand, if speaker B accepts the contribution and gives some sort of positive evidence, this evidence may tell speaker A that a misunderstanding has occurred (for example if B misheard the utterance). Speaker A may then initiate a repair:

(25) A: I have a blue building on my left.
B: *How many storeys does <u>the brown building</u> have?*
A: I said a **blue** building!

Schegloff (1992) calls this latter repair type *third-turn repair*, which indicates that the error is detected and initiated in the third turn, counting from the source of the problem. This notion may also be extended to *first-turn repair*, *second-turn repair*, and *fourth-turn repair* (McRoy & Hirst, 1995). First-turn repair is the same thing as self-corrections, that is, a kind of disfluency (see 2.2.4). Second-turn repair means that the detection occurs and the repair is initiated in the second turn, as in example (24).

Hirst et al. (1994) provide a more general way of analysing the cause for repair:

> Participants in a conversation rely in part on their expectations to determine whether they have understood each other. If a participant does not notice anything unusual, she may assume that the conversation is proceeding smoothly. But if she hears some-

> thing that seems inconsistent with her expectations, she may hypothesize that there has been a misunderstanding, either by herself or the other, and produce a repair - an utterance that attempts to correct the problem. (p.223)

Thus, not only direct evidence of understanding, but inconsistencies in general, may act as sources for detecting errors. This may lead to error detection and repair at later stages in the dialogue and give rise to for example fourth-turn repair:

(26) A: I am on Blackberry Street.
B: *Take to the left.*
A: Ok, now I am on Cranberry Street.
B: *Weren't you on Blueberry Street before you turned?*

"Repair", in this context, means that the speakers try to identify and remove (or correct) an erroneous assumption which is caused by a misunderstanding. In the case of non-understanding, the speakers are not trying to repair an erroneous assumption, but instead recover understanding. In this thesis, the terms *misunderstanding repair* and *non-understanding recovery* will therefore be used, which correspond to third-turn and second-turn repair, respectively.

The same factors that influence the choice of positive evidence (uncertainty, cost of task failure, and cost of providing evidence) apply, of course, to the choice of negative evidence. In other words, they apply to the choice of grounding behaviour in general.

3.1.4 Clarification

When a non-understanding recovery (or second-turn repair) is initiated with a request after partial or full non-understanding, it is often called a *clarification request*. If the clarification is due to a lack of hypotheses, the clarification can be initiated with a *request for repetition* (formed as a wh-request). If the clarification is due to a lack of confidence, it can be initiated with a *request for confirmation* (formed as y/n-request). We can also make a distinction between partial and complete clarification requests, that is, whether they concern parts of the previous utterance (concept-level clarification) or the complete previous utterance (utterance-level clarification). Examples of combinations of these are provided in Table 3.1.

Table 3.1: Categorisation of clarification requests, depending on whether they concern the complete previous utterance or parts of it, and whether they express a request for confirmation or repetition.

Scope	**Request**	**Example**
Partial	Confirm	Did you say *red*?
Partial	Repeat	What colour did you say?
Complete	Confirm	Did you say that you have a red building on your left?
Complete	Repeat	What did you say?

While a clarification request always gives some sort of negative evidence, it may also give positive evidence at the same time, concerning other parts of the utterance:

(27) A: I have a red building on my left.
B: *Did you say that the building was red?*

Clarification requests may (as other CA's) be classified based on their *form* and *function*. Purver (2004) presents a study on the different forms of clarification requests that occur in the British National Corpus. The different forms that were identified and their distributions are presented in Table 3.2.

Table 3.2: The first two columns show the distribution of different clarification forms in the British National Corpus according to Purver (2004). This is complemented with examples, as well as a mapping to the categories presented in Table 3.1.

Form	Distr.	Example	Scope	Request
Non-reprise clarifications	11.7 %	*What did you say?*	Complete	Repeat
Reprise sentences	6.7 %	*Do you have a red building on your left?*	Complete	Confirm
WH-substituted reprise sentences	3.6 %	*What can you see on your left?*	Partial	Repeat
Reprise sluices	12.9 %	*A red what?*	Partial	Repeat
Reprise fragments	29.0 %	*Red?*	Partial	Confirm
Gaps	0.5 %	*A red …?*	Partial	Repeat
Gap fillers	4.1 %	*A: I see a red…* *B: building?*	Partial	Confirm
Conventional	31.1 %	*Huh? Pardon?*	Complete	Repeat
Other	0.5 %			

Different approaches have been taken to classify the functions, or readings, of clarification requests. Ginzburg & Cooper (2001) make a distinction between the *constituent* and the *clausal* reading. The following example, with paraphrases, illustrates the difference:

(28) A: Did Bo leave?
B: *Bo?*
clausal: Are you asking whether Bo left?
constituent: Who's Bo?

The clausal reading can, more generally, be understood as "Are you asking/asserting P?", or "For which X are you asking/asserting that P(X)?" and the constituent reading as "What/who is X?" or "What/who do you mean by X?". Purver et al. (2001) adds the *lexical* reading to this list, which could be paraphrased as “Did you utter X?" or “What did you utter?”, that is, an attempt to identify or confirm a word in the source utterance, rather than a part of the semantic content of the utterance (as in the clausal reading).

As pointed out by Schlangen (2004), these different readings can be mapped to the different levels of action (as described in 3.1.1). Such a mapping is shown in Table 3.3, where the understanding level has been split into two levels.

Table 3.3: Mapping between the readings identified by Purver et al. (2001) and levels of action, loosely based on Schlangen (2004). The rightmost column shows the distribution in the British National Corpus according to Purver (2004).

Level		Reading	Distr.
Understanding	Understanding the meaning of fragmentary utterances. Mapping from discourse entities to referents.	constituent	14.4 %
	Understanding syntax, semantics and speech act.	clausal	47.1 %
Perception	Hearing the words that were spoken.	lexical	34.7 %
		other	3.9 %

It is also possible to imagine clarification on the acceptance level. Take the following example:

(29) A: I think we should paint the house pink.
B: *Pink?*

We could make a reading of this where B means “Pink?, that’s an ugly colour I would never consider.” In this case, B has no problem with hearing what was said, nor understanding what A means by “pink”, he just has a problem accepting this. However, as discussed in 3.1.1, this should perhaps not be regarded as a case of miscommunication.

By the rules of upward completion and downward evidence, a clarification on one level (i.e., negative evidence) also provides positive evidence on the levels below it. For example, if B says (or implies) “Who’s Bo?” in a clarification request, A gets positive evidence that B has perceived the words and understood the speech act, but negative evidence about B’s abilities to find a referent to the entity “Bo”.

3.2 Errors in spoken dialogue systems

Mostly due to the error prone speech recognition process, a dialogue system can never know for certain what the user is saying, it can only make *hypotheses*. Thus, it must be able to deal with *uncertainty* and *errors*. Before discussing error handling in spoken dialogue systems, we will discuss the concept of error in the context of human-human and human-computer dialogue.

3.2.1 What is an error?

In the psychological ("human factors") tradition, errors by humans have been defined in the following way:

> a generic term to encompass all those occasions in which a planned sequence of mental or physical activities fails to achieve its intended outcome, and when these failures cannot be attributed to the intervention of some chance agency. (Reason, 1990, p. 9).

In this tradition, a distinction is made between slips (unintentional action) and mistakes (intentional but mistaken action). The expression "slip of the tongue" suggests that the term "error" also may be applied to speech, and that humans indeed make errors when engaging in a dialogue. In this view, an ambiguous referring expression or a self-correction may fail to "achieve its intended outcome" or at least make the communication less efficient than the speaker ideally would wish. However, there is a problem with the concept of "error" in spoken dialogue between humans. As Clark (1996) points out, it is not at all obvious who has actually made the mistake when miscommunication occurs. Is it the speaker for his muddy pronunciation, or the listener for not listening closely enough? As discussed previously, speakers always try to cooperatively balance efficiency against risk. Thus, it may be inadequate to consider misunderstandings as mistakes – they may be part of an agreed compromise.

From a system design perspective, however, an error can be defined as a deviation from an *expected output*. From this perspective, it may be argued that it is only the system that makes errors. Human self-corrections, for instance, are not errors but just another type of input that the system should be built to handle.

The problem is that e*xpected output* is not trivial in this context. In the case of a sorting algorithm, where the input is a list of entities with some associated numeric values, the expected output can be mathematically defined. However, in the case of input such as human speech, the expected output, from for example a speech recogniser, is not possible to define in such a way. First, the mapping from speech to words is something that humans have established in informal contracts with each other. Second, the amount of information carried by the audio signal is vast and filled with noise. Third, the mapping is often ambiguous and dependent on how much context is considered. For example, if an utterance sounds like /wʌn tuː tiː/, it is not obvious what the expected output from a speech recogniser for the third word should be. Heard in isolation, it sounds like "tea", but interpreted in context, "three" is probably a better guess (maybe pronounced by someone with a foreign accent). We would probably want to ask the speaker what was actually intended. However, this person may not be available or he might not remember or be able to consciously reflect over what was actually meant. Expected output for such input is therefore often defined as what a human observer would make of the task at hand. Such a metric is problematic for several reasons, including that humans will differ in their judgement[5] and that the given input and output must be humanly comprehensible. It also leaves no room for the possibility that an automatic process may perform better than a human. Still, the metric is often used, and speech recognisers are commonly measured against a human-made gold standard.

Another way of defining expected output for a system is to relate it to usability. If a spoken dialogue system is designed to meet some human need, then it meets expectations if its users are satisfied; otherwise, it does not. A problem here is that although this is applicable to a dialogue system as a whole, it is considerably harder to relate to the different sub-processes in the system, although attempts have been made (e.g., Walker et al., 2000a). Expectation based on usability is the one that most closely relates to the over-all goal of a dialogue system, but comparing with human performance may be easier to evaluate, especially for sub-processes.

3.2.2 Under- and over-generation

Given an expected output of a process, two types of errors may be distinguished: *under-generation* and *over-generation*. Errors, then, would occur when the process fails to produce some of the expected output, or adds unexpected output, or a combination of both. For ASR, the terms *deletions* and *insertions* are often used for these kinds of errors. A combination of an insertion and a deletion (at the same point in the output) is called a *substitution*. An example is shown in Table 3.4.

Table 3.4: Example of a deletion (DEL), insertion (INS) and a substitution (SUB).

Spoken	*I*	*have*	*a*	*large*		*building*	*on*	*my*	*left*
Recognised	I	have		large	blue	building	on	my	right
			DEL		**INS**				**SUB**

As a measure of the quantity of errors, the *word error rate* (WER) is often used. It is computed by dividing the sum of all insertions, deletions and substitutions (possibly weighting these differently) with the number of words in the original utterance. Correspondingly, *concept error rate* (CER) is used for measuring the quantity of errors on the semantic level, after the utterance has been interpreted.

A process may have a tendency or be tweaked towards over-generation or under-generation. For example, an ASR under-generates if its confidence threshold is set high, and over-generates if it is set low. A rigid parser is likely to under-generate interpretations (by rejecting input that is partially flawed) and a key word spotter may over-generate (by assigning interpretations to any semantically rich word). Under- and over-generation may well occur simultaneously, but increasing one tends to decrease the other. For categorisation tasks, over-generation

[5] Lippmann (1997) reports a 4% transcription error rate for spontaneous conversations recorded over the telephone.

results in lower precision and higher recall, whereas under-generation results in the opposite. These error types result in the two types of miscommunication discussed in 3.1.1; over-generation in misunderstanding and under-generation in non-understanding.

In many classification tasks, the aim is an equal ratio of error types. In spoken dialogue systems, this may not always be optimal, since the two error types have different effects on the dialogue: non-understanding leads to more repetitions and slower progress, while misunderstanding leads to unexpected responses from the system or to wrong actions (task failure) and erroneous assumptions that may be hard to repair. These different consequences are very important to bear in mind when it comes to error handling, and we will return to this issue later on.

The characterisation of over-generation and under-generation above is summarised in Table 3.5.

Table 3.5: Two basic types of error and relating concepts.

	Over-generation	**Under-generation**
Categorisation	Low precision	Low recall
ASR error	Insertions	Deletions
Miscommunication	Misunderstanding	Non-understanding
Consequence	Task failure	Repetitions

3.2.3 Sources of uncertainty and errors

A common observation is that the speech recognition process is the main source of errors in spoken dialogue systems (e.g., Bousquet-Vernhettes et al., 2003; Bohus, 2007). The reason for this is that the input to the ASR exhibits a very large amount of variability. First, there is of course variability between speakers due to factors such as age, gender, anatomy and dialects. Factors such as speaking rate, stress, and health conditions may also vary within the same speaker. Add to this the variability in the channel, such as background noise and microphone properties, and the result is a very large spectrum of different ways the same text may be realised in the waveform that the ASR is supposed to decode. It is of course not possible to model all this variability, nor has the ASR access to all the knowledge sources that a human listener has, such as semantic relations, discourse history (beyond the current utterance) and properties of the domain. Another problem is that the vocabulary and language models used by the ASR never can cover all the things that users may say, which results in *out-of-vocabulary* (OOV) and *out-of-grammar* (OOG) problems with unpredictable results. Given its limited models, the ASR can only choose the hypothesis that is most likely.

It is important to distinguish these kinds of errors from "bugs" or "exceptions" that need error handling (or "exception handling") in all computer systems. The source of such errors can, as soon as they are identified, be fixed (more or less easily). However, speech recognition errors cannot typically be "fixed" in a similar way. A distinction can be made here between *variable* and *constant* errors (Reason, 1990). The difference is metaphorically illustrated in Figure 3.1. Target A illustrates large variable errors, but small constant errors, that is, all shots are centred around the middle but with deviations that could be characterised as noise. There is no straightforward way of solving these errors; the sight seemed to be aligned as well as possible, but the rifleman needs more training. Target B, on the other hand, shows small variable errors, but a large constant error. Once the problem is identified (probably a misaligned sight), the error may be fixed. This doesn't mean that constant errors always give rise to similar behaviours that are easy to discover. For example, if a computer always relied on the same sorting algorithm that always failed to consider the two last elements, this would give rise to a large number of different error forms. Nevertheless, it would be a constant error that could easily be remedied as soon as it was found.

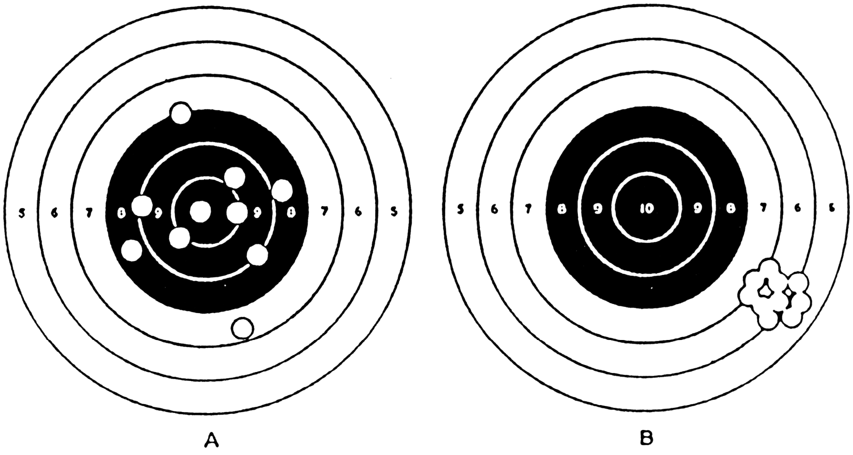

Figure 3.1: Two different target patterns. A exemplifies variable errors and B constant errors. (from Reason (1990), originally from Chapanis (1951)).

The acoustic and language models in the speech recogniser may be improved as more data is collected, and the variable error may be reduced, however probably never completely eliminated, at least not if other knowledge sources are not added to the process.

It should be noted that speech recognition may exhibit constant errors as well, that may be easily fixed once they are found. For example, a word may be incorrectly transcribed in the dictionary.

Another task that is commonly assigned to the ASR is voice activity detection (VAD). This may also be a significant source of errors, for example if the system incorrectly determines that the user has finished his turn, prepares what to say next and then starts to speak at the same time the user completes his turn.

There are of course sources of errors other than the ASR, such as NLU and dialogue management. However, the input to these processes is typically constrained by the language models used in the ASR and therefore exhibits less variability. The main challenge for these components is *error awareness* and *robust processing*, that is, to expect errors in the input and be able to do as much processing as possible despite these errors, with a performance that degrades gracefully. This leads to an error definition problem: given a partly erroneous result from the ASR, what is the expected output from these post-processes? Ideally, we would want such a process to repair the errors made by the ASR and return a result that fits the intentions of the speaker, in other words, to recover deletions and ignore insertions. However, if the number of errors is very large, this may be an unrealistic expectation. Again, it may be useful to compare with what a human could make of the task.

Given a correct result from the ASR, other processes may still make errors. Variable errors may arise in the NLU due to lexical and syntactic ambiguity and in the dialogue manager due to ambiguous elliptical and anaphoric expressions. This may lead to errors at the different levels of action discussed previously. The output processes in the dialogue system may also make errors, for example by using ambiguous referring expressions, so that the user misunderstands the system.

3.3 Error handling in spoken dialogue systems

Variable errors, due to limitations in the system's models, are inevitable in a spoken dialogue system. Even as the coverage of these models is improved, speakers (and developers of dialogue systems) will try to make the interaction more efficient by taking risks and introducing more ambiguity and uncertainty, at least in a conversational dialogue system. That said, there are ways to prevent, detect and repair errors, or minimise their negative consequences. Errors introduced in one process should not make further processing impossible – the processes should be *robust*. But errors introduced in one process may also be repaired in other processes, so that the output of the system as a whole meets the expectations. How is this possible? If we know how to repair an error in another process, why cannot the error be repaired or avoided in the process where it is introduced? There are three answers to this question. First, another process may utilise different knowledge sources which are not available in the first process. For example, the dialogue manager may have access to the dialogue history and domain knowledge which the speech recogniser doesn't have. This is true as long as we do not know how to integrate all processes into one process. Second, if we view the system and user as a joint unit, the user may be involved in the error handling process by grounding. A third, and more practical, answer is that a dialogue system developer working with a set of processes may not have knowledge or access to make the necessary modifications to fix even constant errors in the process in which they are introduced.

Error handling in a spoken dialogue system should not be seen as a single process in the system, but rather as a set of *issues* that should be regarded in all processes. The following human-computer dialogue example illustrates some error handling issues:

(30) U.1: I can see a brown building.
I CAN SEE A BLUE BUILDING
S.2: *A blue building, ok, can you see something else?*
U.3: No, a brown building.
NO A BROWN BUILDING

In this dialogue fragment, we can identify three main error handling issues which are related to the three turns. First, utterance U.1 will be recognised and interpreted, and the example illustrates an ASR substitution. If we consider the ASR to be the main source of errors, we would like to have some sort of technique for detecting potential errors in the ASR output, or in a robust interpretation of the ASR output. We call this *early error detection*. This could result in the system accepting (parts of) the hypothesis of what the user has said or rejecting it. But it could also result in an uncertainty of whether the hypothesis is correct or not. Just as humans do when faced with such uncertainty, the system may initiate a *grounding* process, which is done in S.2. In this example, the system is uncertain about the colour of the building and therefore displays its understanding (“a blue building”), as part of the next turn. This makes it possible for the user to identify the error and repair it (U.3). From the system’s perspective, it must now identify and repair this error based on its understanding of U.3. Since the error was already made in U.1, but detected after U.3, we call this *late error detection*.

In the rest of this chapter, the problems involved and the research done on managing these issues are laid out.

3.3.1 Early error detection

The first important error handling issue to consider is how errors introduced in the recognition and interpretation of the user’s utterance may be detected. If the recognition is poor, the ASR may give no hypothesis at all, which will inevitably result in a non-understanding. However, it is more common that the ASR will produce a result containing errors. The system must then understand which parts are incorrect and *decide* that it should be considered a (partial) non-understanding. In other words, the system must be able to *understand that it does not understand*. If this early error detection fails, it will result in a misunderstanding (which may perhaps be identified later on in late error detection).

Early error detection can be described as the task of deciding which ASR results, which words in the ASR results, or which semantic concepts in the interpretation should be considered as being correct (i.e., binary decisions), but it could also result in a set of continuous confidence scores, so that other processes may take other issues into account when making the decision. Early error detection is sometimes referred to as *recognition performance prediction* (Litman et al., 2000; Gabsdil & Lemon, 2004) or *confidence annotation* (Bohus & Rudnicky, 2002).

Most error detection techniques rely (partly) on the ASR confidence score, and we will start with a brief review of how this score is typically estimated.

3.3.1.1 ASR confidence score estimation

An ASR may be able estimate a confidence score for the whole utterance, but also for the individual words in it. The score is typically a continuous value between 0 and 1, where 0 means low confidence and 1 high confidence. If the score is only to be used for discriminating between the labels incorrect and correct – by setting a threshold for reject/accept – the only important factor for the quality of the score is how accurate such a classification can be made based on it. The standard metric used to asses the quality of a confidence scoring is the normalised cross entropy (NCE), which is an information theoretic measure of how much additional information the scores provide over the majority class baseline (i.e., assigning all words with the same (optimal) score). However, for other purposes, it could also be desirable to have a *probabilistic* score, that is, a confidence score of 0.3 would mean that there is a 30% probability that the hypothesis is correct.

According to Jiang (2005), methods for computing confidence scores in speech recognition can be roughly classified into three major categories: *predictor features*, *posterior probability* and *utterance verification*. The first approach is to collect predictor features from the recognition process, such as the n-best list, acoustic stability and language models, and then combine these in a certain way to generate a single score to indicate correctness of the recognition decision (see for example Hazen et al., 2002).

The second approach is to use the posterior probability (equation (10) on page 20) directly, which would constitute a probabilistic confidence score. However, there is a fundamental problem with this (Wessel et al., 2001). As shown in equation (11) and equation (12), the probability of the acoustic observation, *P(O),* is typically excluded from the model, since it is not needed to calculate the *relative likeliness* and choose the *most likely* hypothesis. Thus, the remaining formula, *P(O|W)P(W)*, does not describe the *absolute probability* of the hypothesis. It does not account for the fact that as the probability of the acoustic observation increases, it becomes more likely that the hypothesis is generated by something else, and the probability of the hypothesis should decrease. Methods for approximating *P(O)* have been proposed, such as using a phoneme recogniser, filler models, or deducing it from the word graph (Wessel et al., 2001).

In the third approach, utterance verification, confidence scoring is formulated as statistical hypothesis testing similar to speaker verification, using likelihood ratio testing,

3.3.1.2 Rejection threshold optimisation

Early error detection can, in the simplest case, be regarded as a choice between *reject* and *accept* by comparing the ASR confidence score against a *threshold*. If the score is above this threshold, the hypothesis is accepted, otherwise it is rejected. This threshold may be set to some default value (such as 0.3), however the performance can typically be optimised if data is collected for the specific application and analysed. Such an optimisation is shown in Figure 3.2. The lower the threshold, the greater the of number of false acceptances (i.e., over-generation). As the threshold is increased, false acceptances will be fewer, but more false rejections (i.e., under-generation) will occur. Such a graph may be used to find the optimal threshold with the lowest total number of false acceptances and false rejections (approximately 0.42 in the example).

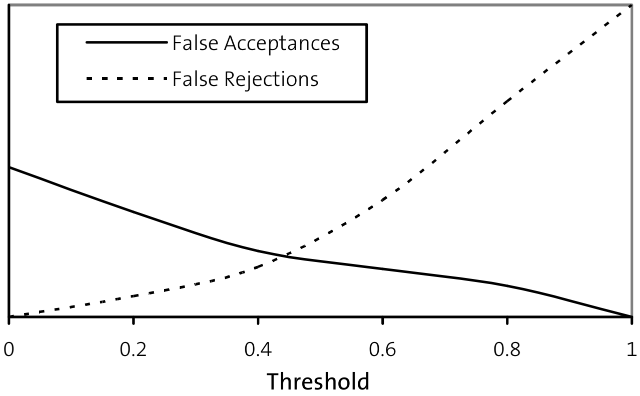

Figure 3.2: Rejection threshold optimisation.

One should bear in mind that this is only true as long as false acceptances and false rejections have the same cost – an assumption that will be questioned later on.

3.3.1.3 Other knowledge sources

To improve early error detection, machine learning has been used in many studies. A corpus of recognised utterances from the application is typically collected and annotated, and supervised learning is used to classify hypotheses as correct or incorrect, based on features from other sources than the ASR. A simple heuristic (such as accepting all hypotheses) is often used as a baseline to compare with.

An obvious argument against early error detection as a post-processing step on the ASR output is that the problems that these techniques attempt to fix should be addressed directly in the ASR. However, as argued in Ringger & Allen (1997), post-processing may consider constant errors in the language and acoustic models, which arise from mismatched training and usage conditions. It is not always easy to find and correct the actual problems in the models and a post-processing algorithm may help to pinpoint them. Post-processing may also include factors that were not considered by the speech recogniser, such as prosody, semantics and dialogue history.

Prosody is a strong candidate feature for early error detection, since people tend to hyperarticulate when they are correcting the system, which often leads to poor speech recognition performance (Oviatt et al., 1996; Levow, 1998; Bell & Gustafson, 1999). Speech recognition can also be sensitive to speaker-specific characteristics (such as gender and age), which may be reflected in prosodic features. Litman et al. (2000) examine the use of prosodic features for early error detection, namely maximum and minimum F_0 and RMS values, the total duration of the utterance, the length of the pause preceding the turn, the speaking rate and the amount of silence within the turn. A machine-learning algorithm called RIPPER (Cohen,

1995) was used. The task was to decide if a given ASR result had a word error rate (WER) greater than zero or not. Using only the ASR confidence score gave a better result than the baseline (guessing that all results were correct). However, adding the prosodic features increased the accuracy significantly. The accuracy was increased further by adding contextual features, such as information about which grammar was used in the recognition.

Other knowledge sources, not considered by the ASR, which should improve error detection are features from the NLU and dialogue manager. In Walker et al. (2000b), the usefulness of such features is studied, using data from the "How May I Help You" call centre-application. 43 different features were used, all taken from the log, which means that they could have been extracted online. The NLU and dialogue manager related features included parsing confidence, grammar coverage, and preceding system prompt. The RIPPER algorithm was used in this study also, but the task was in this case to decide if the semantic label assigned to the utterance was correct or not (i.e., early error detection was performed after interpretation). Again, using the ASR confidence score alone was better than baseline, but adding the other features improved the performance significantly.

The methods discussed above (except the raw ASR confidence score) are all based on binary decisions between correct/incorrect. This is useful if the only choice is between rejecting and accepting the hypothesis, but if other factors are to be taken into account or other options are to be considered (as will be discussed later on), a continuous (possibly probabilistic) confidence score would be more useful as a result of the early error detection. Bohus & Rudnicky (2002) investigated the use of different machine learning approaches to confidence estimation based on a number of features from the ASR, the NLU and the dialogue manager, and found that logistic regression gave the best result.

The common approach to early error detection, as the review above indicates, is to train the classifier on an annotated pre-recorded corpus. Bohus & Rudnicky (2007) present an alternative approach, where the system collects online data from clarification requests. The user's response to a clarification request indicates whether the hypothesis was correct or not. This way, training material may be collected without having a human annotating it. Thus, the system can be said to learn by its own experience. The data collected will contain more noise than manually annotated data, since users do not always act as intended after clarification requests, and their responses sometimes are misrecognised by the system. However, the study shows that the achieved confidence estimation performance is nearly (but not quite) as good as the one that is achieved with manual annotation.

In the studies presented above, whole utterances are considered. This may be useful for shorter utterances with more simple semantics. However, if utterances are longer and contain more complex semantics, it may be useful to consider individual words or concepts for early error detection. In Chapter 5, such a study is presented.

3.3.1.4 Error correction and n-best lists

Another possibility in the post-processing of the ASR result is to not only *detect* errors, but to also *correct* them, in other words not just delete insertions, but also re-insert deletions. An obvious source for such corrections is the n-best list or word lattice typically provided by the ASR, as described in 2.3.1.

In order to explore the upper limit of such an approach, Brill et al. (1998) conducted an experiment in which subjects were given the task of choosing the most likely hypothesis from 10-best lists, but were also asked to manually edit and correct the best hypothesis to improve it further if they thought it was possible. The data was from switchboard, broadcast news and Wall-street journal. The results showed that the subjects were able to improve WER by 1.3-3.1 percent units by just selecting the best hypothesis, and by 2-4 percent units by further corrections.

Examples of studies on automatic reordering of n-best lists include Rayner et al. (1994), Chotimongkol & Rudnicky (2001), Gabsdil & Lemon (2004) and Jonson (2006), which all show how the system sometimes can choose better hypotheses if knowledge sources other than those used by the ASR are considered. In short, n-best list reordering is often done by applying some sort of early error detection technique (as discussed above) to several hypotheses and then picking the one that achieves the best score.

There are some potential problems involved in processing n-best lists. First, it may be computationally challenging to consider many possible alternatives simultaneously, especially if other knowledge sources in the form of other dialogue system components, such as parsing, are to be involved. This is especially true if the dialogue system should operate incrementally, on a word by word basis. Second, as discussed in 2.3.1, n-best lists may be less fruitful to consider when utterances are longer, since many of the top hypotheses will be very similar, with perhaps just some single function words varying in a long list of semantically similar combinations. Thus, n-best lists may be more useful in command-based dialogue systems where utterances may be shorter and incrementality is not an issue.

For conversational dialogue systems, it may be more useful to explore the use of word lattices. However, this may require a more sophisticated approach than just applying an early error detection technique on a list of hypotheses. That is beyond the scope of this thesis.

3.3.1.5 Error prediction

One interesting error handling strategy is to detect the problem before it has even occurred – in other words, to *predict* errors. This way, the dialogue system could adapt its behaviour to avoid the problem. Walker et al. (2000c) report an experiment where the initial segments of a dialogue were used for error prediction. The dialogue system was the "How May I Help You" call centre-application. All dialogues were classified as "task success" or "problematic". The RIPPER machine-learning algorithm (Cohen, 1995) was trained to classify the dialogues based on online features from the ASR, NLU and discourse. By just looking at the first turn, the performance (72.3%) was significantly better than majority class baseline (64%, tagging everything as "task success"), although the improvement is not huge. By also looking at the second turn, the improvement was better (79.8%). Although 16% above baseline, just given the two first exchanges and online features, sounds impressive for predicting problematic dialogues, the question is whether 80% is good enough to be able to take appropriate actions.

3.3.2 Grounding in dialogue systems

In the simplest case, error detection leads to the choice of reject or accept. However, there are other alternatives. As discussed in 3.1.2 above, humans detect and correct errors together by giving evidence of understanding in a grounding process. If the system is uncertain about whether an error is present, it may provide evidence of understanding and detect potential errors based on the user's reaction to this evidence.

3.3.2.1 Explicit and implicit verification

The most well-known and well-tested techniques for grounding are called *explicit* and *implicit verification*. In explicit verification, the system asks a clarification request, typically in the form of a reprise sentence (see Table 3.2 on page 39). The following example is from Bouwman et al. (1999):

(31) U: I'd like to travel to Amsterdam.
S: *Do you want to go to Amsterdam?*
U: Yes. that's right.

In implicit verification, the system instead displays its understanding as in the following example (again from Bouwman et al. (1999)):

(32) U: I'd like to travel to Amsterdam.
S: *To Amsterdam. From where do you want to leave?*

Whereas explicit verification typically requires the user to confirm the hypothesis for the system to consider it as being correct, implicit verification require no such response. As long as the user does not object, the hypothesis is considered to be correct.

3.3.2.2 Other kinds of evidence

As we saw in 3.1.2, clarification in the form of reprise sentence (explicit verification) and display of understanding (implicit verification) are just two kinds of evidence that speakers use in dialogue. Are there other kinds of evidence that may be useful for error handling in spoken dialogue systems? Continued attention and relevant next contribution should perhaps not be regarded as a choice the system can make in order to give positive evidence. Instead, if the system continues with an irrelevant next contribution (due to a misunderstanding), this will probably trigger the user to initiate a repair. However, the system should be capable of understanding such a repair.

Acknowledgements may be used to actively give positive evidence. An acknowledgement like "ok" or "m" may give the user evidence that the system has heard the utterance and processed it on some level of action. However, it cannot be used to actually detect erroneous hypotheses, since the system's understanding of the involved concepts is never signalled to the user.

3.3.2.3 Evidence on different levels of action

As discussed previously, clarification requests may concern different levels of action. In explicit verification, it is normally assumed that the perception level is concerned. However, other levels may be concerned as well, as the following example illustrates:

(33) U: I have the red building on my left.
S(alt. a): *Which red building?*
S(alt. b): *Do you really mean the red building?*

In these examples, the user gets positive evidence that the system has heard and to some extent understood what the user said, but there is some problem in deeper understanding. Such clarification requests could be useful for resolving ambiguous anaphoric expressions (as in the example above), but also for ambiguous fragmentary expressions. The use of clarification on different levels of action is explored in Schlangen (2004) and Rieser (2004).

Larsson (2003) discusses the use of positive and negative evidence on different levels of action. As positive evidence, display of understanding and acknowledgements are considered. The examples on negative evidence are mostly (implicit) requests for repetition:

- Contact: "I didn't hear anything from you"
- Perception: "I didn't hear what you said"
- Semantic understanding: "I don't understand"
- Pragmatic understanding: "I don't quite understand"
- Acceptance: "Sorry, Paris is not a valid destination city"

In 3.1.1, we questioned whether failure on the acceptance level really should be classified as miscommunication. In the same way, we may question whether "evidence" on the acceptance level, as in the example above, really should be classified as evidence of understanding in the same way as the other levels – it is not caused by any uncertainty or lack of hypotheses. Thus, it should perhaps not be considered as being part of error handling.

3.3.2.4 Non-understanding recovery

Non-understandings may not only be frustrating for the user per se, they may also lead to error-spirals, that is, further non-understandings that may be hard to recover from. For example, Levow (1998) found that the probability of experiencing a recognition error after a correct recognition in a dialogue system was 0.16, but immediately after an incorrect recognition it was 0.44. Thus, if the system decides to reject the user's last utterance it should take appropriate actions to recover understanding in subsequent turns.

One reason for non-understandings often leading to other non-understandings is that speakers usually repeat the non-understood utterance in the subsequent turn. If the ASR failed to recognise this utterance the first time (possibly because the utterance was out-of-vocabulary or that the user's pronunciation of the utterance is uncommon), there is an increased risk that it will fail the second time too. Another reason is that people tend to hyperarticulate when making repetitions after non-understanding (Oviatt et al., 1996; Levow, 1998; Bell & Gustafson, 1999), a strategy that is useful when speaking with humans, but may worsen the performance of the ASR. Many speech recognisers lack models for hyperarticulate speech, which makes the understanding of repeated utterances even more difficult.

One approach to this problem is to look at how speech recognition can be improved after non-understanding. For example, Ainsworth & Pratt (1992) investigates how the system can eliminate the misrecognised word from the vocabulary to improve recognition of repetitions.

Another approach is to design the system response after the non-understanding more carefully. A common assumption seems to be that after non-understanding, the system has no option but to request repetition by signalling non-understanding or making a clarification request, just as in the examples on negative evidence on different levels of action listed above. While this mapping between action levels and system responses seems straightforward and intuitive, the usefulness of such signals of non-understanding for handling errors can be questioned, since they encourage repetitions. Neither is it usually fruitful to try to explain or analyse the source of the problem. For example, to use an utterance like “I didn’t hear what you said” to signal that the problem is due to the ASR (and not some other processing step), will probably just encourage hyperarticulated repetitions. As Balentine et al. (2001) writes in a style guide for telephony dialogue systems:

> Avoid apologizing for problems or inadvertently blaming the user for them. Instead, simply move forward by prompting for the next appropriate user action. There are two motivations for this. First, the application can never be certain of the underlying error, so descriptions of the problem may be incorrect or misleading. Second, explaining the problem does not necessarily influence the user in a constructive way. Rather than dwelling on the condition that has led to a problem, it is better to describe what action is now expected from the user. (p. 55)

According to Balentine et al. (2001), system responses after non-understandings should encourage the user to try another wording, and provide incrementally more help on subsequent non-understandings.

The problem of non-understanding recovery is explored and discussed in more depth in Chapter 4.

3.3.2.5 Concept-level grounding

The overview of research on clarification requests in 3.1.4 above showed that humans often (in about 45% of the cases) use fragmentary constructions when making clarification requests. To improve efficiency and naturalness, a dialogue system should also be able to utilise fragmentary utterances in grounding. Fragmentary grounding utterances may not only be realised more efficiently, they may also help to pinpoint the problematic parts of the original utterance. However, as Gabsdil (2003) points out, the use of fragmentary grounding in spoken dialogue systems is not very common. The following example illustrates a possible use:

(34) U.1: I have a red building on my left.
S.2 (alt. a): *Red?*
S.2 (alt. b): *Red, ok, what do you have on your right?*

S.2(b) may look similar to the implicit verification "To Amsterdam" in example (29) above. However, there is a difference. "To Amsterdam" is, in the travel booking domain, equivalent to "I want to go to Amsterdam" or "So you want to go to Amsterdam". It does not need to be resolved. The utterance "red" on the other hand could mean many different things in the navigation domain that the example is taken from, and necessarily has to be resolved and placed in a larger semantic construct. Thus, whereas "To Amsterdam" does not help to pinpoint the problematic part of the utterance, the utterance "Red" does.

The use of fragmentary grounding utterances has some interesting challenges that are addressed in Chapter 6 and 9:

- The problematic concepts in the original utterance must be identified.
- The grounding utterance must have the right textual and prosodic realisation to be understood correctly by the user.
- The system must remember for which concepts it has provided evidence.
- The user's reaction to the request must be understood correctly. If the user negates and/or corrects the proposed concept, the system must understand that it is only parts of the original utterance that have been negated and/or corrected, not the entire contribution.

Rieser (2004) and Schlangen (2004) describe implementations of systems that are capable of posing fragmentary clarification requests based on concept confidence scores on all action levels. However, the models do not handle the user's reactions to those requests.

The use of concept-level clarification requests in dialogue systems has received more interest than the use of concept-level display of understanding. Concepts may be displayed as separate communicative acts, as in S.2(b) in example (34) above. But, as discussed in 3.1.2.1, the display of concepts may also be integrated in the next communicative act, with the primary communicative function relating to the task at hand. The following examples with alternative system reactions are examples of this:

(35) U.1: I have a red building on my left.
S.2(alt. a): *How many stories does <u>it</u> have?*
S.2(alt. b): *How many stories does <u>the building</u> have?*
S.2(alt. c): *How many stories does <u>the red building</u> have?*

By choosing between different referring expressions, the system may display its understanding to different extents, depending on its confidence in the concepts involved. To be able to do this, the challenges listed above must be considered. The issue of choosing between these different realisations and modelling an integrated display of understanding will be addressed in Chapter 6.

3.3.2.6 Alternative clarification

There is another type of clarification request that may be referred to as an *alternative* clarification request (Gabsdil, 2003). If the system can consider several alternative hypotheses from the speech recogniser, it could make a clarification request such as this:

(36) U: I have a red building on my left.
S: *Red or blue?*

It could also be possible to make an alternative clarification request on the understanding level, for example if there are several possible ways to resolve an anaphora:

(37) U: I have the building on my left.
S: *The red or the blue one?*

In order to pose alternative clarification requests, the system must somehow be able to produce several parallel hypotheses, for example by n-best lists or word lattices. See 3.3.1.4 for a discussion on the potential problems associated with this. The use of alternative clarification will not be investigated in this thesis.

3.3.2.7 Making grounding decisions

As we have seen, there are several ways to handle uncertainty and errors in dialogue, either towards risking under-generation or over-generation. As Allen et al. (1996) points out, sometimes it may be better to "choose a specific interpretation and run the risk of making a mistake as opposed to generating a clarification subdialogue". The system may display its understanding, request clarification on what is not understood, presuppose understanding and defer the detection of errors to a later stage in the dialogue, or simply reject the hypothesis. We refer to this choice as the *grounding decision problem*. The grounding decision concerns not only which evidence of understanding to give, but also whether the hypothesis should be regarded as common ground. A common basic approach is to use hand-crafted confidence thresholds as shown in Figure 3.3 (see for example Bouwman et al., 1999).

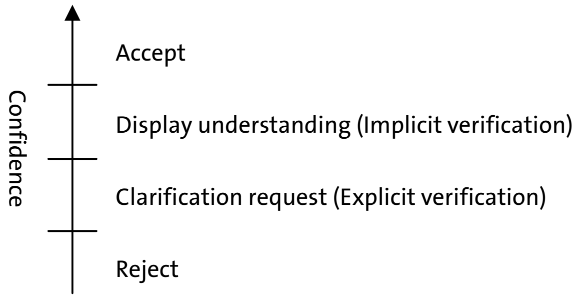
Confidence
Accept
Display understanding (Implicit verification)
Clarification request (Explicit verification)
Reject

Figure 3.3: Typical grounding decision based on a confidence score.

This division seems intuitive, but the problem is how to find optimal thresholds. Another problem is that the thresholds most often are static and not dependent on the dialogue context. Simple threshold optimisation as described in 3.3.1.2 above cannot be applied to this problem.

We will return to this issue in Chapter 8, where a dynamic, data-driven, decision-theoretic model for making grounding decisions is presented and related research is discussed.

3.3.2.8 Evidence of understanding from the user

Most studies on grounding in spoken dialogue systems, including this thesis, are focussed on how to cope with system non-understandings and misunderstanding, in particular those caused by speech recognition errors. As a consequence, the models of grounding proposed are mainly concerned with how the system should provide evidence of understanding based on its hypotheses of the user's utterances. However, it is of course also possible that users may give evidence of understanding. A dialogue system should for example be able to make a repetition after an utterance such as "what did you say?". But as dialogue systems start to utilise more sophisticated methods for providing evidence, we should expect users to do the same. For example, if we endow systems with the capabilities of making fragmentary clarification requests such as "red?" and non-verbal acknowledgements such as "uhu", users are likely to pick up on this. This poses new challenges for the recognition and interpretation of user utterances, including prosodic analysis.

3.3.3 Late error detection and misunderstanding repair

If the system accepts an incorrect hypothesis of the user's communicative acts, it may still be possible to do late error detection at later stages in the dialogue and repair the misunderstanding.

3.3.3.1 Late error detection

Late error detection may typically be performed after the system has displayed its understanding (based on an incorrect hypothesis) and the user initiates a repair:

(38) U.1: I have a blue building on my left.
S.2: *How many stories does the brown building have?*
U.3: I said blue building!

As noted previously, errors may also be detected based on inconsistencies in general, several turns after the actual error occurs. This leads to two issues that should be handled by a dialogue system. First, to facilitate late error detection, the system must be capable of detecting cues from the user in the "third turn" that something is wrong (such as the U.3 in the example above), and to detect inconsistencies in general. Second, to enable repair of misunderstandings, it must know which assumptions to remove or re-evaluate in its model of the common ground.

One problem with detecting errors in the "third turn" is that these problem signals may look very disparate and may depend on subtle prosodic cues, or the user may just ignore the problem. Here are some imaginable variants to U.3 in example (38) above:

(39) U.3a: eeh, brown building?
U.3b: I can't see any brown building.
U.3c: I don't understand.
U.3d: the brown building we talked about before?
U.3e: ehh... I can see a blue building.

Krahmer et al. (2001) calls these signals "go back" cues, as opposed to "go on" cues, which signal that the displayed hypothesis was correct. A number of possible cues are listed in Table 3.6.

Table 3.6: Possible positive and negative cues from the user after the system has displayed its understanding (from Krahmer et al., 2001).

Positive cue	**Negative cue**
Short turns	Long turns
Unmarked word order ("I want to leave from Stockholm")	Marked word order ("It is Stockholm I want to leave from")
Confirm ("Yes")	Disconfirm ("No")
Answer	No answer
No corrections	Corrections
No repetitions	Repetitions
New info	No new info

In an analysis of a hand-labelled corpus based on spoken dialogue systems providing train timetable information, Krahmer et al. (2001) found that users never gave explicit positive cues such as "yes", and rather seldom (in 15.4% of the cases) gave explicit negative cues ("no"), after an implicit verifications (display of understanding). The best determinant is instead whether the user makes any attempts to make a correction or not. It should be noted that this analysis has been performed on what the users actually said, not on the results from the ASR. The question is to what extent these results can be applied to online dialogue, since the ASR for example may miss corrected slots. It should also be noted that given a negative cue, the system has no information on what the problem actually is, just that a problem has occurred.

One approach to this problem is to use machine learning, training on a set of features, similar to early error detection, to distinguish user corrections from non-corrections. Litman et al. (2006) investigate the use of prosodic, ASR-derived, and system-specific features, both for the current turn and for contextual windows, and using summary features of the prior dialogue. An initial analysis showed that the prosody of corrections differ significantly from non-corrections, being higher in pitch, louder, longer, with longer pauses preceding them and less internal silence. As expected, they are misrecognised more frequently than non-corrections. Using machine learning, the best-performing feature set cuts the majority baseline error almost in half, from 29% to 15.7%.

The problem of detecting negative evidence in the third turn has lead many dialogue designers to avoid using display of understanding (implicit verification). A common experience is that users often don't know how to react and feel uncomfortable (see for example Weegels, 2000).

3.3.3.2 Modelling grounding status

As positive or negative evidence is given for a hypothesis and the user reacts to this evidence, we may say that the hypothesis' *grounding status* has been updated. By modelling this grounding status, the system may decide when the grounding criterion has been satisfied, that is, when the hypothesis may be considered to be common ground.

Traum (1994) shows how recursive transition networks (RTN) may be used to track the grounding status. In this account, the semantic units that get grounded are called *discourse units*, and the actions that contribute to the updating of the grounding status of these are called *grounding acts*. Grounding acts may be repairs, requests for repairs, acknowledgements and requests for acknowledgement. The discourse units can be compared to speech acts. Thus, the model can be said to track utterance-level grounding. Grounding actions are also treated as a special kind of communicative act. These two properties are common for most models of grounding status. The problem with such accounts is that they do not explain how for example different kinds of referring expressions in task-related utterances may help to provide evidence of understanding on parts of the previous utterance, as in example (35) above. A third property of many models of grounding status is that the grounding status is only tracked locally within the "subdialogue". This makes it impossible for the system to consider the grounding status later on in the dialogue if inconsistencies that indicate a misunderstanding are found. In Chapter 6, a model that deals with these shortcomings is presented.

Heisterkamp & McGlashan (1996) presents a model in which the grounding status is tracked by assigning a *contextual function* to each information unit.

- new_for_system(X).
- repeated_by_user(X).
- inferred_by_system(X).
- modified_by_user(X).
- negated_by_user(X).

A similar model is used by McTear et al. (2005) under the name of *discourse pegs*.

Another approach to late error detection and grounding status modelling is presented in Bohus & Rudnicky (2005a), where the system models its belief in concepts as a continuous confidence score that gets updated as the concept is grounded. The approach is called *belief updating* and is defined as follows:

> given an initial belief over a concept $Belief_t(C)$, a system action *SA(C)* and a user response *R*, compute the updated belief $Belief_{t+1}(C)$.

Or, in terms of confidence score for a single hypothesis:

> given an initial confidence score for the top hypothesis *h* for a concept *C*, construct an updated confidence score for the hypothesis *h*, in light of the system confirmation action *SA(C),* and the follow-up user response *R*.

Thus, the approach is similar to that of using machine learning for confidence estimation in early error detection, but it is extended to also include features from the subsequent CA's in the grounding process. In their approach, data was used to train a binary logistic regression model on features from the original hypothesis as well as the system's action and the follow-up user response. Some of the most useful features were: the initial confidence score, prosodic features, expectation match, barge-in, lexical features, the presence of repeated grammar slots, as well as the identity of the concept to be confirmed.

3.3.3.3 Misunderstanding repair

When a misunderstanding is detected, it should be repaired. To do this, the system must have some mechanisms for removing erroneous hypotheses from the common ground. For example, in Larsson (2002), a "backup" copy of the dialogue state (a "temporary storage") is kept to restore the information state if the system's hypothesis of the common ground turns out to be incorrect. A drawback with this approach is that the detection of the misunderstanding may only occur immediately after the error, in the "third turn", and that the dialogue state is completely restored, which means that individual concepts that were not erroneous are lost.

It is also possible that the system should not simply restore the state when an old error is suspected. The system could also make a late clarification request, as in example (26) on page 38.

A problem with implementing a more elaborate model of late error detection in many dialogue systems is that the result of early error detection (such as confidence scores) are most often only considered once and not stored for late error detection. In Chapter 6, this issue is discussed in more depth, and a model that supports long-term storage of confidence scores and grounding information is presented.

3.4 Summary

This chapter has reviewed the work done on error handling in spoken dialogue systems and laid-out the issues that are involved. Figure 3.4 shows a diagram which illustrates how the most important error handling issues presented above are connected, from the perspective of this thesis.

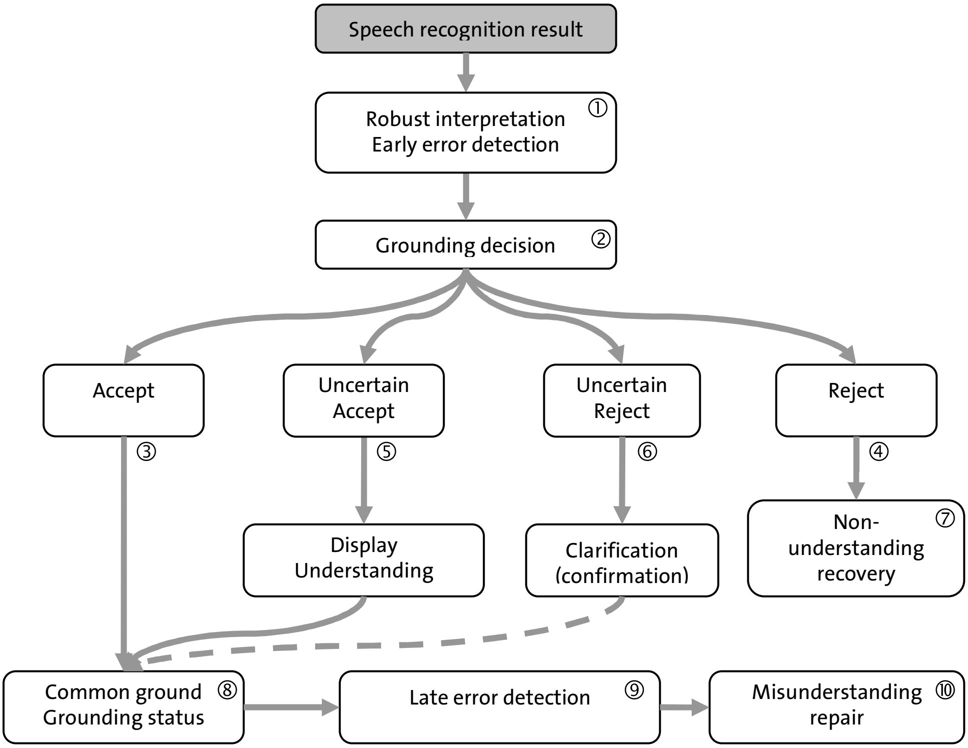

Figure 3.4: Overview of the error handling issues considered in this thesis

The diagram can be described as follows. Given a noisy speech recognition result, the system must make a *robust interpretation* of it into a hypothesis of the semantic concepts in the user's utterance ①. The system must also decide which words in the ASR output or which resulting semantic concepts should be considered to be correct, and/or decide the level of uncertainty (*early error detection*) ①. For each concept in the hypothesis, the system must make a *grounding decision* ②:

- The system could simply *accept* the concept and regard it as common ground ③.
- The system could simply *reject* the concept, i.e., treat it as a *non-understanding* ④.
- The system could accept the concept and add it to the common ground but at the same time *display its understanding*, so that the user has a chance to correct the system if the concept is incorrect ⑤.
- The system could reject the concept (i.e., not treat it as common ground), but make a *clarification request*, so that the user may confirm the concept if it is correct. If so, the concept may be treated as common ground ⑥.

If the system rejects the concept, it might perform a *non-understanding recovery* ⑦, which may be some sort of clarification request. If a concept is treated as common ground, the system's uncertainty of the concept should be stored ⑧, so that errors may be detected later on (*late error detection*) ⑨, for example if the system displays its understanding and the user objects. In such a case, a misunderstanding has been detected, which needs to be repaired ⑩. The concept should be removed from common ground, but the user may also be involved in further clarifications.