CHAPTER 1

Introduction

Most of today's computer applications have a graphical user interface (GUI), which allows the user to perform actions through direct manipulation of graphical elements, such as icons, text and tables, presented to the user on a display device. As an alternative (or complement) to this kind of interaction, a *spoken dialogue system* offers the possibility to interact by means of spoken language. Possible applications which may benefit from spoken language interfaces include flight booking over the telephone, human-robot interaction, speech controlled music players and conversational computer games. Compared to a GUI, a spoken language interface frees the user's hands and eyes for other tasks. Moreover, human-human conversation is generally a natural, intuitive, robust and efficient means for interaction. A lot of effort is being invested in trying to make spoken dialogue systems also benefit from these properties.

To be able to engage in conversation, a spoken dialogue system has to attend to, recognise and understand what the user is saying, interpret utterances in context, decide what to say next, as well as when and how to say it. To achieve this, a wide range of research areas and technologies must be involved, such as automatic speech recognition, natural language understanding, dialogue management, natural language generation and speech synthesis.

One of the greatest challenges when building dialogue systems is to deal with *uncertainty* and *errors*. Uncertainty comes partly from the ambiguity of natural language itself. In addition, in the case of spoken dialogue systems, a great deal of uncertainty comes from the error-prone speech recognition process. Speech recognition errors arise from speaker variability, background noise and unexpected language use, which are all hard (if not impossible) to model exhaustively. 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 and context-dependence of language use; the same information may be conveyed in different ways in the same utterance, or

may be repeated by the speakers in order to ensure understanding, and the context may be used to reduce uncertainty or fill in the gaps. Furthermore, when humans speak to each other, there is a collaborative process of avoiding and recovering from miscommunication that often goes unnoticed (Clark, 1996).

The topic of this thesis is how to draw lessons from this seemingly smooth handling of uncertainty and miscommunication in human-human dialogue, and how to use this knowledge to improve error handling in spoken dialogue systems.

1.1 Error handling issues

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*. Errors in spoken dialogue systems may be classified into two broad categories: under-generation and over-generation of interpretation hypotheses. In terms of miscommunication, these categories correspond to the notions of *non-understanding* and *misunderstanding*. *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.

As is argued in this thesis, error handling is not a separate processing step in a spoken dialogue system, like a speech recogniser or a dialogue manager. Instead, error handling should be regarded as a set of issues that should be considered in all parts of the system to handle the consequences of under-generation and over-generation. As an example, take a dialogue system which is supposed to understand the user's description of her location in a city. This might be the output of the speech recogniser[1]:

(1) User: **I CAN SEE A BLUE BUILDING**

The system must now consider the possibility that this is not what the user actually said – the speech recogniser might have over-generated or under-generated words. One important error handling issue is then to detect such errors in the speech recognition result, so that misunderstanding may be avoided. We will call this *early error detection*. If all words are deemed to be incorrect, or if the speech recogniser does not deliver any hypotheses at all, we may say that a non-understanding has occurred.

[1] The greyscale represent the speech recognition word confidence scores. Dark colour represents high confidence and light colour represents low confidence.

Another important issue is then how to *recover from non-understanding*. This means that the system must gain understanding of future contributions as efficiently as possible, after a non-understanding has occurred.

But, as Clark (1996) points out, the handling of uncertainty and problems in dialogue is not just an individual task. Speakers deal with uncertainty and problems together, by providing and evaluating positive and negative *evidence of understanding* – a process commonly referred to as *grounding*. By this process, the speakers share their private knowledge and beliefs with each other and make sure that what was private can now be regarded as *common ground*. The following example[2] illustrates some alternative responses from the system that all provide different kinds of evidence:

(2) U.1: **I CAN SEE A BLUE BUILDING**
S.2 (alt. a): *Ok, can you see a tree?*
S.2 (alt. b): *Blue?*
S.2 (alt. c): *How many stories does the blue building have?*

As can be seen in the example, the system may provide evidence of its understanding in different ways and for different parts of the original utterance. It may for example assume understanding and provide an acknowledgement (a), it may suspect that some part of the utterance is incorrect and pose a clarification request (b), or it may assume understanding but provide a verbatim repetition as part of the next contribution (c). A key problem for spoken dialogue systems is to choose *which* evidence to provide and *how* it should be realised. It would not be efficient to provide evidence on everything that is said; neither would it be efficient to not provide any evidence at all, since this would lead to a lot of misunderstandings that would need to be repaired later on.

If the system assumes understanding of an incorrect hypothesis, this misunderstanding may still be repaired. For example, the positive evidence that the system provides (such as alt. c above) might let the user detect such misunderstandings and raise an objection:

(3) U.1: **I CAN SEE A BLUE BUILDING**
S.2: *How many stories does the blue building have?*
U.3: **I SAID A BROWN BUILDING**

However, the system must be able to understand such objections and repair the error that was made. This will be referred to as *late error detection* (i.e., detection of misunderstandings) and *misunderstanding repair*. Another opportunity for late error detection is if the system detects inconsistencies in its model of what has been said and what it knows about the world.

[2] In all examples, U refers to the user and S to the system (A and B are used for human interlocutors). The number after the dot indicates the order of the turn. Italics are used to separate the speakers and enhance readability. In the case of human-computer dialogue examples, user utterances written in greyscale capital letters represent speech recognition results, not what the user actually said.

1.2 Thesis contributions

The work presented in this thesis consists of experiments on several of the issues outlined above and the development of models and methods for handling them. Of course, all aspects cannot be covered in the scope of this thesis. However, the ambition has been to not focus on only one particular aspect of error handling, but to study how error handling may be performed in different parts of a complete spoken dialogue system.

There are two general themes in this thesis. The first theme concerns how to explore and draw lessons from *human-like error handling strategies* and how to apply these to human-computer dialogue. The second theme, which is related to the first, is to explore *concept-level error handling.* Most approaches to error handling in spoken dialogue systems have focused on whole utterances as the smallest unit. Typically, whole utterances are assigned confidence scores and decisions are made whether to reject, verify or accept them as correct. Such utterance-level error handling is often feasible in command-based dialogue systems where utterances are relatively short and predictable. However, in conversational dialogue systems, utterance-level error handling is too simplistic. Humans engaging in conversation often focus on parts of utterances to, for example, pose fragmentary clarification requests (as exemplified in alt. b in example (2) above), and thereby increase the efficiency of the dialogue. In dialogue systems that are targeted towards more human-like conversation, speech recognition results and the semantic interpretations of them may often be partly correct. This calls for error handling on a "sub-utterance" level – to base error handling on individual words or concepts in utterances.

This thesis relies to a large extent on an *empirical approach*. In order to understand how humans manage problems in dialogue, such data must be collected and analysed. To understand what kind of problems arise in spoken dialogue systems, we also need data containing real speech recognition errors. The models and guidelines derived from the data have been used to develop a dialogue system that is in turn evaluated with naive users.

The main contributions of this thesis can be summarised as follows:

- An experimental setup that allows us to study the way humans deal with speech recognition errors.
    - Results indicating that humans to a large extent employ other strategies than encouraging the interlocutor to repeat when faced with non-understandings.
- Results confirming previous findings that errors in speech recognition results may be detected by considering confidence scores as well as other knowledge sources that were not accessible to the speech recogniser. While previous studies have shown this to be true for utterance-level error detection, these results show that it is also true for word-level error detection.
- A practical model for how the grounding status of concepts gets updated during the discourse in a spoken dialogue system, how this updating is affected by the use of

anaphora and ellipses, and how this information may be used for various error handling strategies.
    - An implementation of the model in a complete spoken dialogue system in a non-trivial domain.
    - An evaluation of the system with naive users, which indicates that the system and model performs well under error conditions.
- A decision-theoretic, data-driven model, based on task-analysis and bootstrapping, for making grounding decisions in spoken dialogue systems.
- A tentative model for the intonation of synthesised fragmentary grounding utterances in Swedish and their associated pragmatic meaning.

1.3 Thesis overview

In the rest of Part I, the brief initial overview of error handling given in 1.1 above is developed into a detailed discussion of the issues, and research on how they may be handled is reviewed. This will serve as a background for presenting the contributions of this thesis in Part II, III and IV. The thesis ends with a concluding summary and discussion in Part V.

Chapter 2: Spoken dialogue systems

This chapter discusses some basic properties of spoken dialogue and the techniques and issues involved in developing spoken dialogue systems. It is argued that two general types of dialogue systems may be distinguished, command-based and conversational, and that this thesis is targeted towards the latter, that is, dialogue systems that to a larger extent build upon the principles of human conversation. The basic building blocks of spoken dialogue systems are discussed: automatic speech recognition, natural language understanding, dialogue management, natural language generation and text-to-speech synthesis.

Chapter 3: Miscommunication and error handling

This chapter starts with a general discussion on the concepts of miscommunication, grounding, repair, clarification and error in the context of human-human and human-computer dialogue. This is followed by reviews and discussions on previous research related to error handling in spoken dialogue systems, including early error detection, grounding and late error detection. Places where the contributions of this thesis fit in and extend this body of knowledge will be pointed out.

Part II: Exploring Human Error Handling

Chapter 4: Exploring non-understanding recovery

In this chapter, a method for collecting data on human error handling strategies is presented. An experiment was conducted based on this method, in which pairs of subjects were given a

joint task for which they needed to engage in dialogue. A speech recogniser was used to introduce errors in one direction – thereby simulating the roles of human and computer in a dialogue system. The analysis is focussed on how the subjects recovered from non-understanding.

Chapter 5: Early error detection on word level

One interesting result from the experiment presented in Chapter 4 was that humans were extremely good at *early error detection*, that is, to understand which recognition hypotheses were incorrect in order to avoid misunderstandings. Most studies on automatic early error detection have focused on detecting whether a given hypothesis of a user utterance contains any errors at all or is error-free. For concept level error handling, it would be more useful to detect which words or concepts in the hypothesis that are erroneous, and it is obvious that this is what the human subjects did. This chapter explores which factors humans rely on when detecting such errors. A machine learning experiment is also presented where the data from Chapter 4 is used for automatic detection.

Part III: The Higgins Spoken Dialogue System

Chapter 6: Concept-level error handling in Higgins

As part of the work presented in this thesis, a complete spoken dialogue system, called HIGGINS, has been developed to serve as a test-bed for implementing and evaluating error handling methods and models. For this system, a set of modules have been developed, most notably a robust interpreter called PICKERING and a discourse modeller called GALATEA, which models the grounding status of concepts. This chapter describes how concept level error handling is done in the different parts of the system. In most previous accounts, a special set of grounding actions are used to provide evidence of understanding for complete user utterances. In this chapter, it is shown how all utterances instead may contribute to the grounding process on the concept level.

Chapter 7: Higgins evaluation

This chapter presents two separate evaluations. First the performance of the robust interpreter PICKERING is studied under different error conditions. Second, a data collection is presented in which naive users interact with the HIGGINS system. The data is used to evaluate the system in general and the discourse modeller GALATEA in particular, as well as the users' reactions to fragmentary clarification.

Part IV: Deciding and Realising Grounding Actions

Chapter 8: Making grounding decisions

The different grounding strategies supported by HIGGINS leave the system with decisions that have to be made: what kind of evidence should it provide and which recognition hypotheses

should it accept as correct? These grounding decisions should ideally be based on the system's uncertainty in its hypothesis, the cost of misunderstanding, and the cost of making a grounding action. In this chapter, the framework of decision making under uncertainty is applied to the problem, where the utility and costs of different actions are weighted against the probabilities of different outcomes. The data collected in Chapter 7 is used to derive a data-driven, dynamic model for making these decisions.

Chapter 9: Prosody in fragmentary grounding

Since fragmentary grounding utterances (such as alt. b in example (2) above) lack syntactic and lexical information, their interpretation depends to a large extent on their prosodic realisation. This chapter presents two experiments which show how the prosody of synthesised grounding utterances affects their interpretation, as well as users' behaviour in a human-computer dialogue setting.

Part V: Conclusion

Chapter 10: Summary and discussion

In the final chapter, the contributions made in this thesis are summarised, followed by a discussion on how the methods and models presented may be extended and integrated further. Finally, the generalisation of the results to other dialogue systems and domains is discussed.